

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Т.в.о.завідувача кафедри
_____ Рязанцев О. І.

« ____ » _____ 2021 р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

_____ Методи бінарної обробки даних для структурної класифікації зображень _____

Освітньо-кваліфікаційний рівень “Магістр”
Спеціальність 122 “Комп’ютерні науки”

Науковий керівник роботи:

(підпис)

С.О. Сафонова

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Я.О. Критська

(ініціали, прізвище)

Студент:

(підпис)

М.С. Левицький

(ініціали, прізвище)

Група:

КН -19дм

Сєверодонецьк 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень “магістр”
Спеціальність 122 – “Комп'ютерні науки”
(шифр і назва)
Спеціалізація _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Т.в.о.зав. кафедри КНІ
к.т.н., доц. В.С. Кардашук
« _____ » _____ 2020 р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Левицькому Максиму Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Методи бінарної обробки даних для структурної
класифікації зображень

керівник проекту (роботи) Сафонова С.О., к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від « _____ » _____ р. № _____

2. Строк подання студентом роботи 15.01.2021

3. Вихідні дані до роботи Матеріали науково-дослідної практики.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Класифікація зображень на підставі множини ключових точок.

2. Бінарна обробка дескрипторів ключових точок.

3. Моделі оброблення особливих точок на зображенні.

4. Програмна реалізація.

5. Аналіз результатів дослідження.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	<i>Критська Я.О.</i>		

7. Дата видачі завдання 05.10.2020

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання до магістерської роботи	06.10.20-07.10.20	
2	Аналіз завдання, огляд літератури	08.10.20-15.10.20	
3	Аналіз технічних засобів	16.10.20-19.10.20	
4	Розробка методу	20.10.20-06.11.20	
5	Програмна реалізація	17.11.20-08.12.20	
6	Охорона праці	09.12.20-15.12.20	
7	Оформлення пояснювальної записки	16.12.20-29.12.20	
8	Підготовка презентації та доповіді	03.01.21-12.01.21	

Студент

_____ (підпис)

Левицький М.С.

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

Сафонова С.О.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Левицький М.С. Методи бінарної обробки даних для структурної класифікації зображень.

Метою роботи є дослідження та модернізація методу класифікації зображень за допомогою бінарного центрального дескриптора.

Розроблено програмне забезпечення для класифікації зображень варіантами алгоритмів. Проведено аналіз працездатності запропонованих алгоритмів, дослідження на варіативних вибірках та різних первинних обробках вхідних та еталонних зображень.

Ключові слова: класифікація зображень, ключові точки, центральний дескриптор, дескриптор BRISK, вагові коефіцієнти, база еталонів.

АННОТАЦИЯ

Левицкий М.С. Методы бинарной обработки данных для структурной классификации изображений.

Целью работы является исследование и модернизация метода классификации изображений с помощью бинарного центрального дескриптора.

Разработано программное обеспечение для классификации изображений вариантами алгоритмов. Проведен анализ работоспособности предложенных алгоритмов, исследования на вариативных выборках и различных первичных обработках входных и эталонных изображений.

Ключевые слова: классификация изображений, ключевые точки, центральный дескриптор, дескриптор BRISK, весовые коэффициенты, база эталонов.

ABSTRACT

Levitskiy M.S. Methods of binary data processing for structural image classification.

The aim of the work is to research and modernize the method of image classification based on a binary central descriptor.

Software for classification of images by variants of algorithms is developed.

The analysis of the efficiency of the proposed algorithms, the study of variant samples and various initial processing of input and reference images.

Key words: image classification, key points, central descriptor, BRISK descriptor, weights, base.

ЗМІСТ

Перелік умовних позначень	6
Вступ	7
1 Класифікація зображень на підставі множини ключових точок	9
1.1 Основи задачі класифікації	9
1.2 Методи формування ключових точок зображення	11
1.3 Постановка задачі дослідження	18
2 Моделі класифікації зображень з використанням навчання	19
2.1 Задача класифікації для множини бітових рядків	19
2.2 Метод класифікації за значенням центрального дескриптора	24
2.3 Застосування вагових коефіцієнтів	26
3 Результати дослідження	29
3.1 Обґрунтування вибору середовища програмної реалізації	29
3.2 Особливості програмної реалізації	31
3.3 Інструкція користувача	36
3.4 Аналіз результатів дослідження	41
4 Охорона праці	57
4.1 Загальні питання з охорони праці	57
4.2 Аналіз стану умов праці	57
4.2.1 Вимоги до приміщень	58
4.2.2 Вимоги до організації місця праці	58
4.2.3 Навантаження та напруженість процесу праці	59
4.3 Виробнича санітарія	59
4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу	60
4.3.2 Пожежна безпека	61
4.3.3 Електробезпека	62
4.4 Гігієнічні вимоги до параметрів виробничого середовища	62
4.4.1 Мікроклімат	62
4.4.2 Освітлення	63
4.4.3 Шум та вібрація, електромагнітне випромінювання	64
4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій	64
Висновки	69
Перелік джерел посилань	71
Додаток А Тестові зображення	73
Додаток Б Комп'ютерна презентація	74

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

BRIEF – Binary Robust Independent Elementary Features

BRISK – Binary Robust Invariant Scalable Keypoints

EBNF – Extended Backus–Naur form

FAST – Features from Accelerated Segment Test

ID3 – Identify an MP3

SIMD – Single Instruction, Multiple Data

SSE – Streaming SIMD Extensions

STL – Standard Template Library

SURF – Speeded up Robust Features

SUSAN – Smallest Univalued Segment Assimilation Nucleus

SWIFT – Society for Worldwide Interbank Financial Telecommunications

XOR – eXclusive OR, додавання за модулем два

ВК –Вагові коефіцієнти.

ВВС – Відмова від слабкого

МПЦД – Метод побудови центрального дескриптора

ОТ – Особливі точки

ВСТУП

У системах комп'ютерного зору при розпізнаванні візуальних об'єктів за їх структурним описом у вигляді множини дескрипторів особливих точок зображення постає задача побудови функції (правила, процедури) класифікації, що задає відношення еквівалентності на множині дескрипторів бази зображень, де класи представлені окремими описами еталонів. Зважаючи на подібність значень дескрипторів для різноманіття візуальних об'єктів, така еквівалентність на практиці досягається лише наближено [1, 14, 15].

У задачі розпізнавання об'ємних баз візуальних об'єктів у системах комп'ютерного зору важливими показниками є результативність і час оброблення. Тому останнього часу набули прикладного застосування бінарні детектори особливих точок, як BRISK, ORB [7,2], що отримують дескриптор ключових точок у вигляді бінарного вектора з розміром, кратним ступені двійки. Бінарне подання даних значно прискорює процес порівняння дескрипторів за рахунок можливості застосування двійкових операцій та відповідно спрощує апаратну реалізацію системи розпізнавання. Крім того, бінарна арифметика дає потенцію застосувати ефективний апарат оброблення бінарних даних та синтезувати нові підходи для визначення подібності дескрипторів ОТ при побудові правил класифікації.

Метою магістерської роботи є дослідити та реалізувати модернізації методу класифікації зображень за допомогою бінарного центрального дескриптора, провести аналіз працездатності алгоритмів, провести досліди на варіативних вибірках та при різних первинних обробках вхідних чи еталонних зображень. На основі отриманих даних дати оцінку як алгоритмам, так й проведеним дослідом.

Важлива перевага бінарного дескриптора полягає у швидкості та простоті оброблення. Об'єкти у реальному світі характеризується великою кількістю станів кожного параметра системи. В таких системах неможливо визначити точні значення параметрів функціонування. Тоді використовують, наприклад, інтервальні оцінки або нечіткі числа. В нашому випадку ціле зображення приймається як послідовність чітких значень, як дескриптор, що має довжину у 512 (біт) значень, кожне значення може бути 1 або 0. Можливість побудови центру один раз без подальшої його зміни та можливість пришвидшити процес класифікації у стільки разів, скільки ОТ детектується для конкретного зображення, робить даний підхід досить актуальним під час роботи з великим об'ємом даних. Вже не говорячи про можливість порівняння центра з центром, що дозволяє ще додатково пришвидшити процес класифікації в стільки разів, скільки ОТ містить зображення, що порівнюється з множиною еталонних описів.

Мета і задачі дослідження. Метою магістерської роботи є дослідження та розробка методу класифікації зображень за допомогою бінарного центрального дескриптора, що дозволяє пришвидшити процес класифікації.

Для досягнення поставленої мети в магістерській роботі сформульовані та вирішені наступні задачі:

- 1) аналіз недоліків та переваг алгоритму побудови центрального дескриптора;
- 2) розробка алгоритму класифікації з бінарними даними;
- 3) реалізація комп'ютерної моделі детектування особливих точок (ОТ) та обчислення множини дескрипторів;
- 4) модернізація системи тестування та аналізу даних для оцінювання роботи алгоритму;
- 5) комп'ютерне тестування алгоритму на підготовленій реальній базі зображень;
- 6) аналіз результатів дослідження.

Об'єктом дослідження магістерської роботи є бінарна обробка дескрипторів ключових точок, яка дозволить підвищити швидкість та простоту оброблення зображень.

Предметом дослідження є моделі та методи класифікації зображень.

Методи рішення поставлених задач базуються на комплексному використанні математичного апарату, методів числового моделювання та аналітичного обґрунтування.

Наукова новизна одержаних результатів полягає в наступному:

Одержали подальший розвиток методи класифікації зображень.

Практичне значення одержаних результатів.

- проведено дослідження методу класифікації зображень за допомогою бінарного центрального дескриптора;
- отримано практичні програмні моделі для застосування та оцінювання ефективності методів класифікації з бінарним поданням даних та приведенням їх у центри з підтвердженням їх результативності на прикладних базах зображень.

Апробація результатів роботи.

Основні результати магістерської роботи докладалися на Всеукраїнських науково-практичних конференціях «Електроніка та телекомунікації» та «Майбутній науковець 2020».

Публікації.

Основні результати магістерської роботи опубліковано в наукових працях: 2 матеріалах конференцій.

Структура і обсяг роботи.

Магістерська робота складається зі вступу, 4 розділів, висновків на 70 сторінках, списку використаних джерел з 34 найменувань на 2 сторінках, додатків на 9 сторінках. Загальний обсяг роботи складає 81 сторінку. В магістерській роботі міститься 8 таблиць, 47 рисунків .

1 КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ НА ПІДСТАВІ МНОЖИНИ КЛЮЧОВИХ ТОЧОК

1.1 Основи задачі класифікації

Класифікація - один з розділів машинного навчання, присвячений вирішенню проблеми відношення об'єкту до групи конкретних об'єктів, що мають спільні відзнаки від групи інших об'єктів. Є множиною об'єктів (ситуацій), розділених деяким чином на класи. Визначено кінцеву множину об'єктів, для яких відомо, до яких класів вони належать. Ця множина називається навчальною вибіркою. Класова приналежність інших об'єктів не відома. Потрібно побудувати алгоритм, здатний класифікувати довільний об'єкт з початкової множини [22].

Класифікувати об'єкт - значить, вказати номер (або найменування класу), до якого належить даний об'єкт.

Класифікація об'єкта - номер або найменування класу, що видається алгоритмом класифікації в результаті його застосування до даного конкретного об'єкту.

Математичне формулювання завдання наступне: Нехай X — множина описів об'єктів, Y — множина номерів (чи назв) класів. Існує невідома цільова залежність - відображення $y^* : X \rightarrow Y$, значення якої відомі лише на елементах скінченної навчальної вибірки $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$. Потрібно побудувати алгоритм $a : X \rightarrow Y$, здатний класифікувати довільний об'єкт $x \in X$ [9,10].

Загальнішим є ймовірнісне формулювання завдання. Припускається, що множина пар «об'єкт, клас» $X \times Y$ є ймовірнісним простором з невідомою ймовірнісною мірою P . Є скінченна навчальна вибірка спостережень $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$, згенерована згідно з ймовірнісною мірою P . Необхідно побудувати алгоритм $a : X \rightarrow Y$, здатний класифікувати довільний об'єкт $x \in X$ [9,10].

У математичній статистиці завдання класифікації називаються також завданнями дискримінантного аналізу.

У машинному навчанні завдання класифікації відноситься до розділу навчання з учителем. Існує також навчання без вчителя, коли поділ об'єктів навчальної вибірки на класи не задається, і потрібно класифікувати об'єкти тільки на основі їх подібності між собою. У цьому випадку прийнято говорити про завдання кластеризації або таксономії, і класи називати, відповідно, кластерами або таксонами [22].

Різновидності вхідних даних у задачі класифікації [22]:

- ознаковий опис - найбільш поширений випадок. Кожен об'єкт описується набором своїх характеристик, які називаються ознаками. Ознаки можуть бути числовими або нечисловими;
- матриця відстаней між об'єктами. Кожен об'єкт описується відстанями до всіх інших об'єктів навчальної вибірки. З цим типом вхідних даних працюють деякі методи, зокрема, метод найближчих сусідів, метод парзеновського вікна, метод потенційних функцій;
- часовий ряд або сигнал являє собою послідовність вимірювань в часі. Кожен вимір може представлятися числом, вектором, а в загальному випадку - признаковим описом досліджуваного об'єкта в даний момент часу;
- зображення або відеоряд;
- зустрічаються і більш складні випадки, коли вхідні дані подаються у вигляді графів, текстів, результатів запитів до бази даних, й т.д. Як правило, вони наводяться до першого або другого випадку шляхом попередньої обробки даних і вилучення ознак.

Типи класів [26]:

- двокласова класифікація. Найбільш простий в технічному відношенні випадок, який служить основою для вирішення більш складних завдань;
- багатокласова класифікація. Коли число класів досягає багатьох тисяч (наприклад, при розпізнаванні ієрогліфів або злитого мовлення), завдання класифікації стає істотно більш важким;
- класи, що не перетинаються;
- класи, що перетинаються. Об'єкт може належати одночасно до кількох класів;
- нечіткі класи. Потрібно визначати ступінь належності об'єкта кожному з класів, зазвичай це дійсне число від 0 до 1.

Правила класифікації [22]:

- ділити потрібно по одній конкретній основі. Якщо порушити це правило, відбудеться перетин понять і плутанина. Автомобілі бувають декількох видів: хетчбеки, седани, червоні, чорні, білі, що працюють на бензині і газі. Тут змішали аж три ознаки: тип кузова, колір, вид споживаного палива;
- класифікація повинна бути повною. Якщо складемо всі підвиди, то повинен вийти вид. Це означає, що при розбитті важливо не забути вказати всі частини. Лікарі діляться на стоматологів, терапевтів, педіатрів та кардіологів. Так? Не так. Забули хірургів, косметологів і ще багато кого;
- класифікація не повинна містити зайвих елементів. Типова помилка, коли при поділі одного поняття перестаралися і додали туди предмет, який не має відношення до цілого.

Наприклад, цитрусові бувають такі: апельсини, лимони, банани. Цілком очевидно, що банан - не цитрус. Цей елемент тут зайвий;

- елементи повинні взаємно виключати один одного. Створили в пам'яті комп'ютера дві папки - «Пісні» і «Музика». Тепер треба роздумувати, куди зберігати альбом улюбленої музичної групи;
- ділячи поняття, слід діяти по порядку, безперервно переходити від одного підкласу до іншого найближчого. Перестрибувати не можна.

1.2 Методи формування ключових точок зображення

Особливі точки – це те ж саме, що і точки інтересу на зображенні [5]. Це просторові розташування або точки в зображенні, які визначають, що цікаво чи що виділяється на зображенні.

Причина, по якій ці точки є особливими, полягає в тому, що незалежно від того, як змінюється зображення, стискається або розширюється, (все це будуть афінні трансформації) або схильна до спотворення (тобто проєктивне перетворення або homography), все одно можна знайти ті ж ключові точки в цьому зміненому зображенні в порівнянні з вхідним зображенням.

FAST - це алгоритм, запропонований Ростеном та Драммодом [16] для виявлення ОТ у зображенні. ОТ зображення – це піксель, який має чітко визначену позицію і може бути надійно виявлений. ОТ мають високий інформаційний вміст у своєму форматі та ідеально повторюються між різними зображеннями [15]. Визначення ОТ має програмну реалізацію для знаходження збігу зображень, розпізнавання об'єктів, відстеження, тощо.

Ідея виявлення ОТ або виявлення кутів (обидва використовуються в літературі), не нова. Існують і інші алгоритми, такі як: Moravec алгоритм виявлення кутів, алгоритм виявлення кутів Harris & Stephens, детектор куту SUSAN. Причиною створення алгоритму FAST було створення детектора ОТ для використання в програмах реального часу, таких як SLAM на мобільному роботі, що мають обмежені обчислювальні ресурси [17].

Алгоритм FAST полягає в наступних кроках.

1. Вибрати піксель p у зображенні. Припустимо, що інтенсивність цього пікселя має бути I_p . Це є піксель, який слід визначити як ОТ, або визначити протилежне (рис. 1.1).

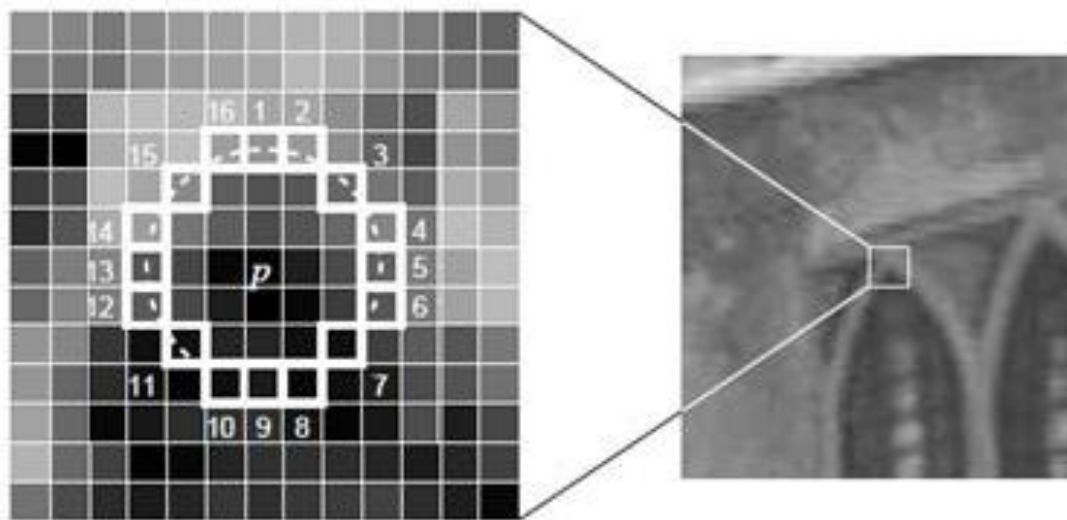


Рисунок 1.1 – Точка-кандидат й оточуючі з 16 пікселів навколо неї

4. Помітні пікселі N з 16, повинні бути вище або нижче I_p за значенням T , якщо піксель потрібно визначити як ОТ. (Автори використовували $N = 12$ у першій версії алгоритму)

5. Щоб зробити алгоритм швидким, спочатку треба порівнювати інтенсивність пікселів 1, 5, 9 і 13 кола з I_p . Як можна побачити з наведених вище цифр, принаймні три з цих чотирьох пікселів повинні задовольняти пороговому критерію, щоб існувала ОТ.

6. Якщо принаймні три з чотирьох значень пікселів - I_1 , I_5 , I_9 чи I_{13} не вище або нижче $I_p + T$, то p не є ОТ (кутом). У цьому випадку треба відхилити піксель p як можливу ОТ. Іншими словами, якщо принаймні три пікселі перебувають вище або нижче $I_p + T$, то треба перевірити або всі 16 пікселів й перевірити, чи 12 сусідніх пікселів потрапляють у критерій.

7. Повторення процедури для всіх пікселів у зображенні.

Існує декілька обмежень алгоритму. По-перше, для $N < 12$ алгоритм працює не дуже добре, оскільки при $N < 12$ кількість виявлених ОТ дуже висока. По-друге, порядок, в якому запитується 16 пікселів, визначає швидкість алгоритму.

Підхід до машинного навчання був доданий до алгоритму для вирішення цих проблем [3,5].

1. Вибрати набір зображень для тренувань.

2. У кожному зображенні запускається алгоритм FAST для виявлення ОТ, приймаючих один піксель одночасно та оцінюючи всі 16 пікселів у колі.

3. Для кожного пікселя p зберігати 16 пікселів навколо нього, як вектор (рис. 1.2).

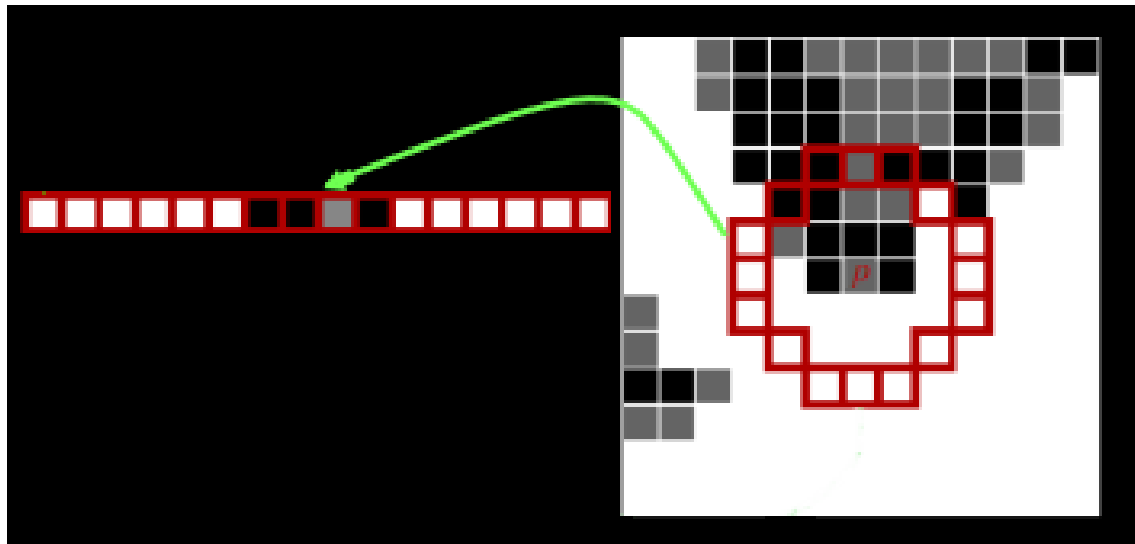


Рисунок 1.2 – 16 значень, що оточують піксель p , зберігаються у векторній формі
(зображення скопійовано з [5])

4. Повторити це для всіх пікселів у всіх зображеннях. Це вектор P , який містить всі дані для розпізнавання (зверніть увагу на різницю між p - пікселем та P - вектор).

