

РЕФЕРАТ

Пояснювальна записка містить 60 сторінок, 31 рисуноків, 7 таблиць, 1 додаток, 18 джерел.

Метою дипломної роботи є дослідження математичного алгоритму верифікації відбитків пальців на основі циліндр–кодів з розробкою методу побудови метрики циліндрів. Для досягнення цього були поставлені наступні цілі дослідження:

- дослідити основні методи розпізнавання відбитків пальців;
- дослідити метод збереження відбитків пальців у вигляді циліндр–кодів;
- розробити метод побудови метрики циліндрів для порівняння відбитків пальців;
- провести експериментальне дослідження розробленого алгоритму на реальних даних.

Об'єктом дослідження дипломної роботи є процес ідентифікації осіб за допомогою відбитків пальців.

Предметом розробки є алгоритм верифікації відбитків пальців на основі циліндр–кодів та метод побудови метрики циліндрів.

Методи дослідження: кількісний аналіз, програмна реалізація та експериментальне дослідження.

Результатами проведеної роботи є:

- Розробка методу побудови метрики циліндрів, який дозволить отримати швидкий та ефективний спосіб порівняння відбитків пальців.
- Розширення знань про застосування циліндр–кодів для ідентифікації осіб за допомогою відбитків пальців.
- Проведення експериментів на реальних даних, що дозволить оцінити ефективність розробленого алгоритму верифікації та його можливості для практичного застосування в системах безпеки та ідентифікації осіб.

Ключові слова: МІКРОНЕРІВНОСТІ, ВІДБИТКИ ПАЛЬЦІВ, ЦИЛІНДРИЧНІ КОДИ, ТЕКСТУРА, ЛОКАЛЬНЕ СОРТУВАННЯ ЗА СХОЖІСТЮ (LSS), ШПФ, СТФ, ФУР'Є, РЕЛАКСАЦІЯ, МСС, МТСС, БАЗА ДАНИХ (БД), ГАБОР, ПОСЛІДОВНЕ СЕРЕДНЄ КВАНТУВАННЯ (SMQT), КОЕФІЦІЄНТ РІВНИХ ПОМИЛОК (EER), КОМПРОМІС ПОМИЛОК ВИЯВЛЕННЯ (DET), КОЕФІЦІЄНТ ПОМИЛКОВОГО ПРИЙНЯТТЯ (FAR), КОЕФІЦІЄНТ ПОМИЛКОВОГО ВІДХИЛЕННЯ (FRR), РОБОЧІ ХАРАКТЕРИСТИКИ ПРИЙМАЧА (ROC).

ABSTRACT

The explanatory note contains 60 pages, 31 figures, 7 tables, 1 annexes, 18 sources.

The aim of the thesis is the investigation of a mathematical algorithm for fingerprint verification based on cylinder–codes, with the development of a method for constructing cylinder metrics. To achieve this, the following research objectives were set:

- Explore the main methods of fingerprint recognition;
- Investigate the method of storing fingerprints in the form of cylinder–codes;
- Develop a method for constructing cylinder metrics for fingerprint comparison;
- Conduct an experimental investigation of the developed algorithm on real data.

The subject matter research is the process of person identification by fingerprints.

The subject of the work is the algorithm for fingerprint verification based on cylinder–codes and the method of building cylinder metrics.

Research methods: quantitative analysis, software implementation, and experimental investigation.

The results of the work are:

The development of a method for constructing cylinder metrics, which will provide a fast and efficient way to compare fingerprints.

Expanding knowledge about the application of cylinder–codes for person identification by fingerprints.

Conducting experiments on real data, which will allow the evaluation of the efficiency of the developed verification algorithm and its potential for practical use in security and person identification systems.

Keywords: MICROSTRUCTURES, FINGERPRINTS, CYLINDER CODES, TEXTURE, LOCAL SORTING BY SIMILARITY (LSS), SHPF, STF, FOURIER, RELAXATION, MCC, MTCC, DATABASE (DB), GABOR, SEQUENTIAL AVERAGE QUANTIZATION (SMQT), EQUAL ERROR RATE (EER), DETECTION ERROR TRADEOFF (DET), FALSE ACCEPTANCE RATE (FAR), FALSE REJECTION RATE (FRR), RECEIVER OPERATING CHARACTERISTICS (ROC)."