5. Кожне значення (один з 16 пікселів, скажімо x) у векторі, може мати три стани. Темніший, ніж p , світліший за p або аналогічний p .

Математично,

$$S_{p \rightarrow x} = \begin{cases} d, & I_{p \rightarrow x} \leq I_p - t \text{ (darker)} \\ s, & I_p - t < I_{p \rightarrow x} < I_p + t \text{ (similar)} \\ b, & I_p + t \leq I_{p \rightarrow x} \text{ (brighter)} \end{cases} \quad (1.1)$$

$S_{p \rightarrow x}$ – це стан, $I_{p \rightarrow x}$ – інтенсивність пікселя x , t - поріг.

6. Залежно від стану весь вектор P буде підрозділятися на три підмножини P_d , P_s , P_b .

7. Визначити змінну K_p , яка є істиною, якщо $p \in OT$ або помилковою, якщо $p \notin OT$.

8. Використовувати алгоритм ID3 (класифікатор дерева рішень) для запиту кожного піднабору, використовуючи змінну K_p для знань про справжній клас.

9. Алгоритм ID3 працює за принципом мінімізації ентропії. Запитувати 16 пікселів таким чином, щоб знайти справжній клас (є точка OT чи ні) з мінімальною кількістю запитів. Або іншими словами, вибрати піксель x , який містить найбільшу інформацію про піксель p . Ентропія для вектора P може бути математично представлена як

$$H(P) = (c + \bar{c}) - c \log_2 c - \bar{c} \log_2 \bar{c}$$

where $c = \left| \left\{ p \mid K_p \text{ is true} \right\} \right|$ (number of corners)

and $\bar{c} = \left| \left\{ p \mid K_p \text{ is false} \right\} \right|$ (number of non corners)

(1.2)

10. Рекурсивно застосовувати цю мінімізацію ентропії до всіх трьох підмножин.

11. Припинити процес, коли ентропія підмножини дорівнює нулю.

12. Цей порядок запитів, який вивчається деревом рішень, можна також використовувати для швидкого виявлення в інших зображеннях.

Мета створення BRIEF-дескриптора (Binary Robust Independent Elementary Features) полягала в тому, щоб забезпечити розпізнавання однакових ділянок зображення, які були зняті з різних точок обзору. При цьому ставилося завдання максимально зменшити кількість виконуваних обчислень. Алгоритм розпізнавання зводиться до побудови випадкового лісу (randomize classification trees) або наївного байєсівського класифікатора на деякій тренувальній множині зображень з подальшою класифікацією ділянок тестових зображень. У спрощеному варіанті може використовуватися метод найближчого сусіда для пошуку найбільш схожого патча в тренувальній вибірці. Невелика кількість операцій забезпечується за рахунок подання вектора ознак у вигляді бінарного рядка, а як наслідок, використання в якості міри схожості відстані Хемінга.

Схема побудови векторів ознак досить проста. Зображення розбивається на патчі (окремі перекриті ділянки). Припустимо патч P має розміри $S * S$ пікселів. Із патча деяким чином вибирається множина пар пікселів $\{(X, Y), \forall X, Y \text{ в околиці}\}$ для яких будується набір бінарних тестів:

$$\tau(P, X, Y) = \begin{cases} 1, & I(X) < I(Y) \\ 0, & \text{інакше} \end{cases}, \quad (1.3)$$

де $I(X)$ - інтенсивність пікселя X . Для кожного патча вибирається множина, що містить n_d пар точок, які однозначно визначають набір бінарних тестів. Далі на підставі цих тестів будується бінарний рядок:

$$f_{n_d}(P) = \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(P, X_i, Y_i). \quad (1.4)$$

Автори наводять результати експериментів (якості розпізнавання) при виборі пар точок відповідно до закону рівномірного розподілу в патчі, а також нормального розподілу з різними значеннями математичного очікування і середньоквадратичного відхилення. Відзначимо, що при однакових умовах проведення експериментів на деяких тестових зображеннях точність детектування за допомогою BRIEF майже в 1.5 рази вище, ніж з використанням SURF-дескрипторів.

Почнемо з наступного зображення, що показує приклад використання BRISK для порівняння зображення без змін й над яким провели деякі перетворення (рис.1.3). Зелені лінії - дійсні співвідношення між точками, червоні кола виявляються ключовими точками.

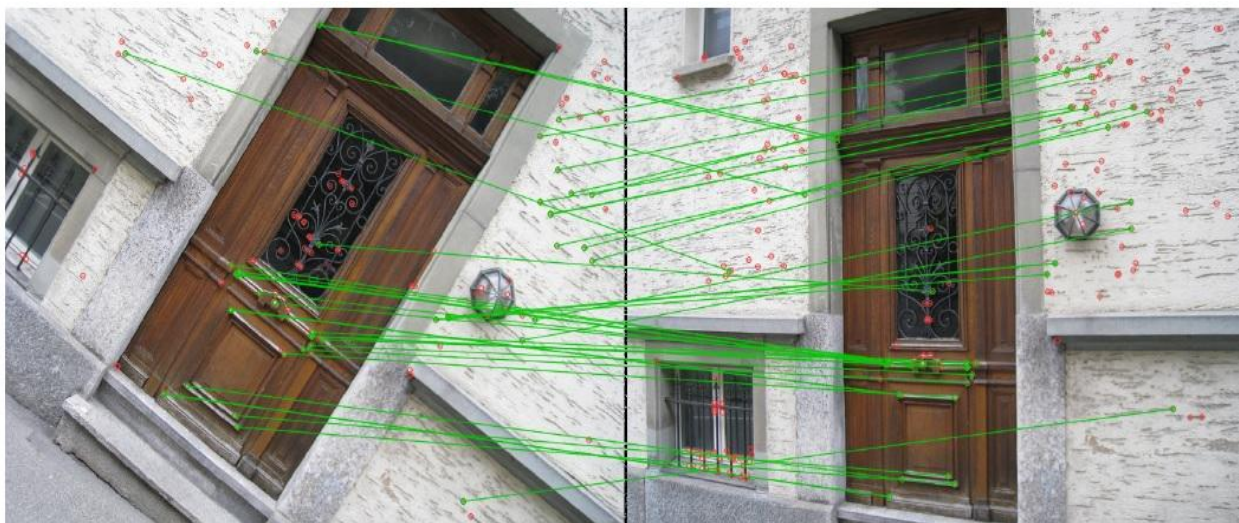


Рисунок 1.3 – Дескриптор BRISK - приклад точок збігу за допомогою BRISK

Двійковий дескриптор BRISK складається з трьох частин:

1. Зразок вибірки: де взяти зразок точок у регіоні навколо дескриптора.
2. Компенсація орієнтації: якийсь механізм для вимірювання орієнтації ОТ та його обертання, щоб компенсувати зміни обертання.
3. Пари вибірки: які пари порівнювати при побудові остаточного дескриптора.

Щоб побудувати двійковий рядок, що представляє регіон навколо ключової точки, потрібно перейти за всіма парами і для кожної пари ($p1$, $p2$) - якщо інтенсивність у точці $p1$ перевищує інтенсивність у точці $p2$, пишемо 1 в двійковий рядок і 0 в іншому випадку.

Дескриптор BRISK є модифікованим алгоритмом FAST (розділ 1.2) та має ручний шаблон вибірки. Зразки вибірки BRISK складаються з концентричних кілець (рис.1.4):

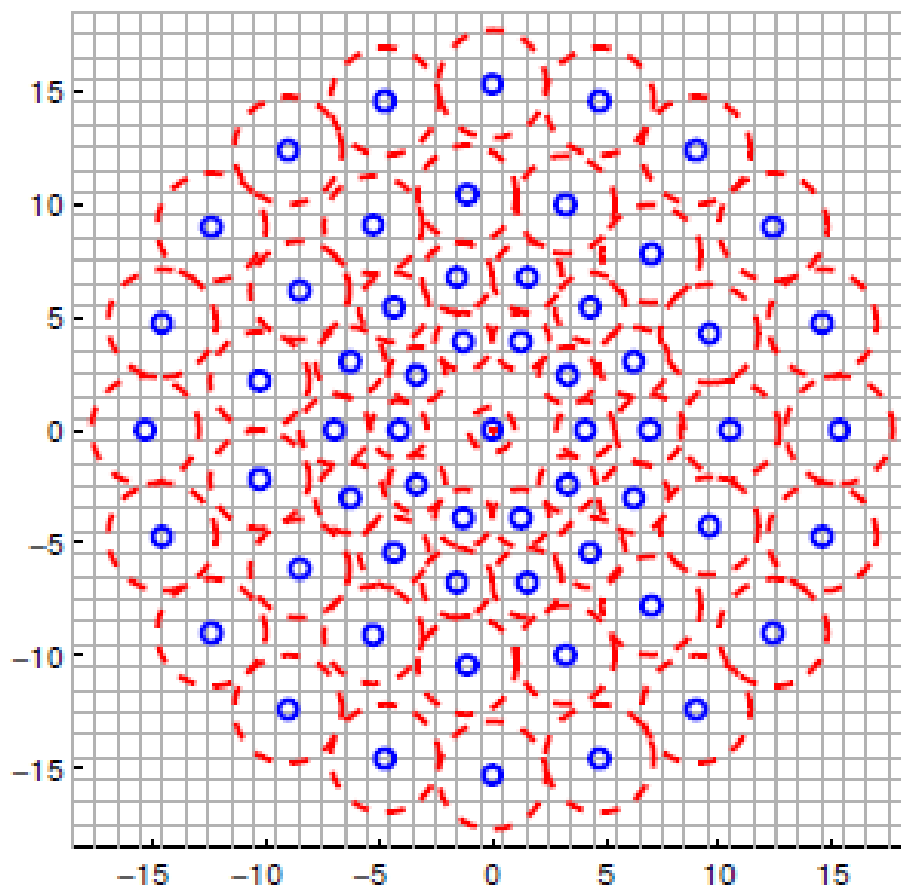


Рисунок 1.4 – Дескриптор BRISK - схема вибірки BRISK

Розглядаючи кожну точку відбору, беремо навколо неї невеликий патч і застосовуємо гаусове згладжування. Червоне коло на рисунку - розмір стандартного відхилення гаусового фільтра, застосованого до кожної точки. Використовуючи цей шаблон вибірки, розрізняємо короткі пари та довгі пари. Коротка пара - це пара точок відбору проб, що їхня відстань нижча за певну порогову величину d_{\max} , а довгими парами є пари точок відбору проб, що їх відстань перевищує певний інший поріг d_{\min} , де $d_{\min} > d_{\max}$, так що немає коротких пар, які також є довгими парами

Довгі пари використовуються в BRISK для визначення орієнтації, а короткі пари використовуються для порівняння інтенсивності, що складають дескриптор, як у BRIEF (пункт 1.3) на основі якого й було збудовано алгоритм порівняння ОТ у BRISK.

Щоб проілюструвати це і зрозуміти, рисунок 1.5 – фігура коротких пар BRISK - кожна червона лінія представляє одну пару. Кожна фігура показує 100 пар.

BRISK оснащений механізмом компенсації орієнтації; намагаючись оцінити орієнтацію ключової точки та обертання шаблону вибірки за цією орієнтацією, BRISK стає дещо інваріантним до обертання.

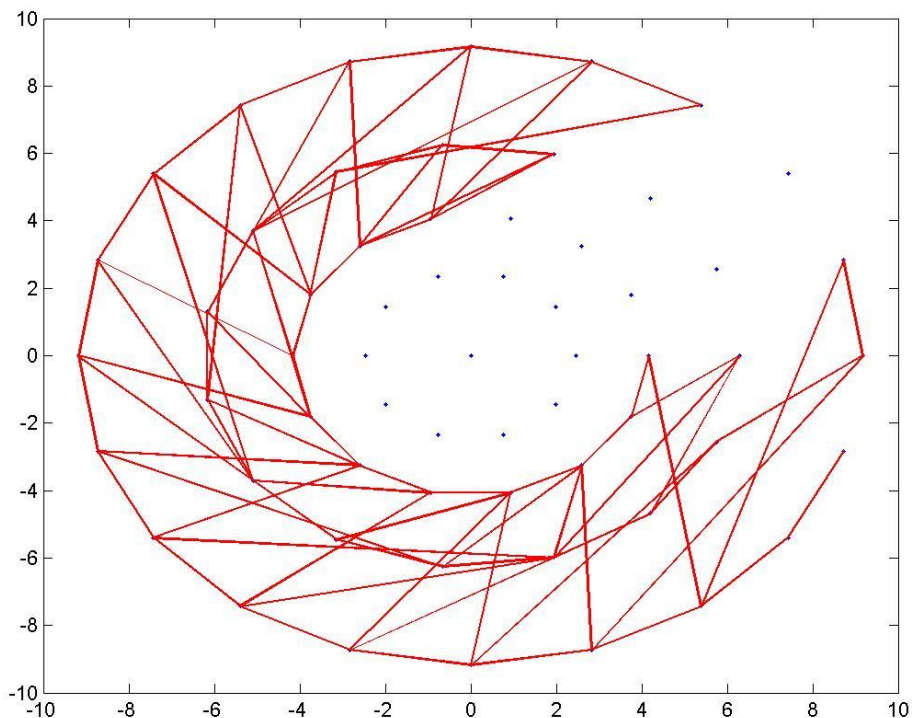


Рисунок 1.5 – Дескриптор BRISK, коротка пара BRISK

Для обчислення орієнтації ключової точки BRISK використовує локальні градієнти між парами вибірки, які визначаються за допомогою

$$g(p_i, p_j) = (p_j - p_i) \times \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2}, \quad (1.5)$$

де $g(p_i, p_j)$ – локальний градієнт між парою вибірки (p_i, p_j) , I - згладжена інтенсивність у відповідній точці відбору з урахуванням відповідного стандартного відхилення.

Щоб обчислити орієнтацію, підсумовуємо локальні градієнти між усіма довгими парами і вираховуємо $\arctan(g_y / g_x)$ - де арктангенс y - компоненти градієнта, поділений на x компоненту градієнта. За допомогою арктангенсу вираховуємо кут даного арктангенсу. Далі потрібно повернути короткі пари за цим кутом, щоб дескриптор став інваріантним до обертання. BRISK використовує лише довгі пари для обчислення орієнтації, виходячи з припущення, що локальні градієнти скасовують один одного, таким чином, вони не є необхідними для визначення глобального градієнта.

Як і у випадку з усіма двійковими дескрипторами, побудова дескриптора здійснюється шляхом порівняння інтенсивності. BRISK бере набір коротких пар, обертає пари орієнтацією, обчисленою раніше, і робить порівняння форми:

$$b = \begin{cases} 1, & I(p_j^\alpha, \sigma_j) > I(p_i^\alpha, \sigma_i) \\ 0, & \text{otherwise} \end{cases}. \quad (1.6)$$

Це означає, що для кожної короткої пари він згладжує інтенсивність точок відбору проб і перевіряє, чи згладжена інтенсивність першої точки в парі більша, ніж у другій точці. Якщо це так, то він записує 1 у відповідний біт дескриптора та іншим чином 0. BRISK використовує лише короткі пари для побудови дескриптора.

Як завжди, відстань між двома дескрипторами визначається як кількість різних бітів двох дескрипторів, і їх можна легко обчислити як суму оператора XOR між ними.

1.3 Постановка задачі дослідження

Класифікація зображень за допомогою бінарних центрів є актуальною задачею завдяки своїй простоті та забезпеченню високої швидкодії. Ставиться завдання розробки та дослідження алгоритмів побудови бінарного центрального дескриптора з його подальшою обробкою для отримання оцінки швидкодії та якості класифікації зображень. Для отримання ОТ з бінарними дескрипторами будемо використовувати алгоритм BRISK. Він має довжину дескриптора у 512 біт. Планується модернізувати даний алгоритм ваговими коефіцієнтами. Далі планується порівняти нові алгоритми з методом побудови центрального дескриптора та отримати порівнювальні оцінки працездатності та конкурентоздатності алгоритмів.

Потрібно програмно реалізувати алгоритмічну частину та розробити програму, що буде використовувати алгоритми для розпізнавання зображень у базі еталонів, провести досліди, на основі яких можна буде провести оцінювання результативності.

Для цього необхідно вирішити такі завдання:

- провести аналіз недоліків та переваг алгоритму побудови центрального дескриптора;
- розробити нові алгоритми чи модернізацію побудованого алгоритму класифікації з бінарними даними;
- реалізувати комп'ютерну модель детектування ОТ та обчислення множини дескрипторів;
- модернізувати систему тестування та аналізу даних для оцінювання роботи алгоритму;
- провести комп'ютерне тестування алгоритму на підготовленій реальній базі зображень;
- після побудови графіків встановити оцінки працездатності та конкурентоздатності для алгоритмів з розглядом їх подальшої модернізації;
- провести аналіз результатів дослідження.

2 МОДЕЛІ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ НАВЧАННЯ

2.1 Задача класифікації для множини бітових рядків

Нехай $Z = \{Z^j\}_{j=1}^J$ – це множина дескрипторів бази зображень, яку складають описи еталонів, J – число класів, Z^j – еталон, $Z \subseteq B^n$, B^n – множина бінарних векторів (ланцюжків, рядків) розмірності n , $s(j) = \text{card } Z^j$ – кількість елементів еталону Z^j . Кількість розпізнаваних класів визначається числом J еталонів.

Розглянемо правило класифікації $L: B^n \rightarrow [1, \dots, J]$ у вигляді функції, що $\forall z \in Z$ виконується $L(z) \in [1, \dots, J]$, так що кожний елемент буде зараховано до одного із еталонів Z^j . При цьому конструктивно та параметрично функція L опирається на апіорні дані конкретної бази через те, що приналежність кожного з елементів $z \in Z$ до відповідного еталону всередині бази задана до початку класифікації.

Для побудови методу класифікації застосуємо дискретну модель простору B^n ознак [14], де для кожного з еталонів Z^j попередньо сформуємо так званий «центр класу» або «центр маси класу», $m_j \in B^n$, $j = 1, \dots, J$. У процесі здійснення класифікації довільний вектор $b \in B^n$ будемо конкурентно відносити до класу ν у відповідності до найменшої відстані

$$\nu = \arg \min_{j=1, \dots, J} \rho(b, m_j), \quad (2.1)$$

де $\rho(b, m_j)$ – метрика у просторі B^n .

Якщо розпізнавання відбувається на універсумі B^n , то в даному випадку правило (2.1) треба параметрично доповнити логічним аналізом із застосуванням порогу [2]: віднесення дескриптора O до класу ν здійснюється тільки у тому випадку, коли для вектора b буде також виконано умову $\rho(b, m_j) \leq \delta$, де δ – заданий поріг еквівалентності елементів у просторі. Якщо таке порогове обмеження до класу відсутнє, то буде віднесено зовсім не схожий елемент, але мінімум в (2.1) буде завжди досягнуто.

Відповідно з результатами у запропонованій процедурі класифікації множина дескрипторів $O = \{o_1, \dots, o_v\}$, $o_i \in B^n$, що описує довільний візуальний об'єкт, буде подана у вигляді цілочисельного вектора $h[O] = \{h_1, \dots, h_J\}$, де $h_j = \text{card } \{o_i \in O \mid o_i \rightarrow Z^j\}$ – кількість

елементів еталону, віднесених до класу з номером j . Одним із методів визначення класу об'єкта є парадигма

$$d = \arg \max_{j=1, \dots, J} h_j . \quad (2.2)$$

Правило (2.2) безпосередньо можна застосувати тільки у ситуації, де для подання $h[Z^j]$ кожного з еталонів, максимум відповідає номеру j еталона, що класифікується. Коли ця умова не виконується й для даних бази еталонів, то необхідно застосовувати інші, більш універсальні підходи [1]. У той же час правила класифікації, побудовані із застосуванням (2.2), дають ефективну можливість суттєво спростити прийняття рішення через те, що кожний із структурних елементів може безпосередньо бути класифікований без додаткового аналізу їх сукупних властивостей. Застосування бінарного оброблення та подання дає можливість синтезувати такий спосіб.

Взагалі підхід з визначенням центрів класів, забезпечує значне збільшення швидкодії класифікації за рахунок переходу від обчислення міри релевантності множин до порівняння векторів, або окремих їх елементів [6]. Бінарне подання даних суттєво спрощує процес обробки на програмному рівні, даючи можливість, наприклад в (2.1), застосовувати метрику Хемінга замість більш суттєвих за обсягом обчислень метрик.

Застосування методів кластеризації є традиційним підходом до обчислення центрів, спираючись на ознаки, на основі яких різняться значення дескрипторів ОТ зображення (рис. 2.1). Беручи до уваги те, що раніш практикувалося використання дескрипторів SIFT та SURF, які взагалі нормовані. Вагомість норми для дескрипторів BRISK не є достатньо значимою характеристикою окремого класу, так для кластеризації застосовувалися безпосередньо значення елементів множини Z [14].

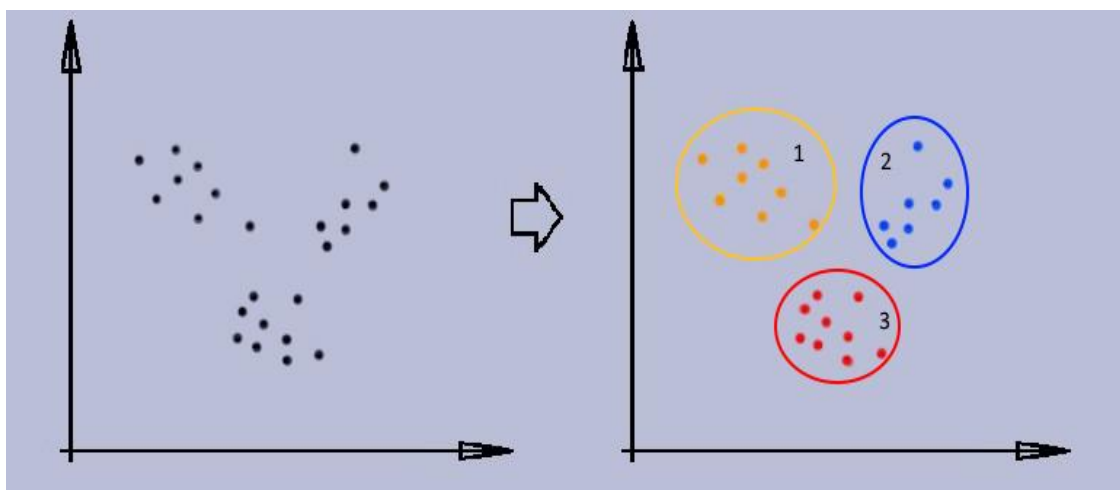


Рисунок 2.1 – Приклад розбиття точок по кластерам

У межах апарату бінарних даних викликає інтерес застосування націленого оброблення множини дескрипторів із визначенням одного з елементів, що буде представляти множину як клас. Двійковий вид дескрипторів дає підставу застосувати апарат логічного оброблення. Зважаючи на бінарний вид BRISK-дескрипторів, для кожного з еталонів Z^j визначимо бінарний вектор m_j центра класу на підставі такого логічного правила

$$m_j(a) = \begin{cases} 1, & f(Z^j, a) \geq s(j) / 2, \\ 0, & \text{інакше,} \end{cases} \quad (2.3)$$

де $m_j(a)$ – значення біту на позиції з номером a для центру m_j , $f(Z^j, a)$ – функція, що підраховує кількість одиничних бітів на позиції з номером a у множині дескрипторів еталону Z^j , $a = 1, \dots, n$.

Функція $f(Z^j, a)$ може бути обчислена безпосередньо додаванням бітів

$$f(Z^j, a) = \sum_{d=1}^{s(j)} x_d(a), \quad x_d \in Z^j, \quad (2.4)$$

де $x_d(a)$ – біт з номером a для дескриптора з номером d в описі еталона.

За виразом (2.3) значення кожного з бітів центру m_j для j -го класу визначається переважно більшістю значень відповідних розрядів усіх дескрипторів ОТ, які належать еталону Z^j . Бітове подання можна ефективно використати при роботі з множинами даних. Підрахунок числа одиничних бітів у (2.4) можна здійснити також застосуванням відповідних бінарних функцій [5].

Аналізуємо елементи $x \in Z$ загального вмісту структурних описів бази еталонів (навчальна вибірка) шляхом віднесення їх до відповідного класу з використанням конкуренції (1). У якості ρ застосуємо метрику Хемінга

$$\rho(x, m_j) = \sum_{a=1}^n |x(a) - m_j(a)|, \quad (2.5)$$

що визначає кількість розбіжних бітів для двійкових послідовностей однакової довжини.

Відстань Хемінга - число позицій, в яких відповідні символи двох слів однакової довжини різні. У більш загальному випадку відстань Хемінга застосовується для рядків однакової довжини

будь-яких q -ічних алфавітів і служить метрикою відмінності (функцією, яка визначає відстань в метричному просторі) об'єктів однакової розмірності (рис. 2.2).

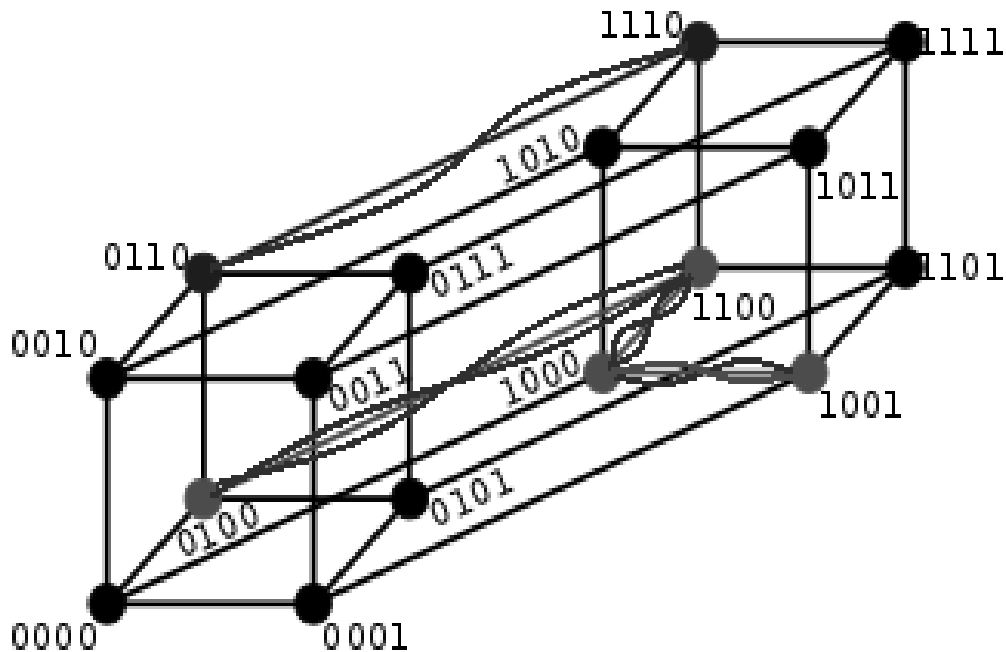


Рисунок 2.2 – Приклади відстаней в двійковому Тессеракті: відстань 1 (подвійна риска) $0110 \rightarrow 1110$, відстань 3 (потрійна риска) $0100 \rightarrow 1001$

У результаті оброблення з використанням попередньо обчислених центрів одержуємо подання для кожного еталона: $h[Z^i] = (h_1^i, \dots, h_j^i)$, яке відповідає розподілу елементів множини Z^i за класами еталонів, де h_a^i – цілі числа, причому $\sum_{a=1}^{s(i)} h_a^i = s(i)$, так як кожний з елементів опису буде віднесено до одного з фіксованих класів.

Важливою характеристикою множини отриманих центрів є симетрична матриця відстаней між ними $\Upsilon = \{\rho(m_i, m_j)\}_{i,j=1}^J$, яка відображає їх сумісну наближеність у просторі B^n та фактично визначає фінальну якість класифікації. Чим більше віддалені центри між собою, тим вище досяжна ступінь розрізненості дескрипторів із різних класів.

Загальним же критерієм якості квантування простору у відповідності до апріорних даних – опису бази зображень у вигляді множин дескрипторів ОТ може бути нормоване значення розкиду даних навколо сформованих центрів

$$E = \frac{1}{sN} \sum_{j=1}^J \sum_{v=1}^{s(j)} \rho(x_v, m_j), \quad (2.6)$$

де N – розмір дескриптора ($N = 512$ для BRISK), $s = \sum_{j=1}^J s(j)$ – загальне число елементів

навчальної вибірки, а для спрощення та приведення до інтервалу 0...1 замість традиційного квадрата метрики вибрано її значення [18].

Другим критерієм при умові правильної класифікації є відношення загального числа дескрипторів, які за результатом оброблення віднесені до правильного класу, до гуртового числа дескрипторів бази

$$\beta = \sum_{i=1}^J h_i^i / s. \quad (2.7)$$

Критерій (2.7) – це відношення кількості правильно класифікованих дескрипторів бази до їх загального числа. Ідеальним варіантом є значення $\beta = 1$.

Зважаючи на бітове подання даних, виникає можливість також застосувати по-байтовий аналіз, використавши замість виразів (2.5), (2.6) міри, обчислені на підставі подібності окремих байтів.

З метою порівняння показників класифікації також розглянемо у якості центру множини її узагальнену медіану. Медіана в математичній статистиці - число, що характеризує вибірку (наприклад, набір чисел). Якщо всі елементи вибірки різні, то медіана - це таке число вибірки, що рівно половина з елементів вибірки більше нього, а інша половина менше нього. У більш загальному випадку медіану можна знайти, упорядкувавши елементи вибірки за зростанням або спаданням і взявши середній елемент (рис. 2.3). Наприклад, вибірка {11, 9, 3, 5, 5} після упорядкування перетворюється в {3, 5, 5, 9, 11} і її медіаною є число 5. Якщо у вибірці парне число елементів, медіана може бути не визначена однозначно: для числових даних найчастіше використовують напів-суму двох сусідніх значень (тобто медіану набору {1, 3, 5, 7} приймають рівною (2.4).

Узагальнена медіана m скінченної множини $X=\{x(i)\}$ визначається шляхом мінімізації функціоналу D :

$$m = \arg \min_{v \in X} D(v), \quad D(v) = \sum_{x(i) \in X} \rho(x(i), v), \quad (2.8)$$

де $\rho(x(i), v)$ – відстань між елементами. Обчислити медіану можна шляхом агрегації значень рядків матриці взаємних відстаней елементів множини [3]. У результаті обчислення (2.8) отримуємо медіану m_j як один із елементів еталону $m_j \in Z^j$.

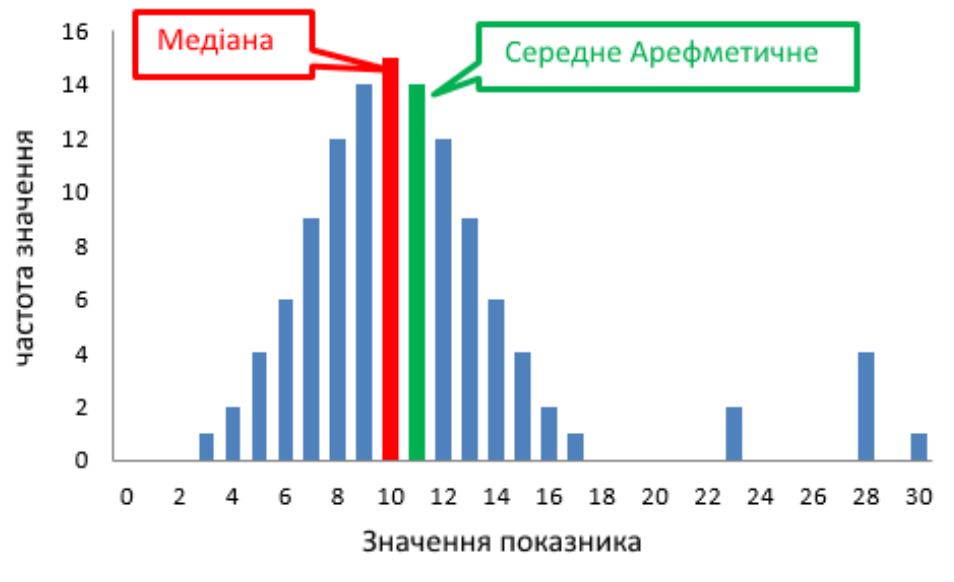


Рисунок 2.3 – Порівняння медіани з середнім значенням

Вектор $h[Z^i]$ також може виступити критерієм якості класифікації на навчальній вибірці. Ідеальним класифікатором є той, для якого $\arg \max_j h_j^i = i$, тобто при реалізації (2.2) максимальне значення досягається для номера еталону, що класифікується.

2.2 Метод класифікації за значенням центрального дескриптора

Даний алгоритм був розроблений шляхом експериментальної модернізації над моделлю навчання Кохонена. Експериментальним шляхом було доказано перевагу побудови найбільш приближеного до ідеального центру кластера (центрального дескриптора), як за швидкістю так й за результатами класифікації. Також варто зазначити, що в даному випадку послідовність надходження даних не впливає на побудову центру, що робить алгоритм більш стійким.

Нехай задана навчальна множина даних $W = \{x \mid x \in R^n\}$, $W \subseteq R^n$, що складає набір дескрипторів ОТ бази еталонів. Тепер побудуємо кортеж k опорних (модельних) векторів $M = \{m_i\}_{i=1}^k$, $m_i \in R^n$, $i = 1, 2, \dots, k$. Для сформованого набору M апроксимація довільного вектора $x \in W$ конкурентним способом означає визначення номера v найближчого до нього (у метриці $\rho(x, m_v)$) вектора $m_v \in M$ у просторі модельних векторів:

Модель (2.1) називають конкурентним навчанням Кохонена [14]. Для традиційного варіанту онлайн-навчання, якщо дескриптори ОТ $x[t] \in W$ поступають по черзі, центр m_v

кластера, який став переможцем в (1), на кроці $t = 1, 2, \dots, s$ навчання коригується у відповідності до виразу

$$m_v[t+1] = m_v[t] + \alpha[t](x[t] - m_v[t]), \quad (2.9)$$

де $s = \text{card } W$ – повний обсяг навчальної вибірки (загальне число ОТ бази зображень), а параметр $\alpha[t]$ задається дослідником та визначає швидкість навчання, причому $\alpha[t] \rightarrow 0$ при $t \rightarrow s$. Існує величезне різноманіття стратегій та технологій навчання (2.9), включаючи навіть моделювання динаміки топології мережі [1, 21].

У випадку, коли число k кластерів дорівнює числу еталонів бази зображень, кластерування виконує класифікацію на k класів у просторі W . Множина дескрипторів ОТ, що описує довільний візуальний об'єкт, може бути класифікована як один із еталонів.

Зауважимо, що якість класифікації зображень, як правило, оцінюється іншим критерієм, наприклад, значенням ймовірності правильного розпізнавання об'єктів.

Тепер перейдемо до нашого більш простого евристичного підходу бінарного аналізу (Метод класифікації за допомогою центрального дескриптора), суть якого полягає в наступному:

1. Зважаючи на бінарний вид дескрипторів BRISK, для кожного з еталонів Z^i визначимо вектор m_i центра класу на підставі логічного правила як

$$m_i(\mathbf{b}) = \begin{cases} 1, & \sum_{d=1}^{s(i)} x_d(\mathbf{b}) \geq s(i) / 2, \\ 0, & \sum_{d=1}^{s(i)} x_d(\mathbf{b}) < s(i) / 2, \end{cases} \quad x_d \in Z^i, \quad b = 1, \dots, 512, \quad (2.10)$$

де $x_d(\mathbf{b})$ – біт з номером b для дескриптора з номером d в описі еталона.

Таким чином значення кожного з бітів центру m_i для i -го класу визначається переважною більшістю значень відповідних бітів усіх дескрипторів ОТ, які належать еталону Z^i .

2. Аналізуємо елементи $x \in Z$ загального вмісту структурних описів бази еталонів (навчальна вибірка) шляхом віднесення їх до відповідного класу з використанням конкурентного способу (2.1). Тут у якості $\rho(x, m_i)$ застосуємо метрику Хемінга

$$\rho(x, m_i) = \sum_{b=1}^{512} |x(\mathbf{b}) - m_i(\mathbf{b})|, \quad (2.11)$$

що визначає кількість розбіжних бітів для двох двійкових послідовностей однакової довжини (детекторів ОТ).

У результаті оброблення для кожного еталона отримуємо його кластерне подання $H[Z^i] = (h_1^i, \dots, h_j^i)$, де h_a^i – цілі числа, яке відповідає розподілу елементів множини Z^i за класами еталонів.

Етап попередньої обробки на базі бінарного аналізу можна вважати різновидом хещування на множині дескрипторів ОТ еталону. Аналогічну обробку можна використати і для аналізу множини ORB-дескрипторів.

Порівнюємо кожен дескриптор зображення, що надійшло до алгоритму з усіма кластерами, що знаходяться в базі. Нехай буде чотири кластери, а кожне зображення має по 40 дескрипторів. Коли перший дескриптор надійшов до алгоритму, то за відстанню Хемінга порівнюємо його з усіма кластерами й обираємо той, відстань до якого буде найменшою. Нехай перший дескриптор надійшов до 3-ого кластеру, на відміну від Кохонена, цей алгоритм не змінює центр кластеру, а продовжує свою роботу. Так само надходять один за одним інші дескриптори доки вони не закінчаться. Після того, як усі 40 дескрипторів з прикладу будуть розкидані по кластерам, починаємо підраховувати скільки дескрипторів надійшло до кожного кластеру. Нехай до першого надійшло 3 дескриптори, до другого 2, до третього 30, а до четвертого 5. Таким чином буде вирішено, що зображення буде віднесено до 3-го кластеру. В нашому випадку 75% дескрипторів буди віднесені до 3-го кластеру - чим більший відсоток, тим більша точність розрахунків й ймовірність того, що робота алгоритму була вірна. При випадку коли декілька кластерів отримали однакову кількість дескрипторів, або різниця у відсотковому співвідношенні між ними менше 20%, це говорить про невдало підібрані параметри алгоритму BRISK, або невдало обрані еталонні зображення; а також про те, що здатність гарної роботи програми є хиткою.

2.3 Застосування вагових коефіцієнтів

Цей метод є модернізацією алгоритму застосування центрального дескриптору тим, що у кожного біта центрального дескриптора додатково з'являється свій ваговий коефіцієнт. Дана модернізація дозволяє зменшити вплив нечітких значень центрального дескриптора. Під нечітким значенням мається на увазі один з 512 бітів дескриптора, для якого при побудові нулів та одиниць було майже порівну. Таким чином вказуємо, що при класифікації дескриптора к центру, полягати на даний біт не варто. Також робимо більш значущим біт (значення), який зустрічається частіше. Якщо якийсь з значень дескриптора при побудові центрального дескриптора зустрінеється у 100% випадків, то було би логічно, якщо би цей біт впливав на класифікацію більш значуще ніж інші.

Для кожного значення вектора m_i у (2.10) будуємо ваговий коефіцієнт W_i .

$$W_i(\mathbf{b}) = \begin{cases} \frac{\sum_{d=1}^{s(i)} x_d(\mathbf{b})}{n}, & \sum_{d=1}^{s(i)} x_d(\mathbf{b}) \geq s(i) / 2, \\ 1 - \frac{\sum_{d=1}^{s(i)} x_d(\mathbf{b})}{n}, & \sum_{d=1}^{s(i)} x_d(\mathbf{b}) < s(i) / 2, \end{cases} \quad x_d \in Z^i, \quad b = 1, \dots, 512, \quad (2.12)$$

де $x_d(\mathbf{b})$ – біт з номером b для дескриптора з номером d в описі еталона, n - кількість бітів дескриптора (512).

Таким чином для кожного біту дескриптора отримуємо значення коефіцієнту від 0.5 до 1 в незалежності від значення біту, 0 чи 1. Важливо тільки чи співпадає біт чи ні. А цей ваговий коефіцієнт якраз вказує на вірогідність появи біту.

Етап аналізу відрізняється більш суттєво, аналізуємо елементи $x \in Z$ загального вмісту структурних описів бази еталонів (навчальна вибірка) шляхом віднесення їх до відповідного центру (класу) з використанням конкурентного способу (2.14).

$$v = \arg \min_{j=1, \dots, J} \rho(\mathbf{b}, m_j), \quad (2.13)$$

Тут у якості $\rho(x, m_i, W_i)$ застосуємо алгоритм оцінки за вагою:

$$\rho(x, m_i, W_i) = \sum_{b=1}^{512} \begin{cases} W_i m_i, & x(b) = m_i(b) \\ -W_i m_i, & x(b) \neq m_i(b) \end{cases}, \quad (2.14)$$

що визначає кількість розбіжних бітів для двох двійкових послідовностей однакової довжини (детекторів ОТ).

У результаті оброблення для кожного еталона отримуємо його подання $H[Z^i] = (h_1^i, \dots, h_J^i)$, де h_a^i – ціле число, яке відповідає розподілу елементів множини Z^i за класами еталонів.

Різновидністю розглянутого підходу є метод «відмови від слабких». Цей метод є близьким до алгоритму з ваговими коефіцієнтами. Але замість діапазону значень від 0.5 до 1 вагові коефіцієнти приймають дискретні значення 0 або 1.

Для кожного значення вектора m_i (2.13) обчислюємо ваговий коефіцієнт W_i .

Аналізуємо елементи $x \in Z$ загального вмісту структурних описів бази еталонів (навчальна вибірка) шляхом віднесення їх до відповідного центру (класу) з використанням конкурентного способу (2.14).

У якості $\rho(x, m_i, W_i)$ застосовуємо алгоритм оцінки за вагою (2.15).

$$W_i(b) = \begin{cases} 1, & \sum_{d=1}^{s(i)} x_d(b) \geq (s(i) / 2), & \frac{\sum_{d=1}^{s(i)} x_d(b)}{n} > 0.5 + r \\ 1, & \sum_{d=1}^{s(i)} x_d(b) < (s(i) / 2), & 1 - \frac{\sum_{d=1}^{s(i)} x_d(b)}{n} > 0.5 + r \\ 0, & \sum_{d=1}^{s(i)} x_d(b) \geq (s(i) / 2), & \frac{\sum_{d=1}^{s(i)} x_d(b)}{n} < 0.5 + r \\ 0, & \sum_{d=1}^{s(i)} x_d(b) < (s(i) / 2), & 1 - \frac{\sum_{d=1}^{s(i)} x_d(b)}{n} < 0.5 + r \end{cases} \quad (2.15)$$

де r - це радіус слабких. Будемо брати значення 0.05.

Для проаналізованих алгоритмів під час класифікації брали до уваги два випадки. Перший випадок, коли біт дескриптора однаковий з відповідним бітом вхідного дескриптора. Другий випадок – коли вони різні. Тепер беремо до уваги тільки ті випадки, коли біти не однакові, аналогічно підрахунку відстані Хемінга. Для вагових коефіцієнтів та відмови для слабких замість формули (2.13) тепер будемо використовувати (2.1). А формула підрахунку вагових коефіцієнтів (2.14) буде приймати наступний вигляд:

$$\rho(x, m_i, W_i) = \sum_{b=1}^{512} \begin{cases} 0, & x(b) = m_i(b) \\ W_i m, & x(b) \neq m_i(b) \end{cases}, \quad (2.16)$$

Способи отримання самих вагових коефіцієнтів не змінюються. Дана модифікація частково спрощує підрахунок коефіцієнтів й повністю імітує відстань Хемінга.

3 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

3.1 Обґрунтування вибору середовища програмної реалізації

У рамках роботи розроблено алгоритми бінарної обробки даних для структурної класифікації зображень. Для реалізації обрана бібліотека з відкритим доступом Open Source Computer Vision Library (OpenCV) [4]. Це обумовлено можливістю безкоштовного використання бібліотеки та можливістю вільного продажу програмного забезпечення, яке написано з її використанням, а також високою швидкістю, що обумовлюється високопродуктивною мовою C++, й великим обсягом готових рішень, зв'язаних з обробленням зображень.

OpenCV - це бібліотека програмного забезпечення з відкритим кодом для комп'ютера та машинного навчання. OpenCV була побудована, щоб забезпечити загальну інфраструктуру для комп'ютерних програм зору та прискорити використання сприйняття машин у комерційних продуктах.

Бібліотека має понад 2500 оптимізованих алгоритмів, що включає в себе повний комплект як класичного, так і сучасного комп'ютерного бачення та алгоритмів машинного навчання. Ці алгоритми можуть бути використані для виявлення та розпізнавання облич; визначення об'єктів; класифікації людських дій у відео; відстеження рухомих об'єктів; витягування 3D-моделей об'єктів; створення 3D-точкових хмар із стерео камер; зшивання зображень разом для отримання високої роздільної здатності зображення цілісної сцени; пошуку схожих зображення з бази даних зображень; видалення червоних очей з зображень, створених за допомогою спалаху; стеження за рухами очей; визначення декорації та встановлення маркеру для накладання його на додану реальність, тощо. Бібліотека широко використовується в компаніях, дослідницьких групах та урядових структурах [4].

OpenCV має інтерфейси C++, Python, Java та MATLAB і підтримує Windows, Linux, Android та Mac OS. OpenCV спрямовується в основному на додатки в режимі реального часу та використовує інструкції MMX та SSE, коли це можливо. В даний час активно розвиваються повнофункціональні інтерфейси CUDA та OpenCL. Існує понад 500 алгоритмів і приблизно в 10 разів більше функцій, які складають або підтримують ці алгоритми. OpenCV написано спочатку на C++ і має шаблонний інтерфейс, який працює без проблем з контейнерами STL [4].

Причини вибору мови програмування C++ [23]:

- підтримуються різні стилі і технології програмування, включаючи традиційне директивне програмування, ООП, узагальнене програмування, метапрограмування (шаблони, макроси);

- передбачуване виконання програм є важливою перевагою для побудови систем реального часу. Весь код неявно генерується компілятором для реалізації мовних можливостей (наприклад, при перетворенні змінної до іншого типу), визначений в стандарті. Також строго визначені місця програми, в яких цей код виконується. Це дає можливість заміряти або розраховувати час реакції програми на зовнішні події;
- висока сумісність з мовою C, що дозволяє використовувати весь існуючий C-код (код на C може бути з мінімальними переробками компільовано компілятором C++; бібліотеки, написані на C, зазвичай можуть бути викликані з C++ безпосередньо без будь-яких додаткових витрат, в тому числі і на рівні функцій зворотного виклику, дозволяючи бібліотекам, написаним на C, викликати код, написаний на C++);
- мова підтримує поняття фізичної (const) і логічної (mutable) константності. Це робить програму надійніше, так як дозволяє компілятору, наприклад, діагностувати помилкові спроби зміни значення змінної. Оголошення константності дає програмісту, що читає текст програми, додаткове уявлення про правильне використання класів і функцій, а також може бути підказкою для оптимізації. Перевантаження функцій-членів за ознакою константності дозволяє визначати зсередини об'єкта мету виклику методу (константний для читання, неконстантний для зміни). Оголошення mutable дозволяє зберігати логічну константність при використанні кешей і ледачих обчислень;
- використовуючи шаблони, можливо створювати узагальнені контейнери і алгоритми для різних типів даних, а також спеціалізувати і обчислювати на етапі компіляції;
- можливість імітації розширення мови для підтримки парадигм, які не підтримуються компіляторами безпосередньо. Наприклад, бібліотека Boost.Bind дозволяє пов'язувати аргументи функцій;
- можливість створення вбудованих предметно-орієнтованих мов програмування. Такий підхід використовує, наприклад бібліотека Boost.Spirit, що дозволяє задавати EBNF-граматику парсерів прямо в коді C++;
- ефективність. Мова спроектована так, щоб дати програмісту максимальний контроль над усіма аспектами структури і порядку виконання програми. Жодна з мовних можливостей, що призводить до додаткових накладних витрат, не є обов'язковою для використання - при необхідності мова дозволяє забезпечити максимальну ефективність програми;
- використовуючи шаблони і множинне спадкування можна імітувати класи-домішки і комбінаторну параметризацію бібліотек. Такий підхід застосований в бібліотеці Loki, клас SmartPrt якої дозволяє, керуючи всього декількома параметрами часу компіляції, згенерувати близько 300 видів «розумних покажчиків» для управління ресурсами;
- є можливість роботи на низькому рівні з пам'яттю, адресами.

3.2 Особливості програмної реалізації

Для роботи програми побудовано декілька баз зображень. Для роботи алгоритму потрібна вибірка з еталонними зображеннями, на основі яких будуть побудовані центри дескрипторів. Друга база зображень - це зображення, які порівнюються з центрами кластеризації еталонних зображень, тобто зображення, що будуть класифіковані до одного з класів (зображень), що знаходяться у вибірці еталонних зображень. За допомогою бібліотеки OpenCV отримуємо дескриптори BRISK для кожного зображення (як для еталонів так й для вхідних зображень). Але перед тим як побудувати вектори дескрипторів ОТ за допомогою алгоритму BRISK, треба позбавити зображення від шумів та провести перетворення зображення у чорно біле при необхідності. Як показано на рисунку 3.1, це робиться для покращення виділення коректних ОТ на зображенні. Для зображень з камер різної якості та при різному освітленні первинна обробка зображень скоріше за все буде різною.



Рисунок 3.1 – Початкова обробка зображення

Наступним кроком після побудови дескрипторів є вирівнювання кількості ОТ між зображеннями. Тобто, щоб усіх зображень була однакова кількість ОТ, як у зображень вхідних так й у еталонів. Для цього було побудовано функцію `sortDescriptorsAndKP`, яка редагує кількість ОТ для кожного зображення.

У якості параметрів функція отримує вектор ОТ та дескрипторів й число особливих точок, що повинно залишитися. Функція позбавляється від найбільш близьких ОТ на зображенні у піксельній відстані. Дозволяє покрити більшу площу на зображенні покрити ОТ, але зменшити їх нагромадження в одній точці. Її програмна реалізація приведена на рисунку 3.2.

```

void sortDescriptorsAndKP(vector<KeyPoint> &kp, Mat &descriptor, int MaxcolElem) {
    int *      Masmindist      = new int[kp.size()-1];
    Mat        descriptorSizeN = descriptor.clone();
    vector<KeyPoint> newKP;
    int        k                = 0;
    int delElem = descriptorSizeN.rows - MaxcolElem;
    descriptorSizeN.pop_back(delElem);
    for (int i = 0; i < kp.size() - 1; ++i) {
        Masmindist[i] = calcHammingDistance(kp[i].pt.x + kp[i].pt.y, kp[i + 1].pt.x
+ kp[i + 1].pt.y);
    }
    for (int ss = 9; ss >= 0; --ss) {
        for (int i = 0; i < kp.size() - 1; ++i) {
            if (ss == Masmindist[i]) {
                descriptorSizeN.row(k) = descriptor.row(i);
                newKP.push_back(kp[i]);
                ++k;
                if (k == MaxcolElem) { goto a; }
            }
        }
    }
a:
    kp        = newKP;
    descriptor = descriptorSizeN; }
}

```

Рисунок 3.2 – Програмна реалізація функції sortDescriptorsAndKP

Це робиться заради кращого порівняння наших центрів еталонних зображень й кластерів, що розпізнаються, бо після побудови векторів ОТ з однаковими параметрами для алгоритму BRISK для кожного зображення знаходиться різна кількість ОТ. Для зображення у якого буде 215 ОТ, центр буде знайдений більш чітко, але коли у інших зображень буде по 110 ОТ, їх порівняння буде більш чітке між собою ніж з тим же зображенням, що має більшу кількість ОТ, через нерівні умови побудов кластерів, тому жертвуємо точністю наших центрів кластерів, але тим самим покращуємо сам алгоритм розпізнавання відносно еталонних зображень. На рисунку 3.3 показано, з правої частини - зображення з 250 ОТ до використання алгоритму й з лівої - 105 після.

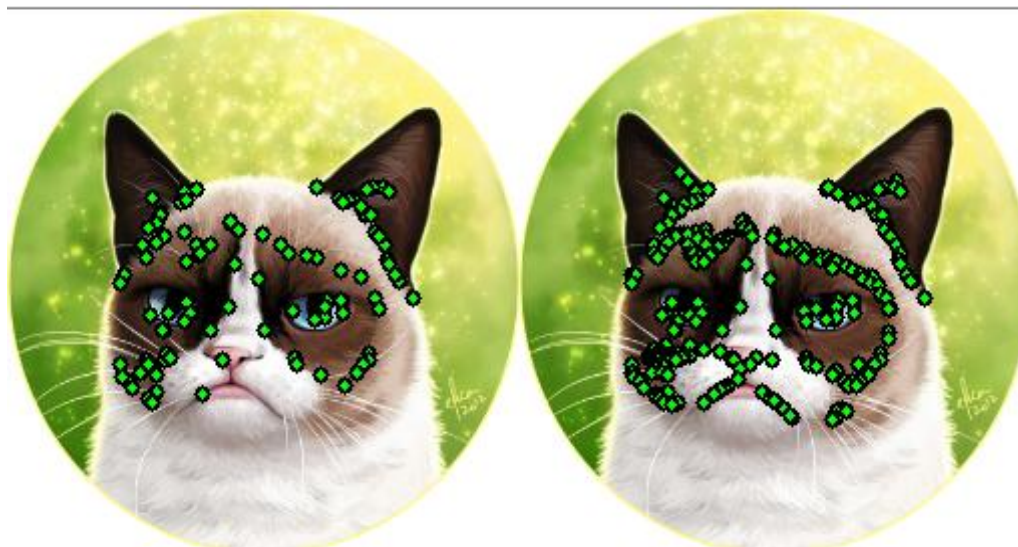


Рисунок 3.3 – Демонстрація роботи функції sortDescriptorsAndKP

Наступним кроком будеться центральний дескриптор для кожного з еталонних зображень. Для цього було реалізовано декілька алгоритмів: метод класифікації за допомогою центрального дескриптора, метод побудови центрального дескриптора з вагою та метод відмови від слабких.

Оскільки дескриптори мають бінарний вид, то для всіх алгоритмів порівнюємо сумарну кількість нулів та одиниць для кожного з 512 бітів та обираємо той, що найбільш зустрічається для усього вектора в цілому, отриманий дескриптор є нашим центром. На рисунку 3.4 представлено програмну реалізацію методу класифікації за допомогою центрального дескриптора.

```
int findCenterForKlaster(Mat descriptorsImage, int n){
    int AVG = 0;
    int mas[8] = { 0,0,0,0,0,0,0,0 };
    bitset< 8 > a;
    for (int i = 0; i < descriptorsImage.rows; ++i) {
        a = bitset< 8 >(descriptorsImage.at<uchar>(i, n));
        for (int j = 0; j < 8; ++j)
            mas[j] += a[j];
    }
    for (int j = 0; j < 8; ++j) {
        if (descriptorsImage.rows / 2 > mas[j])
            a[j] = 0;
        else
            a[j] = 1;
    }
    AVG = a.to_ullong();
    return (int)AVG;
}
```

Рисунок 3.4 – Програмна реалізація функції findCenterForKlaster

На рисунку 3.5 показано реалізацію знаходження центру зображення з вагою.

```
int findCenterForKlaster(Mat descriptorsImage, int n, double
*mas_frequency_of_occurrence)
{
    double mas[8] = { 0,0,0,0,0,0,0,0 };
    bitset< 8 > a;

    for (int i = 0; i < descriptorsImage.rows; ++i) {
        a = bitset< 8 >(descriptorsImage.at<uchar>(i, n));
        for (int j = 0; j < 8; ++j)
            mas[j] += a[j];
    }
    for (int j = 0; j < 8; ++j) {
        if (descriptorsImage.rows / 2 > mas[j]) {
            a[j] = 0;
            mas_frequency_of_occurrence[n * 8 + j] = 1.0 - mas[j] / descriptorsImage.rows;
        }
        else {
            a[j] = 1;
            mas_frequency_of_occurrence[n * 8 + j] = mas[j] /
(double)descriptorsImage.rows;
        }
    }
    return (int)a.to_ullong();
}
```

Рисунок 3.5 – Програмна реалізація функції findCenterForKlaster з вагою

Для цього передаємо до нашої функції вектор, в якому будемо зберігати вагу для кожного біту побудованого центра. Сам центр будується так само як у випадку з попереднім алгоритмом. Таким чином, отримавши біт переможець 1 чи 0, отримуємо під вагою його співвідношення до біту переможеного. Тобто, якщо маємо 100 ОТ й перший біт був одиниця у 70-ти випадках, а нулем у 30-ти випадках, то перший біт нашого центру буде '1', а його вага буде '0.7'. Перейдемо до другого біту, він тепер у 10-ти випадках дорівнював одиниці, а у 90-то випадках був нулем. Тобто другий біт нашого центру буде '0', а його вага буде дорівнювати '0.9'.

На рисунку 3.6 показано реалізацію знаходження центру зображення за допомогою алгоритму відмови від слабких. Він схожий на алгоритм з ваговими коефіцієнтами, але в даному випадку під вагою отримуємо значення 1 чи 0. А також маємо поріг слабкості, що дорівнює '0.05'.

```
int findCenterForKlasterWithoutWeek(Mat descriptorsImage, int n, double
*mas_frequency_of_occurrence)
{
    double mas[8] = { 0,0,0,0,0,0,0,0 };
    bitset< 8 > a;

    for (int i = 0; i < descriptorsImage.rows; ++i) {
        a = bitset< 8 >(descriptorsImage.at<uchar>(i, n));
        for (int j = 0; j < 8; ++j)
            mas[j] += a[j];
    }
    for (int j = 0; j < 8; ++j) {
        if (descriptorsImage.rows / 2 > mas[j]) {
            a[j] = 0;
            mas_frequency_of_occurrence[n * 8 + j] = floor((1.0 - mas[j] /
descriptorsImage.rows)+0.45);
        }
        else {
            a[j] = 1;
            mas_frequency_of_occurrence[n * 8 + j] = floor((mas[j] /
(double)descriptorsImage.rows)+0.45);
        }
    }
    return (int)a.to_ulong();
}
```

Рисунок 3.6 – Програмна реалізація функції findCenterForKlasterWithoutWeek

Приведемо приклад. Знову маємо 100 ОТ. Перший біт був нулем у 58 випадках, а один у 42 випадках. Таким чином перший біт буде '0', а його вага буде '1'. Для другого біту 0 зустрівся у 46 випадках, а один у 54. Таким чином другий біт буде '1', а його вага '0' через те, що його переважна кількість менша, ніж поріг слабкості. Цей алгоритм можна використовувати для оцінки якості збудованого центру на даній вибірці ОТ, чим більше нулів, тим більш не якісний центр ми отримали. Його слабкість - це центри, що є не якісними. В цьому випадку навіть при переможній кількості ОТ, що приєдналися до нього, його вага може бути меншою ніж у іншого центра, у якого

вага всіх бітів одиниця, про те тим самим ми звільняємося від ОТ, що приєдналися би через не якісний біт. Тому треба бути обережним при підборі порогу слабкості.

Отримавши центри, можемо надіслати до програми вхідні зображення, які будуть віднесені до одного з зображень, що зараз знаходиться в виборці еталонів.

Першим кроком так само як й для еталонних зображень, будемо вектори ОТ. Після цього кожен дескриптор порівнюється з усіма центрами за допомогою розрахунку відстані по Хемінгу й відноситься до найближчого за відстанню для методу класифікації за допомогою центрального дескриптора. Для алгоритму з ваговими коефіцієнтами перед початком порівнювання дескриптора кожен центр має значення ваги 0. Після цього починаємо порівнювати вхідний дескриптор з кожним центром по черзі. Уявимо, що в нас 3 центри. Спочатку порівнюємо з 1-шого по 512 біт нашого першого центра з вхідним дескриптором. Якщо біт співпадає, то збільшуємо значення на відповідну вагу, якщо не співпав, то зменшуємо. Таким чином можемо отримати такі значення: 7.48, 130.67, 15.8 для першого, другого та третього дескрипторів. Переможе той центр, у якого значення найбільше. Для відмови від слабких даних етап такий самий як й у вагових коефіцієнтів.

Коли усі дескриптори віднесені до своїх центрів, обирається центр, до якого віднесено найбільша кількість ОТ. Даний евристичний метод не змінює центр зображення, а проводить порівняння дескрипторів з центром, що був збудований у самому початку, що дає перевагу в швидкості над тим самим Кохоненом або іншими алгоритмами, у яких центр кластеризації постійно зміщується. На цьому етапі розпізнавання завершується.

Також було програмно реалізовано додаткові функції, такі як побудова графіків та діаграм дескрипторів, конвекторів даних як внутрішнього використання у алгоритмах, так й для виводу інформації для користувача, що дозволяє оцінити роботу алгоритмів й порівняти їх (на рисунку 3.16 можна побачити один з додаткових класів MySort).

На рисунку 3.7 приведено діаграму послідовності роботи програми.

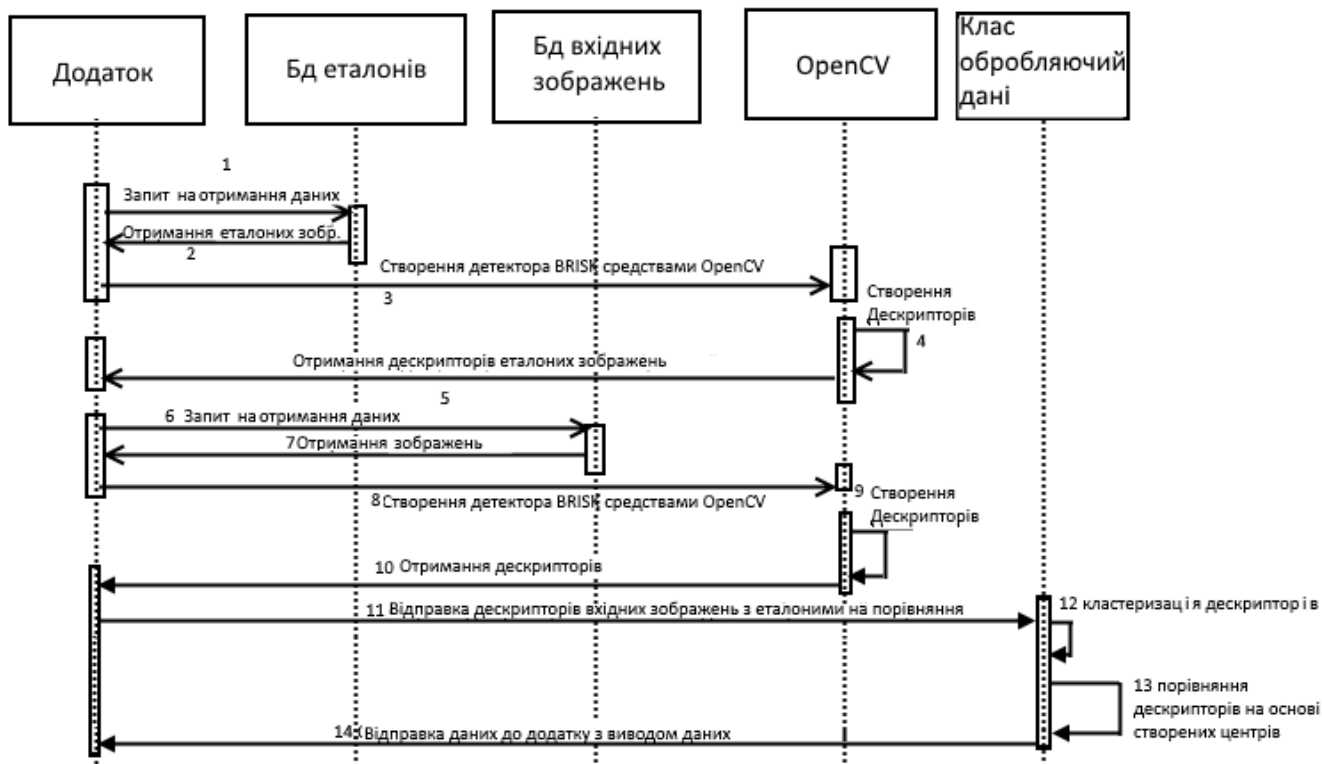


Рисунок 3.7 – Діаграма послідовності

3.3 Інструкція користувача

Оскільки програма представляє собою алгоритмічну реалізацію вирішення задач класифікації, то будемо розглядати її як бібліотеку. Для її використання спочатку до проекту треба підключити бібліотеку OpenCV, додатком до якої є дана робота. Виконуємо наступну послідовність дій:

Крок 1. Завантаження та установка OpenCV.

Насамперед необхідно завантажити інсталяційний файл OpenCV (доступний на офіційному сайті проекту: <http://opencv.org/>). Версія для Windows (рис.3.8).

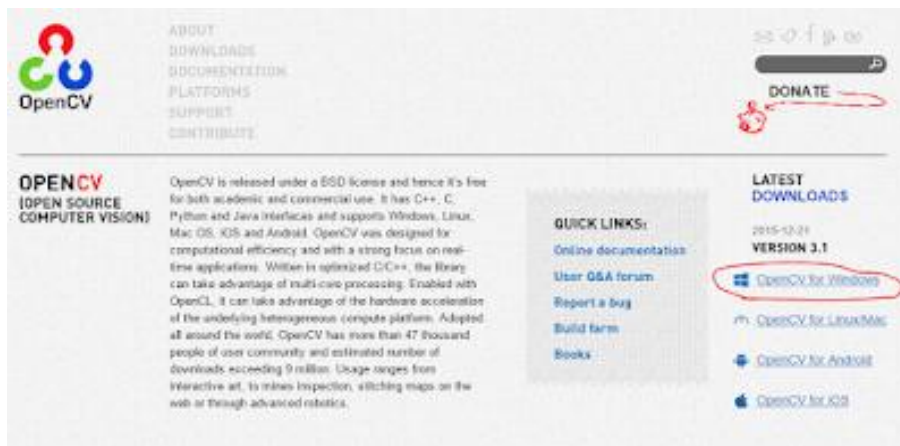


Рисунок 3.8 – Сторінка офіційного сайту OpenCV

Після того, як файл завантажиться, запускаємо його і вибираємо каталог для установки. Виберемо наприклад диск D: \ (створювати папку не треба: при розпакуванні в обрану директорію додасться папка "opencv").

Крок 2. Налаштування змінних середовища (рис.3.9).

Для комфортного використання рекомендується налаштувати змінні середовища.

Відкриваємо: "Панель управління" > "Система" > "Додаткові параметри системи" > Додатково > "Змінні середовища ..."

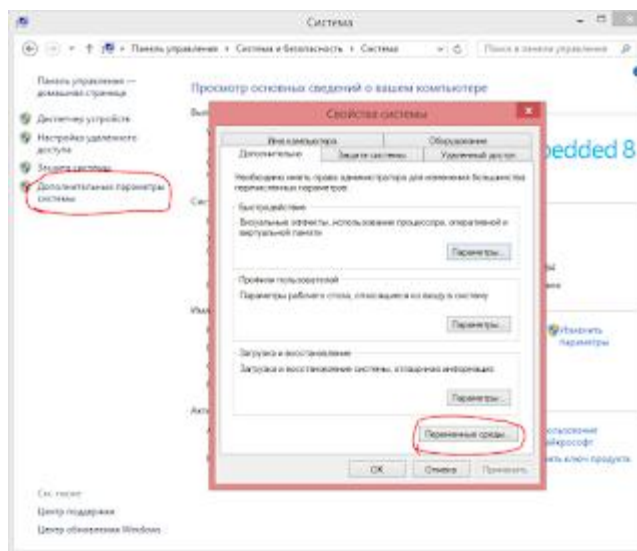


Рисунок 3.9 – Налаштування системи

Створимо таку системну змінну:

- Змінна: OPENCV_DIR.
- Значення: [Місце установки opencv] \ opencv \ build \ x64 \ vc14 (В нашому випадку: D: \ opencv \ build \ x64 \ vc14).

Далі виправляємо змінну Path: Додаємо в кінець її значення наступний рядок:

;% OPENCV_DIR% \ bin.

Після додавання змінної середовища необхідно перезавантажити комп'ютер;

Крок 3. Налаштування проекту Visual Studio.

Якщо під час попередніх кроків Visual Studio була запущена, її необхідно перезапустити.

У проекті, де необхідно використовувати OpenCV, відкриваємо властивості проекту (Проект > Властивості або Alt + F7)

Перед внесенням змін виберіть конфігурацію "Усі зміни", а платформу "x64".

У закладці C / C++ вказуємо додаткові каталоги включення файл: \$ (OPENCV_DIR) \ .. \ .. \ include (рис.3.10, 3.11).

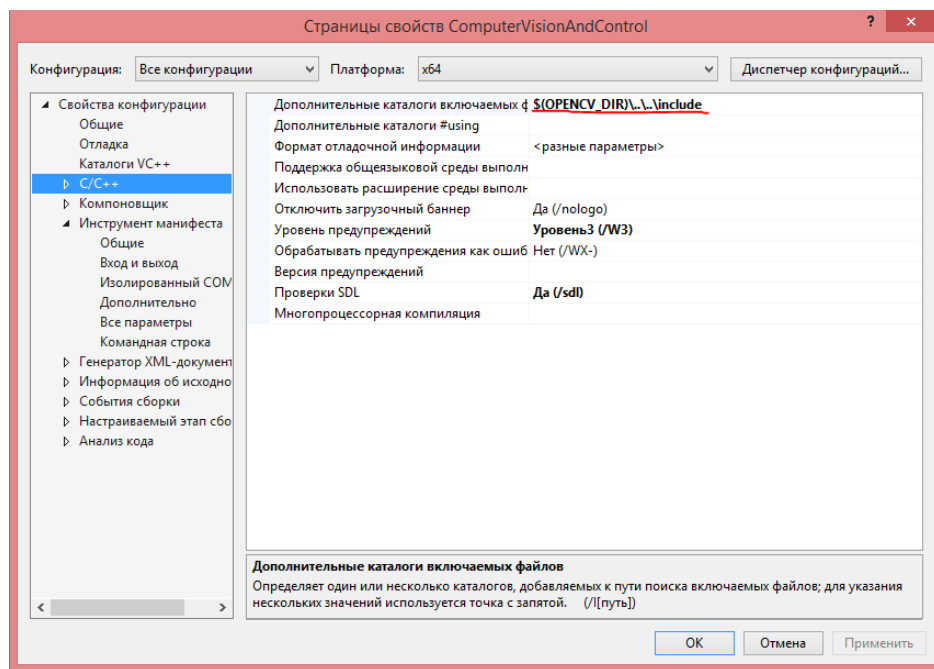


Рисунок 3.10 – Конфігурація проекту

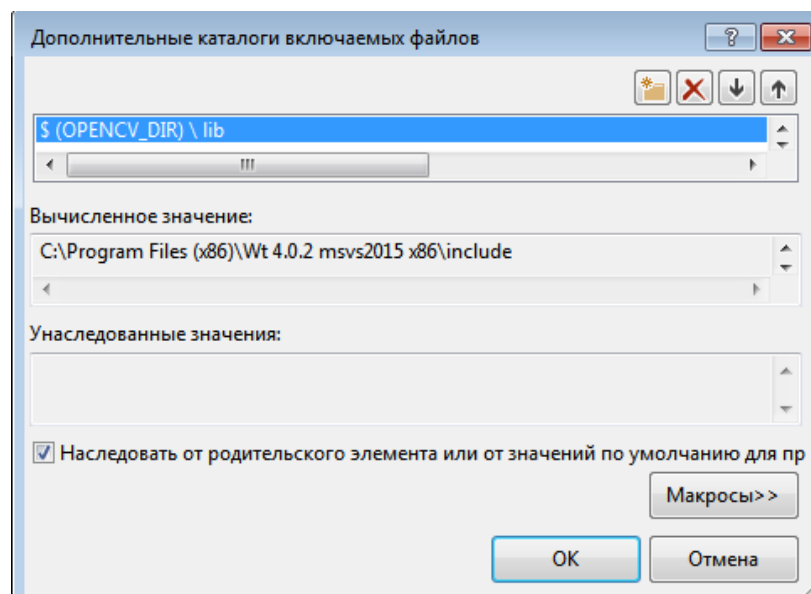


Рисунок 3.11 – Додаткові каталоги включень

У закладці Лінкер > Загальні вказуємо додаткові каталоги бібліотек: \$(OPENCV_DIR) \ lib (рис.3.12, 3.13).

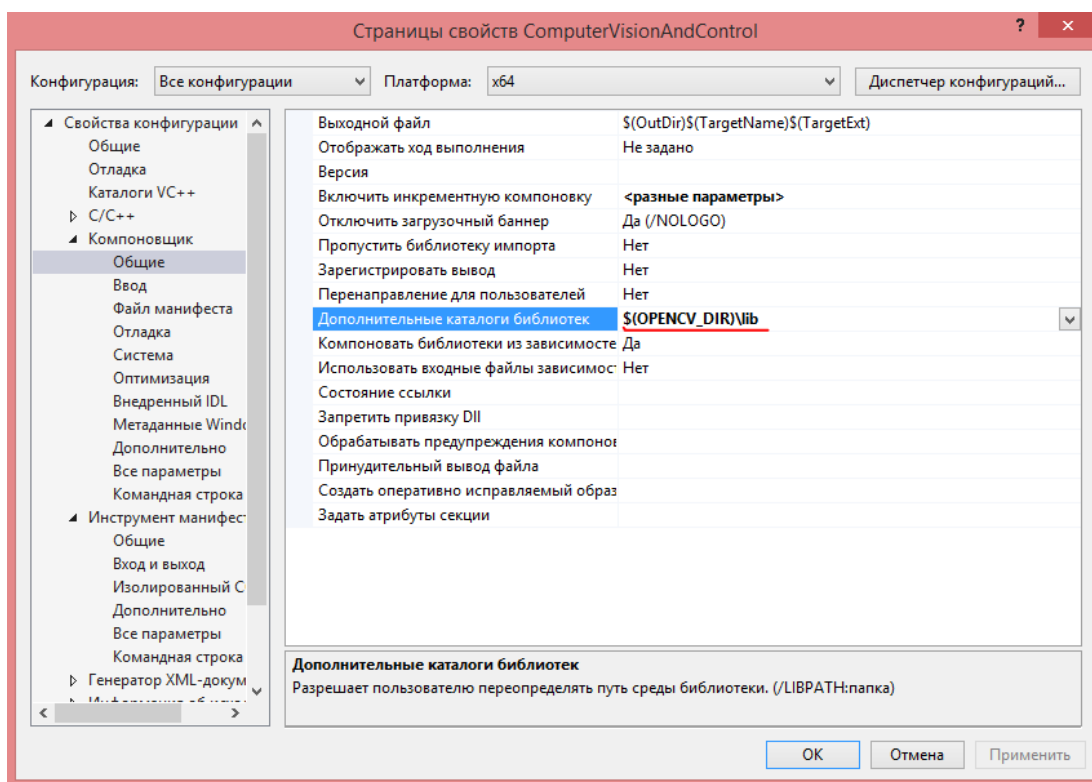


Рисунок 3.12 – Конфігурація проекту (Лінкер)

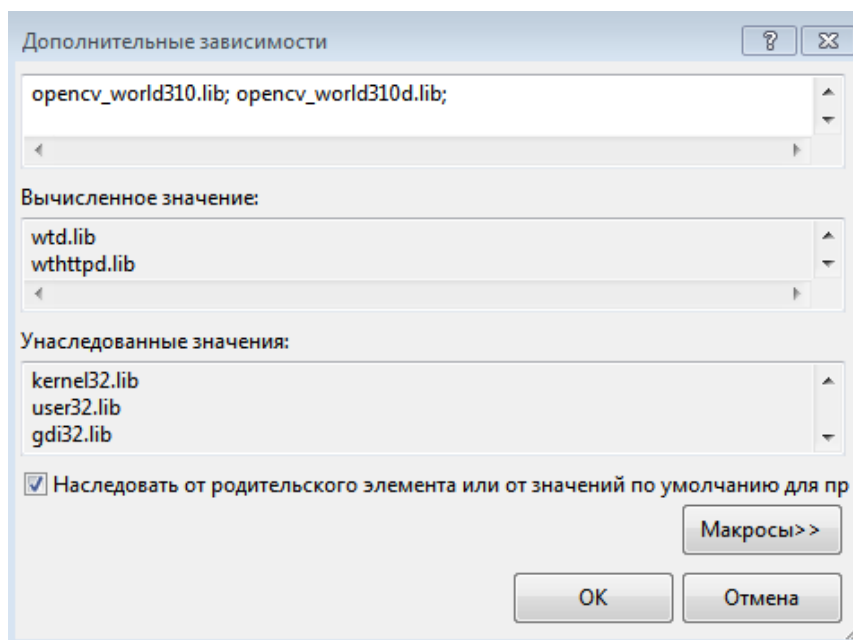


Рисунок 3.13 – Додаткові каталоги бібліотек

У закладці Лінкер > Введення вказуємо додаткові залежності:

opencv_world310.lib; opencv_world310d.lib (рис.3.14); та копіюємо їх у головну папку проекту, а також до папки з виконуючим файлом.

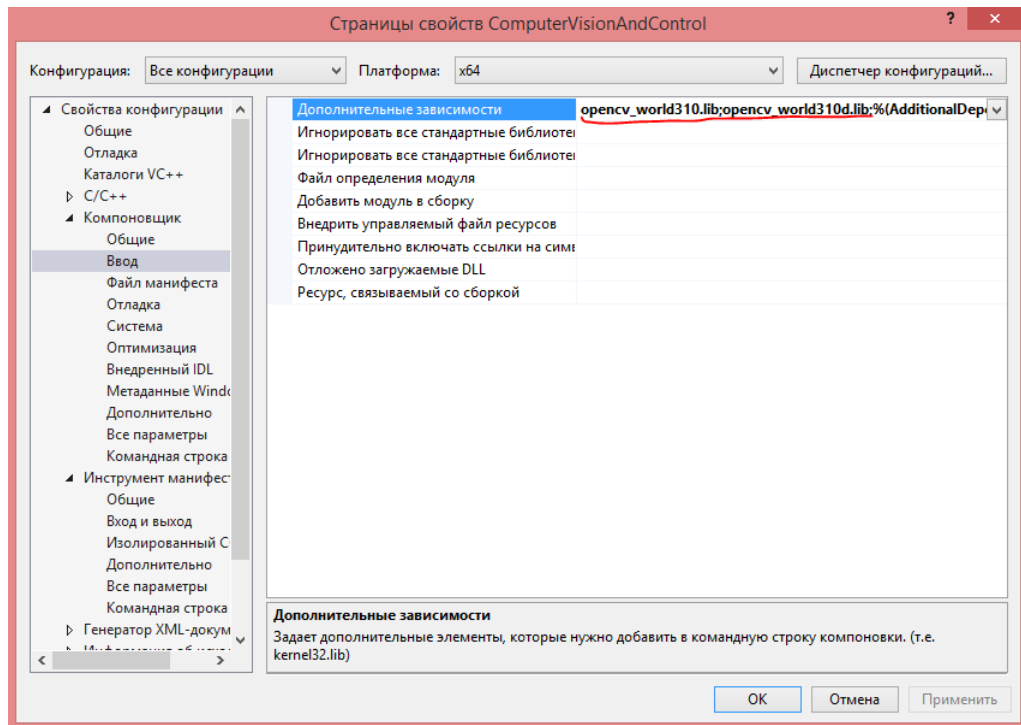


Рисунок 3.14 – Конфігурація проекту. Лінкер

Тепер можливо приступати до підключення файлу. Його можливо узяти як відкритий код або як бібліотеку. Скопіюємо відкритий код до проекту. Три файли `kasterization.h`, `Work_With_Brisk.h` та `myMatche.h`. Як користувач, маємо два класи `Create_Central_descriptors` та `Image_definition`. Перший реалізує алгоритми створення центрів. Для цього, п'ятим параметром передається 'H', 'W' або 'S' – метод побудови центрального дескриптора, вагових коефіцієнтів та відмови від слабких відповідно. А других клас відносить вхідне зображення до одного з побудованих нами центрів, що був переданий до програми попереднім кроком. В даній конфігурації програма розрахована на використання з дескрипторами BRISK. Можна шостим параметром передати `true`, або `false` для друку даних, з їх обробкою.

На рисунку 3.15 приведено приклад, в якому до бази було внесено зображення цифр.

```

Create_Central_descriptors centers(detectorBRISK, img, number_of_image/2, 51, 'H');
Image_definition::definition(detectorBRISK, centers, img[i], 51, 'H');

```

Рисунок 3.15 – Побудова дескрипторів BRISK

Метод `Klast::distributionDescriptors` проводить розподілення дескрипторів OT по збудованим нами центрам й повертає бінарний результат (рис.3.16). Також маємо внутрішній клас

MySort, завдяки якому проводимо обробку даних й можемо виводити наступні данні, що показані на рисунку 3.17.

```

for (int i = 0; i < n2; ++i) {
    MySort          mySort;
    vector<vector<vector<bitset< 8 > > > > klaster =
Klast::Convert_to_vector_bits(klasterDescripIm, n);
    mySort.matr = Klast::distributionDescriptors(klaster,
Klast::Convert_to_vector_bits(&descriptorsImage2[i]));
    mySort.sort();
    mySort.show();
    std::cout << "This is number: " << mySort.maxId() << std::endl;
    imshow("matches", imgMatch2[i]);
    while (true) {
        if (cv::waitKey(10) == 27)
            break;
    }
}

```

Рисунок 3.16 – Обробка дескрипторів

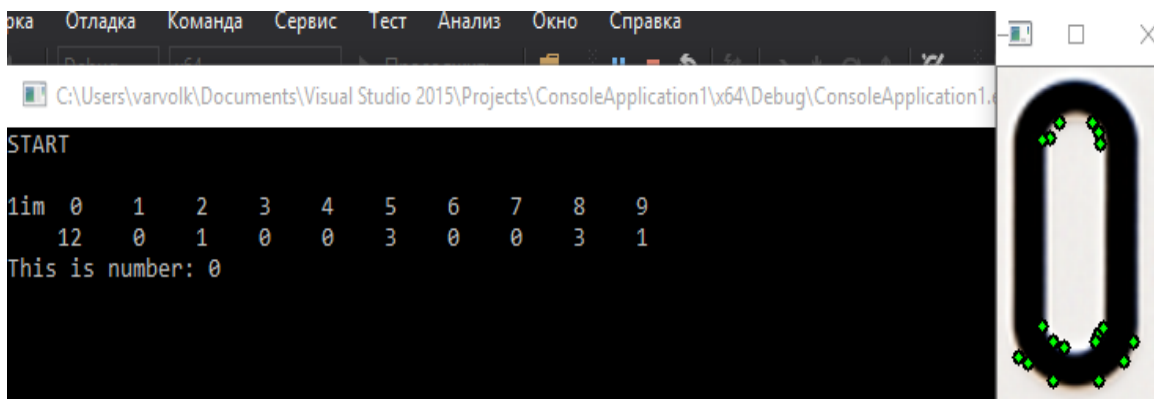


Рисунок 3.17 – Приклад роботи програми

3.4 Аналіз результатів дослідження

У середовищі C++ з використанням бібліотеки програм OpenCV [4] розроблено програмну модель для класифікації зображень на підставі описів, сформованих детектором BRISK.

Проведено ряд дослідів. Для першого, у якості бази еталонів було взято 15 зображень (рис. 3.18). Усі зображення мають однаковий розмір 256x256 та обмежені кількістю особливих точок на момент передачі дескрипторів до алгоритмів, кожне зображення має 51 дескриптор. Через обчислення центрів класів у відповідності до логіки (3) сформовано вектори $h[Z^i]$ для кожного із еталонів. Приклади центрів показано на рисунку 3.19.

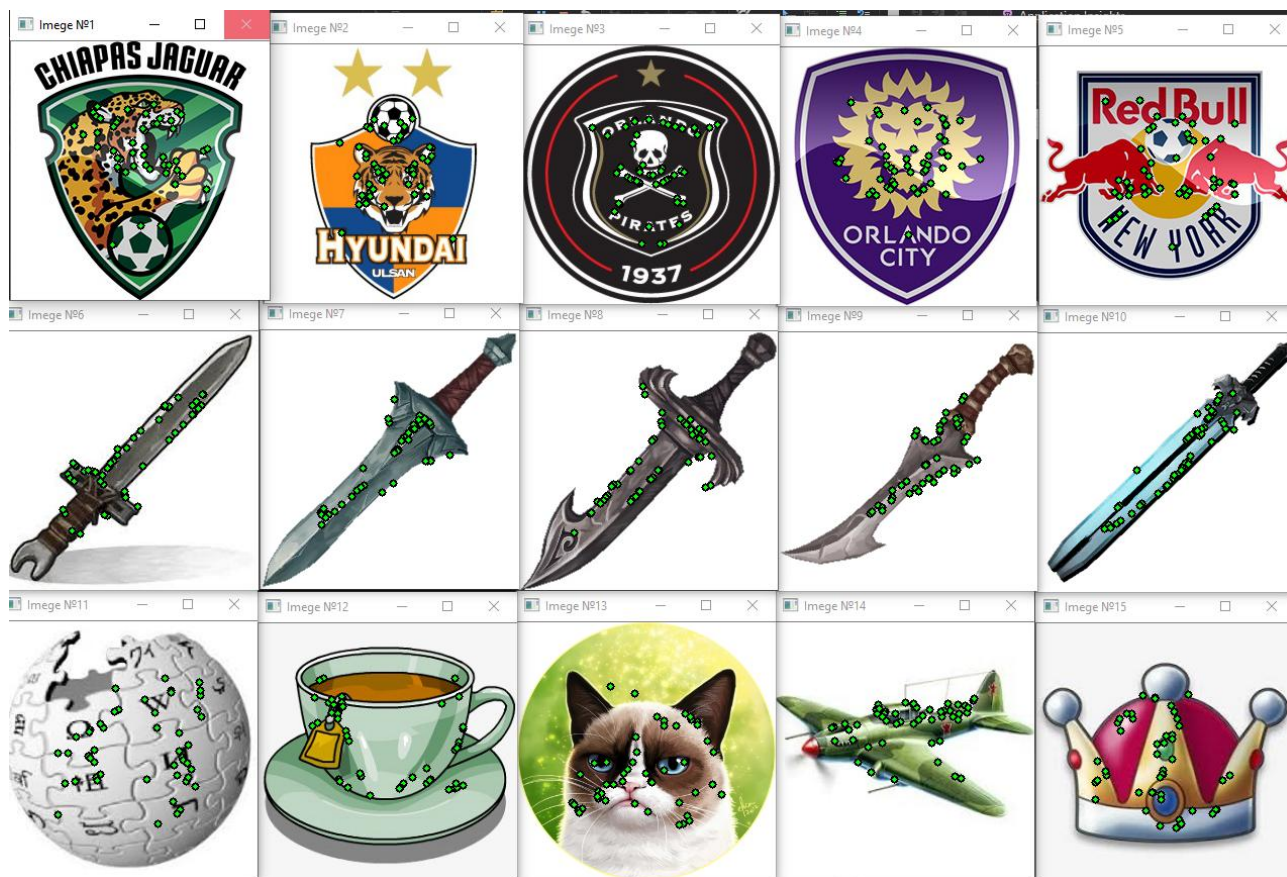


Рисунок 3.18 – Тестова вибірка зображень №1



Рисунок 3.19 – Бітове подання (2.3) для центрів зображень рисунку 3.18 для 7-мого, та 8-мого зображень відповідно

В першому досліді порівнюється швидкодія роботи алгоритмів та їх працездатність. Швидкодію будемо вимірювати за допомогою Runtime у режимі відлагодження програми, тому час будемо вимірювати умовними одиницями (УО). А результативність відображається кількістю правильно віднесених ОТ зображення X до зображення X.

На рисунку 3.20 по діагоналі, що переважно темно зеленого кольору, можемо побачити скільки дескрипторів зображення 1-15 віднесли з 51-го до свого же зображення. На обробку 15-ти зображень було затрачено 6000 УО. Найбільш ефективно було розпізнане зображення 3 та 4, вони мають більше 40 ОТ віднесених вірно. Проблеми були виявлені з зображенням 7. В цьому

випадку всього лиш 16 ОТ були віднесено вірно, а ще й 13 ОТ було віднесено до 1-шого центру. В цілому алгоритм відніс усі зображення вірно за доволі швидкий час.

	з.1	з.2	з.3	з.4	з.5	з.6	з.7	з.8	з.9	з.10	з.11	з.12	з.13	з.14	з.15
з.1	28	5	4	0	0	0	6	0	1	1	0	2	0	2	2
з.2	5	25	0	8	0	1	0	1	4	0	2	0	2	2	1
з.3	1	0	43	2	5	0	0	0	0	0	0	0	0	0	0
з.4	0	0	0	48	0	0	0	1	0	0	0	0	1	1	0
з.5	1	0	1	2	27	1	1	4	1	0	4	1	5	3	0
з.6	0	1	0	1	0	25	1	2	6	7	0	0	2	3	3
з.7	13	0	0	0	0	6	16	7	1	0	0	4	0	3	1
з.8	6	1	0	1	0	6	2	20	3	2	0	4	1	3	2
з.9	0	0	0	0	1	2	0	7	25	3	0	5	1	6	1
з.10	0	0	0	0	1	7	2	2	12	25	0	1	0	1	0
з.11	0	0	2	2	8	0	0	0	2	1	23	3	7	3	0
з.12	3	2	0	1	6	1	4	2	1	3	0	20	0	7	1
з.13	0	0	1	4	0	0	0	4	0	1	1	0	33	0	7
з.14	0	2	0	2	1	13	0	4	4	0	0	0	0	23	2
з.15	2	1	1	4	2	2	0	0	2	1	0	3	2	0	31

Рисунок 3.20 – Результат класифікації

Метод вагових коефіцієнтів показів не зовсім вдалі результати (рис. 3.21). Крім того, що було затрачено більше часу, це – 8200 УО, так й загальна працездатність класифікації зменшилась. Зображення 8 зовсім не було розпізнане, а для зображень 7, 10, 12, 13 результати погіршились доволі значуще. Проте для 1- 4 та 6 результати трохи покращилися, але доволі не значуще.

Метод відмови від слабких (рис. 3.22) для всіх зображень окрім 15-го відніс правильно більшу кількість ОТ у порівнянні з методом центрального дескриптора. Він є більш вдалою модифікацією, проте час, який він витратив, – 10062 УО є доволі значним у співвідношенні з часом методу центрального дескриптора.

	з.1	з.2	з.3	з.4	з.5	з.6	з.7	з.8	з.9	з.10	з.11	з.12	з.13	з.14	з.15
з.1	29	1	3	0	0	0	8	2	1	2	0	1	0	2	2
з.2	2	26	0	12	0	1	0	2	2	0	0	0	1	4	1
з.3	1	0	44	2	2	0	0	0	0	0	0	0	0	1	1
з.4	0	0	0	49	0	0	0	1	0	0	0	0	0	1	0
з.5	1	0	1	8	23	2	0	3	1	0	2	1	6	3	0
з.6	0	1	0	2	0	32	0	1	6	2	0	0	1	3	3
з.7	13	0	0	0	0	15	16	1	1	2	0	2	0	1	0
з.8	5	1	0	2	0	17	2	9	3	1	0	4	1	5	1
з.9	0	0	0	0	0	14	0	2	22	1	0	4	0	6	2
з.10	0	0	0	0	0	10	0	0	14	24	0	1	0	1	1
з.11	0	0	2	7	5	0	0	0	4	1	19	3	7	2	1
з.12	3	2	0	3	0	3	5	2	0	3	0	16	0	12	2
з.13	0	0	0	19	0	3	0	3	0	1	0	0	24	0	1
з.14	0	0	0	4	0	18	0	0	1	0	0	0	0	28	0
з.15	0	1	1	8	0	5	0	0	0	3	0	0	1	1	31

Рисунок 3.21 – Результат класифікації з ваговими коефіцієнтами

	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15
S1	29	3	3	0	0	0	8	2	1	1	0	1	0	2	1
S2	4	29	0	6	0	0	0	2	4	2	1	0	2	0	1
S3	1	0	44	2	4	0	0	0	0	0	0	0	0	0	0
S4	0	0	0	49	0	0	0	1	0	0	0	0	0	1	0
S5	1	0	2	5	27	1	0	5	0	0	3	1	4	2	0
S6	0	1	0	1	0	25	0	3	9	4	0	0	2	3	3
S7	12	0	0	0	0	12	19	1	1	1	0	3	0	2	0
S8	8	0	0	0	0	7	2	21	4	2	0	3	0	2	2
S9	0	0	0	0	0	5	0	6	29	3	0	4	0	2	2
S10	0	0	0	0	1	8	0	2	12	26	0	1	0	1	0
S11	0	0	3	2	7	0	0	0	2	1	25	3	5	2	1
S12	3	1	0	1	5	0	5	4	0	6	0	21	0	5	0
S13	0	0	0	4	0	0	0	7	0	1	0	0	34	0	5
S14	0	1	0	1	1	13	0	4	2	0	0	0	0	27	2
S15	3	1	1	4	1	2	0	0	2	4	0	2	2	0	29

Рисунок 3.22 – Відмова від слабких

Перейдемо до методів з дистанцією (рис. 3.23). На методі вагових коефіцієнтів дана модифікація впоралася за 8200 ОУ та показала куди більш цікавий результат. Значно покращилась працездатність, проте 7-ме зображення було класифіковано невірно. Хоча треба зазначити, що для частини зображень детектування стало кращим, ніж для методу побудови центрального дескриптора, на основі якого й проводимо модифікації. Так, для 1, 2, 5, 11 та 12-го зображень кількість ОТ була значно більша від МПЦД. Треба зазначити, що усі дані мали однакову первину обробку, що для методу вагових коефіцієнтів могло сказатися в погану сторону, але інакше були б порушені умови порівняння.

	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15
s1	35	0	5	0	0	0	5	0	0	0	0	4	0	0	2
s2	6	30	0	0	1	0	0	0	1	0	6	3	0	0	4
s3	1	0	44	0	5	0	0	0	0	0	0	0	0	0	1
s4	0	0	0	43	1	0	0	2	0	0	1	0	2	1	1
s5	1	0	2	1	38	0	1	2	0	0	5	0	1	0	0
s6	0	2	0	0	0	22	2	4	8	3	1	2	1	3	3
s7	16	0	0	0	0	2	15	6	1	0	0	4	0	6	1
s8	10	2	0	0	0	4	1	19	3	1	0	4	0	3	4
s9	0	0	0	0	2	2	0	7	25	3	0	5	0	6	1
s10	0	0	0	0	2	5	2	3	11	21	1	4	0	2	0
s11	0	0	2	0	10	0	0	0	1	0	32	3	2	1	0
s12	4	2	0	0	10	0	2	1	1	1	0	25	0	5	0
s13	1	0	2	0	3	0	0	3	0	0	3	0	31	0	8
s14	0	2	0	0	3	4	0	6	1	3	1	0	0	29	2
s15	3	3	1	2	2	0	0	1	1	0	0	3	0	2	33

Рисунок 3.23 – Метод вагових коефіцієнтів з дистанцією

Метод відмови від слабких з дистанцією (рис. 3.24) – дана модифікація впоралася за 8200 ОУ. У порівнянні з МПЦД працездатність однакова, проте гірша ніж для звичайного методу Відмови від слабких.

	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15
s1	37	1	3	0	0	0	6	0	0	0	0	2	0	1	1
s2	7	30	0	0	1	0	0	0	0	0	8	1	0	0	4
s3	4	0	41	0	5	0	0	0	0	0	0	0	0	0	1
s4	0	0	0	34	5	0	0	2	0	0	4	0	4	1	1
s5	2	0	3	0	38	0	2	1	0	0	5	0	0	0	0
s6	1	2	0	0	0	18	2	9	8	0	2	2	0	3	4
s7	16	0	0	0	0	0	19	8	1	0	0	1	0	5	1
s8	10	3	0	0	0	3	2	23	1	1	0	3	0	1	4
s9	0	1	0	0	1	1	4	9	24	1	2	2	0	5	1
s10	0	0	0	0	1	4	4	3	9	20	3	4	0	3	0
s11	0	0	2	0	9	0	0	0	1	0	36	3	0	0	0
s12	5	0	0	0	13	0	8	2	1	0	0	19	0	3	0
s13	2	0	2	0	9	0	0	3	0	0	6	0	19	0	10
s14	0	2	0	0	3	2	0	9	4	0	4	0	0	24	3
s15	9	3	0	1	3	0	0	0	1	0	0	5	0	2	27

Рисунок 3.24 – Відмова від слабких з дистанцією

Підведемо результати першого етапу тестування на вибірці з 15 випадкових зображень, побудувавши таблицю оцінювання (табл.3.1).

Таблиця 3.1 – Результати першого досліду

Метод	УО	Загальна кількість вірних ОТ	Кількість Зображень у яких 25+ ОТ вірних	Помилки	Місце AVG
МПЦД	6000	412	10	0	2
Вагових Коефіцієнтів (ВК)	8200	392	7	1	5
Відмова від сдабкого (ВВС)	8200 (8150)	434	12	0	1
ВК з дистанцією	8200 (8100)	442	11	1	3
ВВС з дистанцією	8200 (8150)	409	7	0	4

Через невеликий розмір вибірки знайти точну різницю в швидкості не вдалося, бо для одних й тих самих даних комп'ютер видає недетермінований час. Цей час то для однієї модифікації кращий, то для другої. В дужках вказано примірний час, який мав би бути у швидкості до ВК.

Для другого експерименту було взято вибірку 4 облич 2-ох жінок та 2-ох чоловіків. Кожне з облич має по 2-фотографії зроблені у невеликий проміжок часу. В нашій виборці еталонів знаходяться тільки перші чотири зображення. В виборці зображень, що будуть класифікуватися,

будуть знаходитися усі 8 зображень. В цьому експерименті перевіriamo ефективність методів на класифікацію обличчя за одним фото. Треба зазначити, що на зображені фото пари близнюків (брат-сестра) (рис.3.25).



Рисунок 3.25 – Тестова вибірка зображень №2

Для даного експерименту брали 41 ОТ для зображення. На рисунку 3.26 представлена тестова вибірка зображень №2 з їх ОТ.

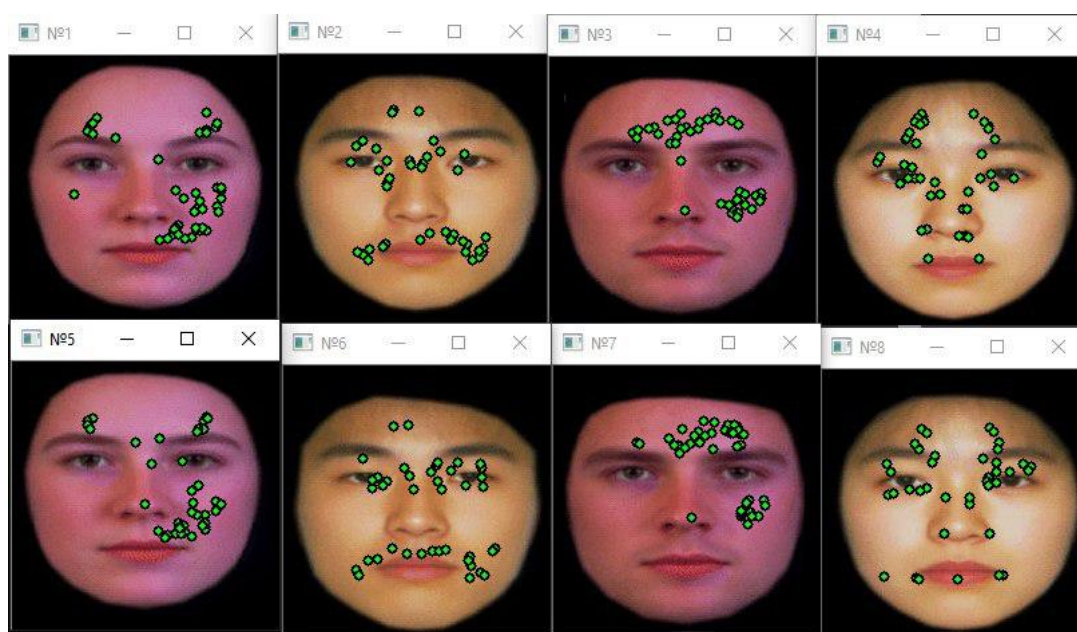


Рисунок 3.26 – Тестова вибірка зображень №2 з їх ОТ

МПЦД показав доволі впевнені результати, класифікувавши усі зображення (рис.3.27). Дані для інших алгоритмів будемо порівнювати саме з цим методом.

	s1	s2	s3	s4
s1	31	0	3	7
s2	7	24	5	5
s3	1	2	38	0
s4	1	8	1	31
s5	22	2	2	15
s6	10	19	10	2
s7	8	5	28	0
s8	0	9	6	26

Рисунок 3.27 – Метод центрального дескриптора

Метод вагових коефіцієнтів знову показав не зовсім стійкий результат, не розпізнавши зображення 6, окрім значно кращого детектування зображень 1 та 5 європеїдної жінки (рис.3.28). У метода вагових коефіцієнтів чітких переваг виявлено не було.

	s1	s2	s3	s4
s1	37	0	1	3
s2	12	18	4	7
s3	3	1	36	1
s4	6	2	2	31
s5	32	0	1	8
s6	17	14	7	3
s7	15	1	25	0
s8	3	6	4	28

Рисунок 3.28 – Метод вагових коефіцієнтів

Метод відмови від слабких показав результат кращий, ніж МПЦД й отримав майже ідеальні результати (рис.3.29). У МПЦД були більш погано класифіковані зображення 5 та 6. Для інших зображень класифікація пройшла ідентично МПЦД.

	s1	s2	s3	s4
s1	32	0	2	7
s2	5	24	5	7
s3	2	2	37	0
s4	0	8	1	32
s5	27	1	1	12
s6	8	20	5	8
s7	7	4	28	2
s8	0	10	6	25

Рисунок 3.29 – Відмова від слабких

Цікавіше, ніж в першому експерименті показав себе алгоритм ВК з дистанцією (рис.3.30), показавши гарний результат детектування перших чотирьох зображень та трохи гірший результат з другою четвіркою. Цікавим є той факт, що зображення 6 розпізнається набагато краще, ніж в інших алгоритмах. Проте виникли деякі проблеми з зображенням 5, результати не такі ідеальні, як очікувалось. Якщо говорити в цілому, то важко дати однозначну оцінку алгоритму. Проблема в тому, що для даного алгоритму однакова первина обробка вибірки з МПЦД не гарантує честність їх порівняння. Далі в цьому ж експерименті буде порівняно ці два алгоритми при різних первинних обробках (будуть узяті різні групи ОТ для зображень).

	s1	s2	s3	s4
s1	31	0	3	7
s2	3	33	1	4
s3	0	6	35	0
s4	0	12	0	29
s5	16	9	3	13
s6	6	29	2	4
s7	0	9	32	0
s8	0	17	2	22

Рисунок 3.30 – Метод вагових коефіцієнтів з дистанцією

Перейдемо до ВВС з дистанцією (рис.3.31). Дана модифікація ВВС є повністю невдалою. На відміну від ВВС, що показав найкращий результат в другому експерименті, ВВС з дистанцією напроти показав найгірший. Хоча можна й зазначити, що розпізнані погано були тільки брат-сестра близнюки азіатського походження. Вірніше брат був віднесений до сестри, сама сестра була

розпізнана добре. Але якщо брати до уваги, що інші алгоритми справилися з цією проблемою, то результати можна вважати невдалими.

	s1	s2	s3	s4
s1	30	3	3	5
s2	0	41	0	0
s3	0	6	35	0
s4	0	22	0	19
s5	17	13	4	7
s6	3	34	3	1
s7	0	12	29	0
s8	0	26	2	13

Рисунок 3.31 – Відмова від слабких з дистанцією

Тепер для ВК з дистанцією збудуємо за допомогою коректування параметрів алгоритму BRISK та первинної обробки інші ОТ, при яких даний алгоритм буде показувати більш гарні результати (рис. 3.32).

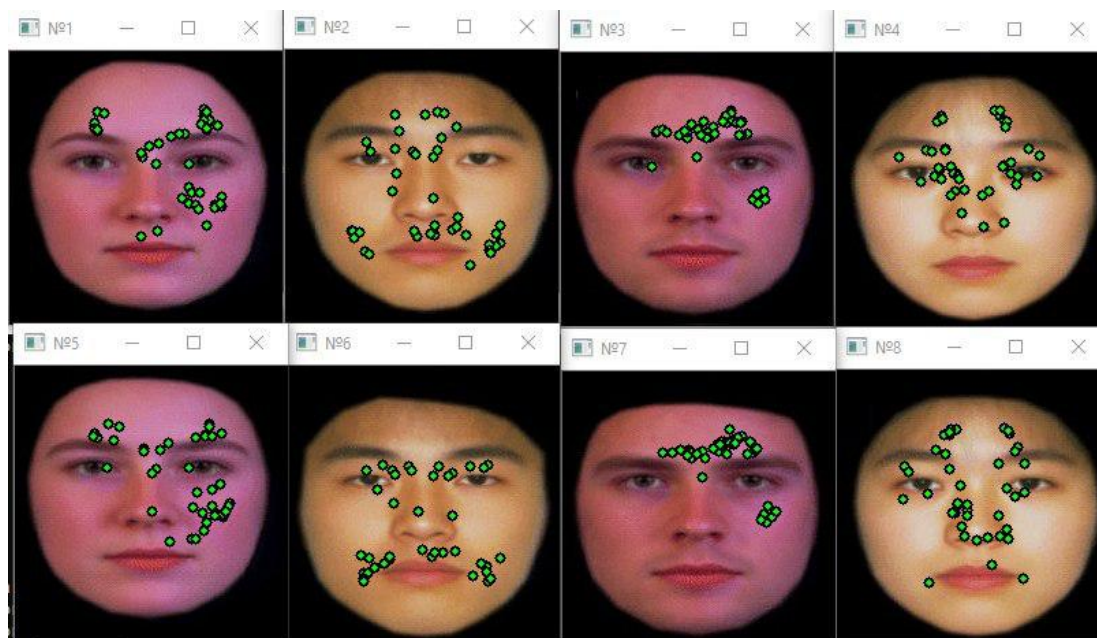


Рисунок 3.32 – Тестова вибірка зображень №2 з перебудованими ОТ

Це було зроблено, оскільки минулі параметри первинної обробки підбиралися саме під МПЦД. Для достовірності буде проведено порівняння МПЦД з ВК з дистанцією при параметрах побудови ОТ, що підбиралися під сам ВК з дистанцією (рис.3.33).

	s1	s2	s3	s4		s1	s2	s3	s4
s1	30	0	0	11	s1	31	0	0	10
s2	6	29	1	5	s2	9	25	0	7
s3	1	0	39	1	s3	4	0	36	1
s4	1	13	0	27	s4	4	9	0	28
s5	22	10	2	7	s5	24	4	3	10
s6	12	23	0	6	s6	15	12	2	12
s7	1	0	40	0	s7	2	0	39	0
s8	0	14	0	27	s8	3	12	0	26

Рисунок 3.33 – Порівняння ВК з дистанцією з МПЦД

На цей раз результати стали протилежними. На рисунку 3.33 з правої частини результат класифікації ВК з дистанцією, а з лівої частини МПЦД. В цьому випадку для зображення номер 6 виникли проблеми у МПЦД, коли в минулі рази проблема була якраз у ВК з дистанцією.

Розглянувши рисунок 3.34 бачимо доволі віддзеркалений результат для ВВС та ВВС з дистанцією. Тільки метод вагових коефіцієнтів продовжив показувати доволі невтішні результати. Тому для фінального підсумку другого експерименту будемо брати дані з різними параметрами для знаходження ОТ й вже на основі двох тестувань обирати найбільш стійкий з алгоритмів.

	s1	s2	s3	s4		s1	s2	s3	s4		s1	s2	s3	s4
s1	27	1	1	12	s1	29	0	0	12	s1	38	0	0	3
s2	2	29	0	10	s2	8	25	0	8	s2	16	21	0	4
s3	2	0	38	1	s3	2	0	38	1	s3	3	0	38	0
s4	0	15	0	26	s4	2	9	0	30	s4	9	6	0	26
s5	20	13	3	5	s5	24	5	1	11	s5	36	0	1	4
s6	7	21	0	13	s6	12	13	1	15	s6	20	12	2	7
s7	2	0	39	0	s7	2	0	39	0	s7	2	0	39	0
s8	0	17	0	24	s8	0	12	0	29	s8	12	8	0	21

Рисунок 3.34 – Результати ВВС з дистанцією, ВВС та ВК

Результати другого дослідження приведено у таблиці 3.2.

Таблиця 3.2 – Результати другого дослідження

Метод	Загальна кількість вірних ОТ №1	Загальна кількість вірних ОТ №2	Стійкість №1	Стійкість №2	AVG ОТ	AVG стійкості	Загальна оцінка
МПЦД	219	221	23.5	15.5	220	19.5(195)	415 (4)
Вагових Коефіцієнтів (ВК)	221	231	21.5	20	226	20.75(207)	433 (3)
Відмова від здабкого (ВВС)	225	222	27.5	18	223	22.75(227)	450 (2)
ВК з дистанцією	227	237	27.5	24	232	25.75(257)	489 (1)
ВВС з дистанцією	218	224	12.5	23	221	17.75(177)	398 (5)

Стійкість підраховується наступним шляхом. За 25+ вірних ОТ +5 показник стійкості, за просто вірно класифіковане зображення +3. За кожний еталон, який переміг вірний еталон. За кожний неправильний еталон, який не переміг, але до нього було віднесено 10+ точок віднімаємо 2 бали стійкості, та -0.5, за кожні невірно віднесені від 5 до 9 точок до одного невірного еталону.

Загальна оцінка вираховується сумою середньої кількості ОТ та загальною стійкістю помноженою на 10. В даному експерименті найкращі результати показали ВВС та ВК з дистанцією - набрали найбільшу кількість балів та показали найкращі результати. ВВС з дистанцією на відміну від першого дослідження показав найкращий результат як й при двох різних параметрах.

Останній експеримент проходив на тій же виборці облич, що й другий дослід. Але головною відмінністю було те, що для вхідних зображень також будуємо центр за МПЦД та порівнюємо його за допомогою наших алгоритмів. Для кожного алгоритму використовували два варіанти первинної обробки з попередніх дослідів. Оскільки кожний алгоритм отримує тільки один дескриптор ОТ, який є центральним дескриптором вхідного зображення, то будемо розглядати дані, за якими класифікується наш алгоритм. Оскільки для одних алгоритмів використовуємо відстань, а для інших їх вагу, порівнювати будемо тільки за кількістю правильно класифікованих зображень, та тим, наскільки однозначний результат. Результат неоднозначний, коли різниця між значеннями для вірного центру більше від значення, що набрав невірний центр, на відносно невелике значення.

Для МПЦД використовується відстань Хемінга, тому перемагає центр, до якого відстань найменша. МПЦД у другому варіанті вибірки ОТ, невірно класифікував зображення 6, віднісши його до сестри близнючки. В першому варіанті присутні неякісно класифіковані з 5 по 7 зображення (рис. 3.35).

	s1	s2	s3	s4		s1	s2	s3	s4
s1	0	84	54	67	s1	0	85	64	63
s2	84	0	86	57	s2	85	0	131	54
s3	54	86	0	79	s3	64	131	0	115
s4	67	57	79	0	s4	63	54	115	0
s5	68	110	84	77	s5	28	91	62	71
s6	70	38	68	41	s6	60	53	96	45
s7	42	88	38	79	s7	64	127	22	133
s8	122	78	118	67	s8	76	63	128	33

Рисунок 3.35 – Метод центрального дескриптора

На відміну від МПЦД ВК підраховує вагу, й перемагає той центр, вага якого найбільша. Метод ВК не впорався в двох варіантах, не розпізнавши жодного разу 6 й 7 зображення при першому варіанті побудови вибірки ОТ (рис. 3.36).

	s1	s2	s3	s4		s1	s2	s3	s4
s1	393	271	358	316	s1	433	284	343	325
s2	275	373	300	332	s2	307	386	230	342
s3	323	268	423	300	s3	347	219	424	241
s4	309	309	315	401	s4	346	324	258	409
s5	309	233	311	294	s5	399	275	341	311
s6	303	332	332	353	s6	349	327	291	356
s7	341	265	380	296	s7	344	222	399	242
s8	218	280	244	319	s8	325	313	236	372

Рисунок 3.36 – Метод вагових коефіцієнтів

Метод Відмови від слабких показав результат антологічний МПЦД, а переможець підраховується так само як у випадку з методом ВК (рис. 3.37).

ВК та ВВС з дистанцією показали трохи кращі результати ніж їх попередники, але дати однозначну відповідь, що результати набагато кращі не можемо, бо усі алгоритми окрім ВК не змогли класифікувати тільки в одному з випадків (рис. 3.38, 3.39).

	s1	s2	s3	s4		s1	s2	s3	s4
s1	483	339	401	382	s1	492	353	391	390
s2	331	457	337	392	s2	342	459	263	398
s3	395	335	475	358	s3	394	273	486	282
s4	375	399	353	474	s4	380	399	297	478
s5	379	293	345	344	s5	450	339	389	374
s6	371	427	369	344	s6	390	403	335	420
s7	423	331	433	356	s7	390	279	459	286
s8	263	355	273	362	s8	352	379	267	444

Рисунок 3.37 – Відмова від слабких

	s1	s2	s3	s4		s1	s2	s3	s4
s1	0	50	32	42	s1	0	50	40	41
s2	58	0	61	34	s2	63	0	96	33
s3	34	52	0	50	s3	42	83	0	84
s4	42	31	54	0	s4	43	30	83	0
s5	41	69	56	53	s5	16	55	41	49
s6	45	20	45	24	s6	41	29	66	26
s7	25	54	21	52	s7	44	81	12	83
s8	87	46	89	41	s8	53	36	94	18

Рисунок 3.38 – Метод вагових коефіцієнтів з дистанцією

	s1	s2	s3	s4		s1	s2	s3	s4
s1	0	59	37	46	s1	0	53	47	44
s2	76	0	69	41	s2	75	0	11	40
s3	44	61	0	58	s3	49	93	0	98
s4	54	29	61	0	s4	56	30	94	0
s5	52	82	65	65	s5	21	60	48	52
s6	54	12	53	28	s6	51	28	75	29
s7	30	63	21	59	s7	51	90	13	96
s8	110	51	101	56	s8	70	40	109	17

Рисунок 3.39 – Відмова від слабких з дистанцією

Для цих же даних було вирішено при різній кількості ОТ та при різних параметрах провести більше 50 дослідів, у яких бралась би до уваги тільки кількість нерозпізнаних зображень. Не брали

до уваги наскільки чітке було розпізнавання та наскільки гарні дані були узяті. Перевіряли, як працюють алгоритми при випадкових ситуаціях.

Результати другого дослідю приведено у таблиці 3.3.

Таблиця 3.3 – Результати третього дослідю

Метод	Кількість помилок При 310Т	Кількість помилок При 410Т	Кількість помилок При 510Т	Кількість помилок При 610Т	Загальна кількість помилок
МПЦД	8	7	6	2	23
Вагових Коефіцієнтів (ВК)	11	9	10	11	41
Відмова від сдабкого (ВВС)	8	7	5	3	23
ВК з дистанцією	8	5	4	3	19
ВВС з дистанцією	8	4	4	3	18

Результати були доволі цікаві, бо на різноманітних вибірках ОТ для зображень найкращий результат показав ВВС з дистанцією, який до цього показував не найкращі дані при більш детальному розгляді самого детектування. Проте ВК показав себе повністю недієздатним в даному досліді й для порівняння центра з центром недієвим варіантом. Зате ВК з дистанцією показав майже такий самий гарний результат як й ВВС з дистанцією, зробивши лише на одну помилку більше. МПЦД та ВВС показали однакові результати, що в даному випадку більше працює у сторону МПЦД, бо він реалізується швидше.

Підведемо підсумок проведених експериментів, але спочатку згадаємо результати при різних дослідях (табл. 3.4).

Найбільш дієвим як й при порівнянні центра з центром, так й центра з дескрипторами вхідного зображення показав себе ВК з дистанцією. Це обумовлюється тим, що надаємо більше значення тим бітам (значенням) центра, які при його побудові були більш чіткі (мали перевагу над своїм протилежним бітом). А нестійкі значення, навпаки, впливали менше.

Таблиця 3.4 – Зведена таблиця результатів дослідження

Метод	Дослід 1	Дослід 2	Дослід 3	Приблизне загальне місце
МПЦД	2	4	3	3 (9)
Вагових Коефіцієнтів (ВК)	5	3	5	4 (13)
Відмова від сдабкого (ВВС)	1	2	4	2 (7)
ВК з дистанцією	3	1	2	1 (6)
ВВС з дистанцією	4	5	1	5 (10)

Доволі гарні результати показав й метод ВВС, давши слабіну лише в третьому досліді. Він при роботі з різноманітними даними не показав значної переваги над МПЦД при порівнянні центрів. Це можна обумовити тим, що він погано працює з центрами, які мають багато нестійких значень, бо повністю їх ігнорує. Проте при роботі з центрами, які мають невелику (або помірну) кількість нестійких значень, алгоритм працює набагато краще від МПЦД. Варто зазначити, що проблему можна нивілювати, якщо при порівнянні замість вагів центру еталону брати ваги центру вхідного зображення.

Метод ВВС з дистанцією показав гарні результати лише в третьому досліді. Можливо при порівнянні замість вагів центру еталону брати ваги центру вхідного зображення, результати би були ще гарніше. Хоча він й показав не найкращі результати в наших дослідях, відкидати його не варто. Останій дослід показав, що можливо були підібрані не найкращі обставини для даного алгоритму.

Алгоритм ВК є повністю невдалим й показав провальні результати у всіх дослідях. Хоча теоретично даний алгоритм мав би працювати краще всіх, але на практиці він показав найгірші результати.

4 РОЗДІЛ ОХОРОНА ПРАЦІ

Завданням даної магістерської роботи було дослідження та модернізація методу класифікації зображень на підставі множини ключових точок. Так як в процесі проектування використовувалося комп'ютерне обладнання, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для умов праці з використанням персонального комп'ютера.

В даному розділі проведено аналіз потенційних небезпечних та шкідливих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

4.1 Загальні питання з охорони праці

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» [28] визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави і особистої відповідальності.

4.2 Аналіз стану умов праці

Робота над завданням даної магістерської роботи проходитиме в приміщенні багатоквартирного будинку. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1.

Таблиця 4.1 – Розміри приміщення

Найменування	Значення
Довжина, м	5
Ширина, м	5
Висота, м	2.8
Площа, м ²	25
Об'єм, м ³	70

Згідно з ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» [29] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [30] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Приміщення знаходиться на другому поверсі трьох поверхової будівлі і має об'єм 70 м³, площу – 25 м². Обладнано одне місце праці укомплектоване ПК.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	770	680 ÷ 800
Висота простору для ніг, мм	750	не менше 600
Ширина простору для ніг, мм	550	не менше 500

Продовження таблиці 4.2 – Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	450	400 ÷ 500
Ширина сидіння, мм	450	не менше 400
Глибина сидіння, мм	470	не менше 400
Висота поверхні спинки, мм	400	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	750	700 ÷ 800

Температура в приміщенні протягом року коливається у межах 18–24°C, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум знаходиться на рівні 50 дБА. Система вентилявання приміщення — природна неорганізована, а опалення — централізоване.

4.2.3 Навантаження та напруженість процесу праці

За фізичним навантаженням робота відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту.

Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви програм тривалістю 15 хв. через кожну годину роботи.

4.3 Виробнича санітарія

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу

Аналіз небезпечних та шкідливих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання НПАОП 0.00-7.15-18 [33] «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга $U=+220\text{В} \pm 5\%$;
- робочий струм $I=2\text{А}$;
- споживана потужність $P=350\text{ Вт}$.

Робоче місце має відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [30].

Таблиця 4.3 – Аналіз небезпечних і шкідливих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
Фізичні			
- підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	система організаційних і технічних заходів і засобів, що забезпечують захист від шкідливого і небезпечного впливу електричного струму, електричної дуги, електромагнітного поля і статичної електрики.	4	[31]
- недостатність природного світла	порушення умов праці (вимог до приміщень)	2	[32]
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів	3	[32]

Продовження таблиці 4.3 – Аналіз небезпечних і шкідливих факторів

<i>психофізіологічні:</i>			
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	4	[33] [30]
- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці- сидіння користувача,) та організації робочого часу - безпервна робота)	2	[33] [30]

4.3.2 Пожежна безпека

Небезпека розвитку пожежі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, вентиляції і кондиціонування.

Запобігти утворенню горючого середовища (замінити горючі речовини і матеріали на негорючі і важкогорючі) не надається технічно можливим. Тому проектом передбачаються засоби запобігання утворення (або внесення) в горюче середовище джерел запалювання.

Згідно ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою» [34] таке приміщення, площею 25 м², відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому можливо встановлення автоматичної пожежної сигналізації із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол.

4.3.3 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів.

Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

4.4 Гігієнічні вимоги до параметрів виробничого середовища

4.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. Оптимальні значення мікроклімату для робочого місця відповідають ДСН 3.3.6.042-99 [29] (табл. 4.4):

Таблиця 4.4 – Норми мікроклімату робочої зони об'єкту

Період року	Категорія робіт	Температура С ⁰	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату. Дане приміщення обладнане системою опалення, кондиціонування повітря. Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному

провітрюванні.

Рівні позитивних і негативних іонів у повітрі мають відповідати ДСН 3.3.6.042-99 [29].

4.4.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає ДБН В.2.5-28:2018 [32]. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення.

Розрахунок освітлення.

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше $1/8$, в побутових – $1/10$:

$$S_b = \left(\frac{1}{5} \div \frac{1}{10} \right) \cdot S_n, \quad (4.1)$$

де S_b – площа віконних прорізів, m^2 ;

S_n – площа підлоги, m^2 .

$$S_n = a \cdot b = 5 \cdot 5 = 25 \text{ м}^2,$$

$$S = 1/8 \cdot 25 = 3,125 \text{ м}^2.$$

Приймаємо 2 вікна площею $S=1,6 \text{ м}^2$ кожне.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні.

Розрахунок кількості світильників n виробляється по формулі (4.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \quad (4.2)$$

де E – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

S – освітлювана площа, m^2 ; $S = 25 m^2$;

Z – поправочний коефіцієнт світильника ($Z=1,15$ для ламп розжарювання та ДРЛ; $Z = 1,1$ для люмінесцентних ламп) приймаємо рівним 1,1;

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

U – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

M – число люмінесцентних ламп в світильнику – 2;

F – світловий потік лампи – 5400лм (для ЛБ-80).

Підставивши числові значення у формулу (рис.4.2), отримуємо:

$$n = \frac{300 \cdot 25 \cdot 1,1 \cdot 1,5}{5400 \cdot 0,575 \cdot 2} \approx 2,0$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з двох люмінесцентних ламп загальною потужністю 160 Вт, напругою – 220.

4.4.3 Шум та вібрація, електромагнітне випромінювання

Рівень шуму, зумовлений як роботою системного блоку, клавіатури, так і друкуванням на принтері, а також зовнішніми чинниками, коливається у межах 50–65 дБА ДСН 3.3.6.042-99 [29].

Віброізоляцію можливо здійснювати за допомогою спеціальної прокладки під системний блок, яка послаблює передачу вібрацій робочого столу. Вібрація на робочому місці в приміщенні, що розглядається, відповідає нормам ДСН 3.3.6.042-99 [29].

4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання передбачено наступні заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;

- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціонування повітря.

Крім того, потрібно дотримуватися правил безпеки під час експлуатації інших електричних приладів та вимоги безпеки при надзвичайних ситуаціях.

Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом, приміщення в якому проводиться робота відноситься до першого класу (без підвищеної небезпеки). Коефіцієнт використання вертикальних заземлювачів η_v в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4...0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги) враховують за допомогою коефіцієнта використання горизонтального заземлювача η_c .

Послідовність розрахунку.

- 1) Визначається необхідний опір штучних заземлювачів $R_{шт.з.}$:
- 2)

$$R_{шт.з.} = \frac{R_d \cdot R_{пр.з.}}{R_{пр.з.} - R_d}, \quad (4.3)$$

де $R_{пр.з.}$ – опір природних заземлювачів; R_d – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то $R_{шт.з.} = R_d$.

Підставивши числові значення у формулу (рис.4.3), отримуємо:

$$R_{шт.з.} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}$$

3) Опір заземлення в значній мірі залежить від питомого опору ґрунту ρ , Ом·м. Приблизне значення питомого опору глини приймаємо $\rho = 40$ Ом·м (табличне значення).

4) Розрахунковий питомий опір ґрунту, $\rho_{розр.}$, Ом·м, визначається відповідно для вертикальних заземлювачів $\rho_{розр.в.}$, і горизонтальних $\rho_{розр.г.}$, Ом·м за формулою:

$$\rho_{розр.} = \psi \cdot \rho, \quad (4.4)$$

де ψ – коефіцієнт сезонності для вертикальних заземлювачів I кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів $\rho_{\text{розр.в}}=1,7$ і горизонтальних $\rho_{\text{розр.г}}=5,5$ Ом·м.

$$\rho_{\text{розр.в}} = 1,7 \cdot 40 = 68 \text{ Ом}\cdot\text{м}$$

$$\rho_{\text{розр.г}} = 5,5 \cdot 40 = 220 \text{ Ом}\cdot\text{м}$$

5) Розраховується опір розтікання струму вертикального заземлювача $R_{\text{в}}$, Ом, за формулою (4.5).

$$R_{\text{в}} = \frac{\rho_{\text{розр.в}}}{2 \cdot \pi \cdot l_{\text{в}}} \cdot \left(\ln \frac{2 \cdot l_{\text{в}}}{d_{\text{ст}}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_{\text{в}}}{4 \cdot t - l_{\text{в}}} \right), \quad (4.5)$$

де $l_{\text{в}}$ – довжина вертикального заземлювача (для труб - 2–3 м; $l_{\text{в}}=3$ м);

$d_{\text{ст}}$ – діаметр стержня (для труб - 0,03–0,05 м; $d_{\text{ст}}=0,05$ м);

t – відстань від поверхні землі до середини заземлювача, яка визначається за формулою (4.6):

$$t = h_{\text{в}} + \frac{l_{\text{в}}}{2}, \quad (4.6)$$

де $h_{\text{в}}$ – глибина закладання вертикальних заземлювачів (0,8 м); тоді $t = 0,8 + \frac{3}{2} = 2,3$ м

$$R_{\text{в}} = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left(\ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

б) Визначається теоретична кількість вертикальних заземлювачів n штук, без урахування коефіцієнта використання $\eta_{\text{в}}$:

$$n = \frac{2 \cdot R_{\text{в}}}{R_{\text{д}}} = \frac{2 \cdot 18,5}{4} = 9,25 \quad (4.7)$$

І визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки $\eta_B = 0,57$ (табличне значення).

7) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання n_B , шт:

$$n_B = \frac{2 \cdot R_B}{R_d \cdot \eta_B} = \frac{2 \cdot 18,5}{4 \cdot 0,57} = 16,2 \approx 16 \quad (4.8)$$

8) Визначається довжина з'єднувальної стрічки горизонтального заземлювача l_c , м:

$$l_c = 1,05 \cdot L_B \cdot (n_B - 1), \quad (4.9)$$

де L_B – відстань між вертикальними заземлювачами, (прийняти за $L_B = 3$ м);
 n_B – необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м}$$

9) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки) R_r , Ом:

$$R_r = \frac{\rho_{\text{розр.г}}}{2 \cdot \pi \cdot l_c} \cdot \ln \frac{2 \cdot l_c^2}{d_{\text{см}} \cdot h_r}, \quad (4.10)$$

де $d_{\text{см}}$ – еквівалентний діаметр смуги шириною b , $d_{\text{см}} = 0,95b$, $b = 0,15$ м;

h_r – глибина закладання горизонтальних заземлювачів (0,5 м);

l_c – довжина з'єднувальної стрічки горизонтального заземлювача l_c , м

$$R_r = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом}$$

10) Визначається коефіцієнт використання горизонтального заземлювача η_c відповідно до необхідної кількості вертикальних заземлювачів n_B .

Коефіцієнт використання з'єднувальної смуги $\eta_c = 0,3$ (табличне значення).

11) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{зар}} = \frac{R_{\text{в}} \cdot R_{\text{г}}}{R_{\text{в}} \cdot \eta_{\text{с}} + R_{\text{г}} \cdot n_{\text{в}} \cdot \eta_{\text{в}}} \leq R_{\text{д}}. \quad (4.11)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова: $R_{\text{зар}} < 4 \text{ Ом}$, а саме:

$$R_{\text{зар}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_{\text{д}}$$

ВИСНОВКИ

У результаті виконання роботи проведено дослідницьку роботу, під час якої розроблено алгоритми покращення детектування зображень за допомогою бінарних центральних дескрипторів. Для детектування ОТ використовувався один з новітніх алгоритмів BRISK. Проведено порівняльний аналіз розроблених алгоритмів на різних вибірках зображень та при різних умовах оброблення.

При практичній побудові правил класифікації візуальних об'єктів із застосуванням детекторів ОТ треба звернути увагу на ефективний вибір та поєднання можливостей як детекторів ОТ, так і апарату для оброблення їх значень для досліджуваних баз зображень.

Серед досліджених варіантів найкращі результати класифікації зображень показав алгоритм «Вагових коефіцієнтів з дистанцією». Це обумовлюється тим, що було приділено більше значення тим бітам (значенням) центра, які при його побудові були більш чіткі (мали перевагу над своїм протилежним бітом).

Методи «Відмови від слабких» та «Відмови від слабких з дистанцією» також є доволі перспективними модифікаціями, але мають деякі проблеми з вибірками ОТ, де центр має множину настійких значень.

Метод побудови центрального дескриптора хоч й був перевершений у якості класифікування, проте всі аналізовані алгоритми програють йому у швидкодії оброблення даних. Варто зазначити, що даний метод класифікує зображення доволі якісно й може використовуватися у ситуаціях, коли потрібна гарна швидкодія та якісне класифікування.

Єдиний з розглянутих алгоритмів, який можна назвати невдалим, – це метод вагових коефіцієнтів, який на перший погляд мав бути найбільш ефективним, проте практика показала протележні результати.

Наукова новизна дослідження полягає у синтезі нового продуктивного методу класифікації зображень з використанням апарату бітового аналізу та оброблення даних у просторі дескрипторів ключових точок зображення з використанням вагових коефіцієнтів, що покращують якість класифікації. Головною перевагою над існуючими алгоритмами є швидкодія. Зведення зображення до масиву (дескриптору) з 512 значень, кожне значення може бути 0 або 1. До центрального дескриптору додається ще один масив з вагами, що дозволяє зменшити фактичний об'єм інформації для класифікації, пришвидшує процес класифікації у стільки разів, скільки особливих точок має зображення. Порівняння центра з центром пришвидшує роботу ще у стільки же разів й при тому якість класифікації майже не зменшується, а в деяких випадках навіть покращується.

Практична значущість роботи – отримання практичних програмних моделей для застосування та оцінювання ефективності методів класифікації з бінарним поданням даних та приведенням їх у центри з підтвердженням їх результативності на прикладних базах зображень.

Зважаючи на бінарний вид дескрипторів BRISK, є можливість для побудови результативних правил класифікації скористатися механізмами дерев рішення чи хешування.

Основною проблемою подальшої модернізації є простота алгоритмів, що дарує велику швидкодію, а нагромадження існуючих алгоритмів ваговими коефіцієнтами робить роботу алгоритмів більш довгою.

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Було наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

У двадцять першому столітті актуальною проблемою є забруднення навколишнього середовища. Проблеми з екологією зустрічаються в повсякденному житті і в будь-якій діяльності людини і не є винятком сфери, пов'язані з інформаційними технологіями. Важливо розуміти, що використання несправного обладнання або неправильної експлуатації впливає не тільки на здоров'я людини, а також на навколишнє середовище. Так само потрібно сортувати і утилізувати відходи в процесі роботи. Виходячи з вищесказаного можна зробити висновок, що дотримуватися нормативних документів охорони праці є обов'язковим.

Результати були апробовані на XI Всеукраїнській науково-практичній конференції «Майбутній науковець 2020» та X Всеукраїнській науково-практичній конференції «Електроніка та телекомунікації».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Duda, R.O., & Hart, P.E., & Stork, D.G. (2000) Pattern classification. Wiley.
2. Rublee, E., Rabaud, V., Konolige, K., & Bradski, G.R. (2011, November). ORB: An efficient alternative to SIFT or SURF. In ICCV (Vol. 11, No. 1, p. 2).
3. Gorokhovatskiy, V. A. (2016). Efficient Estimation of Visual Object Relevance during Recognition through their Vector Descriptions. Telecommunications and Radio Engineering, 75(14).
4. opencv.org. (2019). Available at: <https://docs.opencv.org/3.4.1/> [Accessed 5 Nov. 2019].
5. Rad, A.A. & Faez, K. & Qaragozlu, N. (2003). Fast Circle Detection Using Gradient Pair Vectors. Sydney, Australia: Proc. VIIth Digital Image Computing, 879-887.
6. Shapiro, L. (2001). Computer vision, Prentice. Prentice Hall.
7. Leutenegger, S., & Chli, M. & Siegwart, R.Y. (2011). BRISK: Binary Robust Invariant Scalable Keypoints. Computer Vision (ICCV), 2548 – 2555.
8. Анисимов, Б.В. & Курганов, В.Д. & Злобин, В.К. (1983). Распознавание и цифровая обработка изображений. Высшая школа.
9. Вапник, В. Н. (1979). Восстановление зависимостей по эмпирическим данным. М.: Наука.
10. Вапник, В. Н. & Червоненкис, А. Я. (1974). Теория распознавания образов. М.: Наука.
11. Глумов, Н.И. & Коломиец, Э.И. & Сергеев, В.В. (1993). Информационная технология распознавания объектов на изображении в режиме скользящего окна. Научное приборостроение, 72-88.
12. Глумов, Н.И. (1994). Яркостная нормализация цифрового изображения в скользящем окне. Деп. в ВИНТИ. Самара: Самарский государственный аэрокосмический университет, № 1881-В94.
13. Гонсалес, Р. & Вудс, Р. (2005). Цифровая обработка изображений. М.: Техносфера.
14. Гороховатский, В. А., Путятин, Е. П., & Столяров, В. С. (2017). Исследование результативности структурных методов классификации изображений с применением кластерной модели данных.
15. Гороховатский, В. А. (2014). Структурный анализ и интеллектуальная обработка данных в компьютерном зрении.
16. Дуда, Р. & Харт, П. Харт (1976). Распознавание образов и анализ сцен. М.: Мир.
17. Сойфера, В.А. (2003). Методы компьютерной обработки изображений. М.: ФИЗМАТЛИТ.
18. Осовский, С. (2002). Нейронные сети для обработки информации. М.: Финансы и статистика.

19. Прэтт, У.К. (1982). Цифровая обработка изображений. М.: Мир.
20. Путьгин, Е. П. (1990). Обработка изображений в робототехнике. Москва.
21. Солонина, А.И. & Улахович, Д.А. & Яковлев, Л.А (2001). Алгоритмы и процессоры цифровой обработки сигналов. СПб.: БХВ-Петербург.
22. machinelearning.ru (2019). Available at: <http://www.machinelearning.ru/wiki/index.php?title=Классификация> [Accessed 5 Nov. 2019].
23. Страуструп, Б. (1993). Язык программирования С++. Часть 1. К.: ДиаСофт.
24. Фор, А. (1989). Восприятие и распознавание образов. М.: Машиностроение.
25. Форсайт, Д., & Понс, Ж. (2004). Компьютерное зрение. Современный подход (р. 928). М.: ИД Вильямс.
26. Шапиро, Л. (2006). Стокман Дж. Компьютерное зрение. М.: Бином, 120-124.
27. Гороховатский, В.А. & Полякова, Т.В. (2018) Применение пространственных структур признаков для классификации изображений в компьютерном зрении (монография). Харьков: ФОП Панов А.Н.
28. Закон України «Про охорону праці». Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12> - 10.14.1992 р.
29. Державні санітарні норми. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» - Режим доступу: <https://zakon.rada.gov.ua/rada/show/va042282-99> - 01.02.1999 р.
30. Державні санітарні правила і норми. ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» - Режим доступу: <https://zakon.rada.gov.ua/rada/show/v0007282-98> - 10.12.1998 р.
31. Державний стандарт України. ДСТУ Б В.2.5-82:2016 «Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом» - Режим доступу: <http://epicentre.co.ua/dstu/doc28522.html> - 01.07.2016 р.
32. Державні будівельні норми. ДБН В.2.5-28:2018 «Природне і штучне освітлення» - Режим доступу: <http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf> - 03.10.2018
33. Нормативно-правовий акт з охорони праці. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» - Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18> - 14.02.2018 р.
34. ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою». Наказ від 15.06.2016 №158. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/v0158858-16](https://zakon.rada.gov.ua/rada/show/v0158858-16)

ДОДАТОК А
Тестові зображення

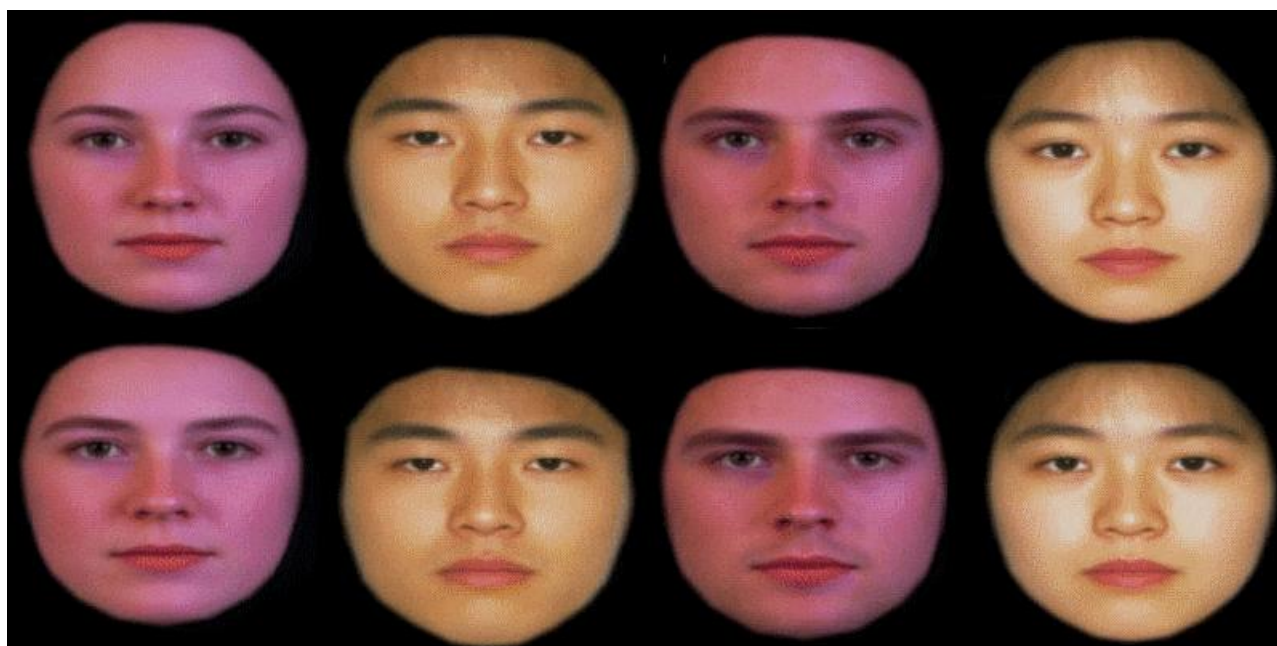


Рисунок А1 – Вибірка облич

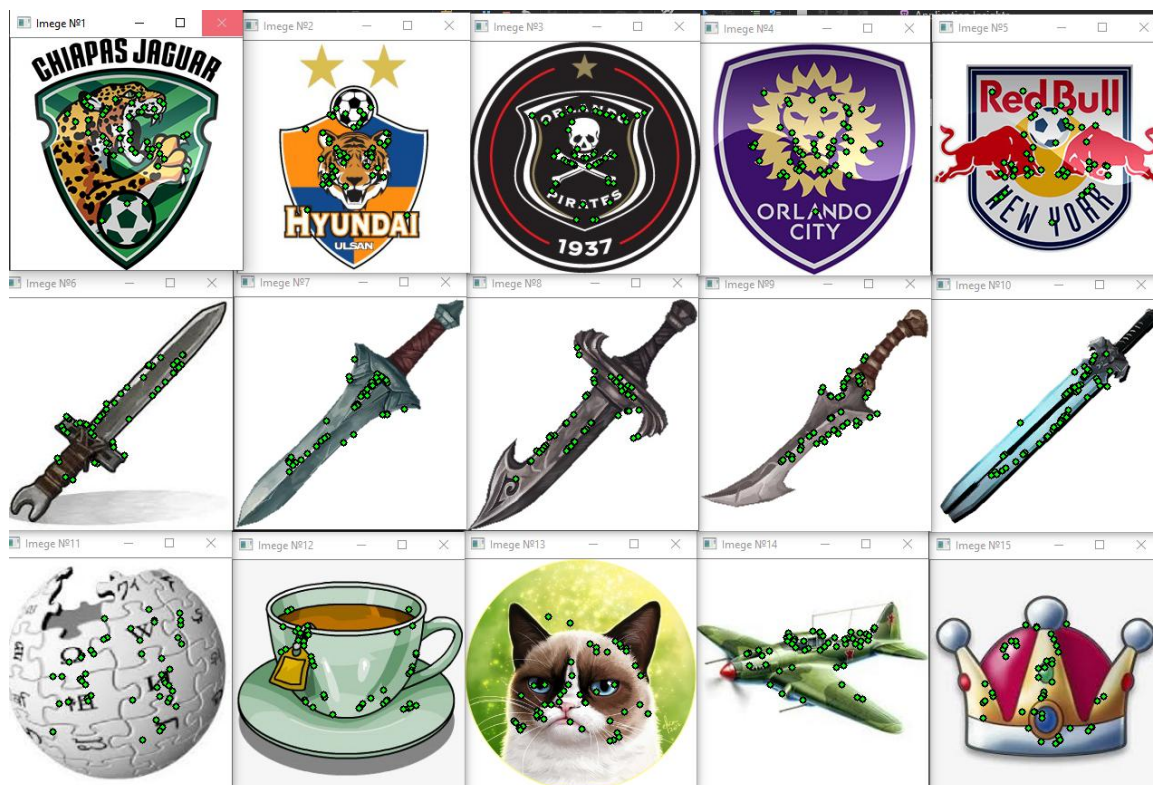


Рисунок А2 – Друга вибірка тестових зображень

ДОДАТОК Б

Комп'ютерна презентація



Рисунок Б1 – Слайд №1



Рисунок Б2 – Слайд №2

Мета роботи і постановка задачі

Метою цієї роботи є дослідження та модернізація методу класифікації зображень за допомогою бінарного центрального дескриптора. Потрібно програмно реалізувати алгоритмічну частину та розробити програму, що буде використовувати алгоритми для розпізнавання зображень у базі еталонів, провести досліді, на основі яких можна буде провести оцінювання результативності.

Для цього необхідно виконати наступні **завдання**:

- ➔ провести аналіз недоліків та переваг алгоритму побудови центрального дескриптора;
- ➔ розробити нові алгоритми чи модернізацію побудованого алгоритму класифікації з бінарними даними;
- ➔ реалізувати комп'ютерну модель детектування ОТ та обчислення множини дескрипторів;
- ➔ модернізувати систему тестування та аналізу даних для оцінювання роботи алгоритму;
- ➔ провести комп'ютерне тестування алгоритму на підготовленій реальній базі зображень;
- ➔ після побудови графіків встановити оцінки працездатності та конкурентоздатності для алгоритмів з розглядом їх подальшої модернізації;
- ➔ провести аналіз результатів дослідження.



Рисунок Б3 – Слайд №3

Середовище програмної реалізації

У рамках роботи розроблено алгоритми бінарної обробки даних для структурної класифікації зображень. Для реалізації обрана бібліотека з відкритим доступом Open Source Computer Vision Library (OpenCV). Це обумовлено можливістю безкоштовного використання бібліотеки та можливістю вільного продажу програмного забезпечення, яке написано з її використанням, а також високою швидкістю, що обумовлюється високопродуктивною мовою С++, й великим обсягом готових рішень, зв'язаних з обробленням зображень.



OpenCV спрямовується в основному на додатки в режимі реального часу та використовує інструкції MMX та SSE, коли це можливо.



Рисунок Б4 – Слайд №4

Задача класифікації

Задача, в якій є безліч об'єктів (ситуацій), розділених на класи. Потрібно побудувати алгоритм, здатний класифікувати довільний об'єкт з початкової множини.

Класифікувати об'єкт - значить, вказати номер (або найменування) класу, до якого належить даний об'єкт.

Класифікація об'єкта - номер або найменування класу, що видається алгоритмом класифікації в результаті його застосування до даного конкретного об'єкту.

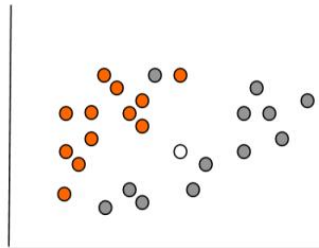


Рисунок Б5 – Слайд №5

Детектори та дескриптори особових точок

При практичній побудові правил класифікації візуальних об'єктів із застосуванням детекторів ОТ треба звернути увагу на ефективний вибір та поєднання можливостей як детекторів ОТ:

- ➔ метод класифікації за значенням центрального дескриптора;
- ➔ метод відмови від слабких;
- ➔ метод вагових коефіцієнтів.

так і апарату для оброблення їх значень для досліджуваних баз зображень:

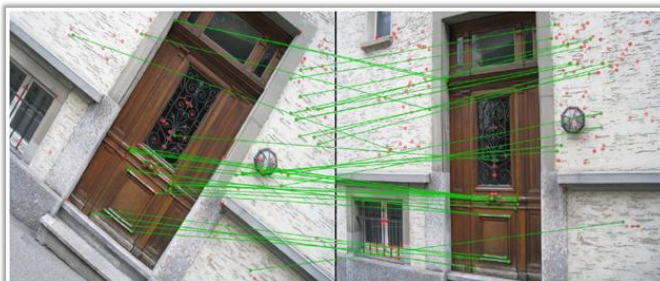
- ➔ дескриптор **BRISK**.



Рисунок Б6 – Слайд №6

BRISK

Наступне зображення показує приклад використання BRISK для порівняння зображення без змін й над яким провели деякі перетворення. Зелені лінії - дійсні співвідношення між точками, червоні кола виявляються ключовими точками.



Дескриптор BRISK є модифікованим алгоритмом FAST та має ручний шаблон вибірки. Зразки вибірки BRISK складаються з концентричних кілець

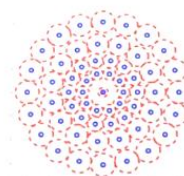


Рисунок Б7 – Слайд №7

Вибірki зображень для тестувань

Перша вибірка зображень



Друга вибірка зображень

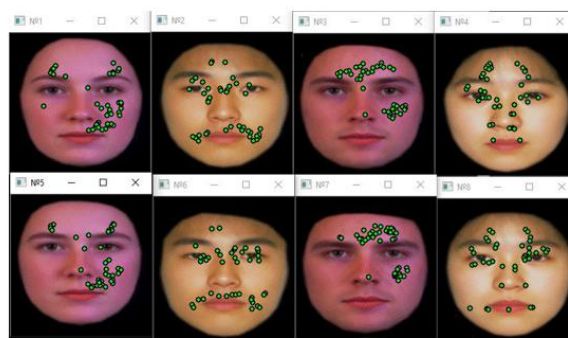


Рисунок Б8 – Слайд №8

Результати дослідів

Результати першого дослідів

Метод	УО	Загальна кількість вірних ОТ	Кількість Зображень у яких 25+ ОТ вірних	Помилки	Місце AVG
МПЦД	6000	412	10	0	2
Вагових Коефіцієнтів (ВК)	8200	392	7	1	5
Відмова від слабкого (ВВС)	8200 (8150)	434	12	0	1
ВК з дистанцією	8200 (8100)	442	11	1	3
ВВС з дистанцією	8200 (8150)	409	7	0	4

Рисунок Б9 – Слайд №9

Результати дослідів

Результати другого дослідів

Метод	Загальна кількість вірних ОТ №1	Загальна кількість вірних ОТ №2	Стійкість №1	Стійкість №2	AVG ОТ	AVG стійкості	Загальна оцінка
МПЦД	219	221	23.5	15.5	220	19.5(195)	415 (4)
Вагових Коефіцієнтів (ВК)	221	231	21.5	20	226	20.75(207)	433 (3)
Відмова від слабкого (ВВС)	225	222	27.5	18	223	22.75(227)	450 (2)
ВК з дистанцією	227	237	27.5	24	232	25.75(257)	489 (1)
ВВС з дистанцією	218	224	12.5	23	221	17.75(177)	398 (5)

Рисунок Б10 – Слайд №10

Результати дослідів

Результати третього досліду

Метод	Кількість помилок При 310Т	Кількість помилок При 410Т	Кількість помилок При 510Т	Кількість помилок При 610Т	Загальна кількість помилок
МПЦД	8	7	6	2	23
Вагових Коефіцієнтів (ВК)	11	9	10	11	41
Відмова від слабкого (ВВС)	8	7	5	3	23
ВК з дистанцією	8	5	4	3	19
ВВС з дистанцією	8	4	4	3	18

Рисунок Б11 – Слайд №11

Результати дослідів

Загальні результати досліджень

Метод	Дослід 1	Дослід 2	Дослід 3	Приблизне загальне місце
МПЦД	2	4	3	3 (9)
Вагових Коефіцієнтів (ВК)	5	3	5	4 (13)
Відмова від слабкого (ВВС)	1	2	4	2 (7)
ВК з дистанцією	3	1	2	1 (6)
ВВС з дистанцією	4	5	1	5 (10)

Рисунок Б12 – Слайд №12

Висновки

У середовищі C++ з використанням бібліотеки програм OpenCV розроблено програмну модель для класифікації зображень на підставі описів, сформованих детектором BRISK.

Головною перевагою над існуючими алгоритмами є швидкодія. Зведення зображення до масиву (дескриптору) з 512 значень, кожне значення може бути 0 або 1. До центрального дескриптору додається ще один масив з вагами, що дозволяє зменшити фактичний об'єм інформації для класифікації, пришвидшує процес класифікації у стільки разів, скільки особливих точок має зображення. Порівняння центра з центром пришвидшує роботу ще у стільки ж разів й при тому якість класифікації майже не зменшується, а в деяких випадках навіть покращується.

Основною проблемою подальшої модернізації є простота алгоритмів, що дарує велику швидкодію, а нагромадження існуючих алгоритмів ваговими коефіцієнтами робить роботу алгоритмів більш довгою

Рисунок Б13 – Слайд №13

Висновки

У результаті виконання роботи проведено дослідницьку роботу, під час якої розроблено алгоритми покращення детектування зображень за допомогою бінарних центральних дескрипторів. Для детектування ОТ використовувався один з новітніх алгоритмів BRISK. Проведено порівняльний аналіз розроблених алгоритмів на різних вибірках зображень та при різних умовах оброблення.



Рисунок Б14 – Слайд №14

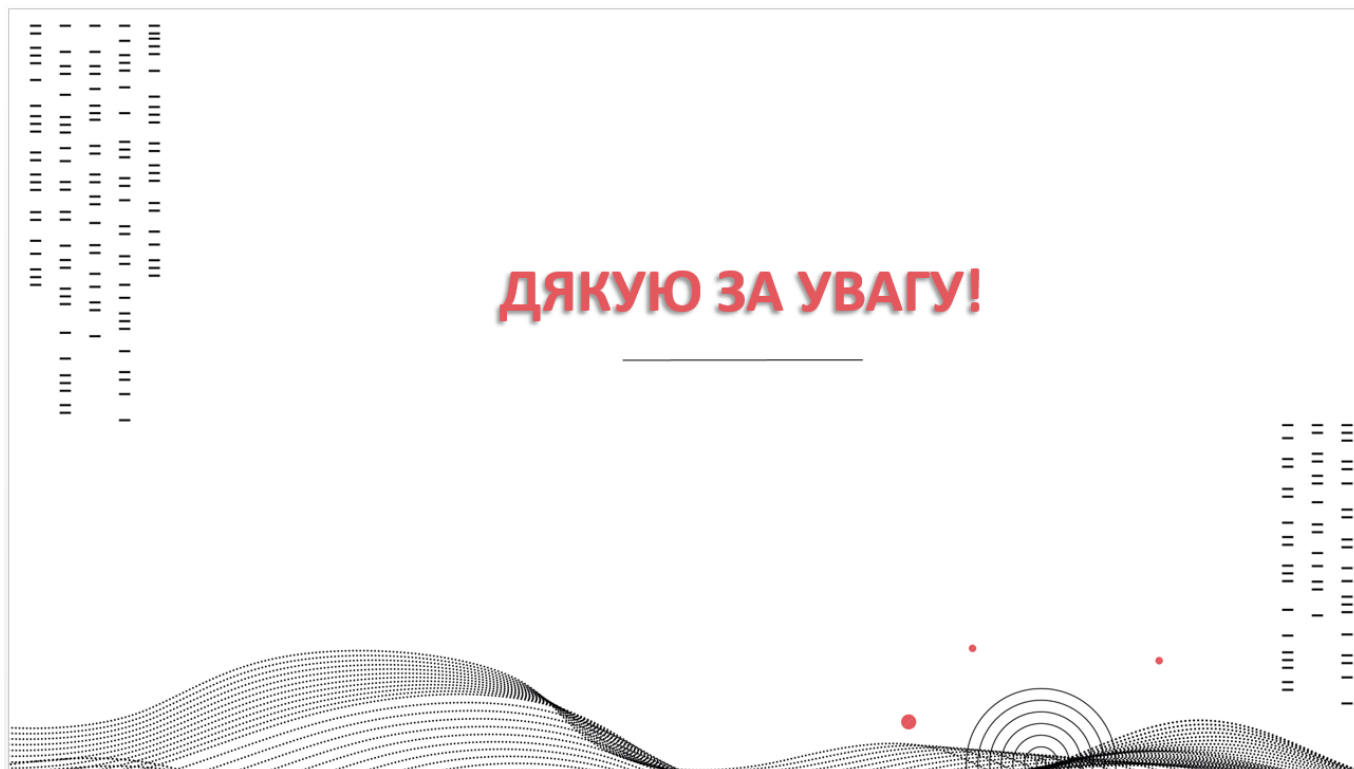


Рисунок Б15 – Слайд №15