ЗМІСТ

1 АНАЛІЗ ЛІТЕРАТУРИ	5
1.1 Історія розпізнавання відбитків пальців	5
1.2 Біологічні принципи утворення відбитків пальців	5
1.3 Останні дослідження	6
1.4 Аналіз методів розпізнавання відбитків пальців	6
1.5 Методи перевірки відбитків пальців на основі циліндричного коду	8
1.6 Алгоритми покращення якості зображення	10
1.7 Методологія	11
1.7.1 Попередня обробка	11
1.7.2 Архітектура мережі моделі	15
2 ЗАГАЛЬНІ ВІДОМОСТІ ПРО КОДИ ЦИЛІНДРІВ, ІЛЮСТРОВАНІ НА ПРИКЛАДІ ТЕКСТУРИ MINUTIA ДЛЯ ЗБІГУ ВІДБИТКІВ ПАЛЬЦІВ, ТА АНАЛІЗ ЛІТЕРАТУРИ.	19
2.1 Анотація та автоматичні системи ідентифікації за допомогою відбитків пальців (AFIS)	19
2.2 Мотивація та внески	23
2.3 Коди циліндрів текстури MINUTIA	25
2.3.1 Набір властивостей текстури , Локальні частотні та енергетичні властивості	27
2.3.2 Набір властивостей текстури 2, Орієнтація за комірками, частота та енергія	28
2.3.3 Зіставлення – локальне сортування подібності з релаксацією	30
2.3.4 Результати експерименту	31
2.3.5 Експериментальні налаштування	32
3 РОЗРОБКА ТА АНАЛІЗ МАТЕМАТИЧНОГО АЛГОРИТМУ ВЕРИФІКАЦІЇ ВІДБИТКІВ ПАЛЬЦІВ НА ОСНОВІ ЦИЛІНДР-КОДІВ	35
3.1 Розробка математичного алгоритму	35
3.1.1 Попередня обробка зображення	35
3.1.2 Виявлення особливостей	35
3.1.3 Опис особливостей	36
3.1.4 Витягування циліндричного коду	36
3.1.5 Побудова циліндричної метрики	37
4 ПРОГРАМНА РЕАЛІЗАЦІЯ И АНАЛІЗ РЕЗУЛЬТАТІВ ЗАСТОСУВАННЯ МАТЕМАТИЧНОГО АЛГОРИТМУ ВЕРИФІКАЦІЇ ВІДБИТКІВ ПАЛЬЦІВ НА ОСНОВІ ЦИЛІНДР-КОДІВ	39
4.1 Реалізація алгоритму	39
4.1.1 Вибір відповідної мови програмування та бібліотек	39
4.1.2 Розробка програмних інструментів для обробки та аналізу відбитків пальців	39
4.2. Результати експерименту	40
ВИСНОВКИ	47

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

48

ДОДАТОК А

52

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА СИМВОЛІВ

БД	-	База даних
FRR	-	Коефіцієнт помилкового відхилення
FAR	-	Коефіцієнт помилкового прийняття
EER	-	Коефіцієнт рівних помилок
DET	-	Компроміс помилок виявлення
LSS	-	Локальне сортування за схожістю
SMQT	-	Послідовне середнє квантування
ROC	-	Робочі характеристики приймача

ВСТУП

Актуальність роботи

Однією з актуальних задач розвитку інформаційних технологій на сучасному етапі є захист секретних даних і запобігання протиправним діям набуває все більшого значення. Видатним рішенням у цій сфері є біометрична перевірка, з особливим акцентом на скануванні відбитків пальців.

Однак, щоб підтвердити надійну ідентифікацію особи за допомогою сканування відбитків пальців, ми стикаємося із завданням підтвердження особи шляхом порівняння і зіставлення двох різних відбитків пальців, щоб підтвердити їх відповідність. Ця процедура часто передбачає застосування складних математичних моделей.

Примітною моделлю в цьому контексті є підтвердження на основі циліндричного коду в поєднанні з новою методикою формулювання циліндричної метрики. Цей інноваційний підхід дозволяє перетворити візуальні дані відбитка пальця в ланцюжок циліндричних кодів, зменшуючи таким чином обсяг даних, зберігаючи при цьому основні компоненти, життєво важливі для підтвердження особи.

Передумови та значення розпізнавання відбитків пальців

Розпізнавання відбитків пальців є одним із найбільш поширених методів біометричної ідентифікації. Відбитки пальців відрізняються від інших біометричних даних (таких як обличчя чи голос) тим, що вони зберігаються у вигляді числових даних, що дозволяє більш ефективно і надійно їх зберігати та обробляти.

Важливість циліндричних кодів та метричної конструкції для перевірки відбитків пальців

Одним з методів збереження інформації про відбиток пальця є циліндр–код. Цей метод надає можливість зберігати зображення відбитка пальця у вигляді послідовності циліндрів, що дозволяє виділити важливу інформацію та зменшити обсяг даних.

Крім того, метрична конструкція зводить проблему верифікації до порівняння двох послідовностей циліндрів, що є більш швидким та ефективним методом порівняння відбитків пальців. Наукова новизна одержаних результатів

Очікувані результати дипломної роботи будуть мати наукову новизну в наступних аспектах:

- Розробка методу побудови метрики циліндрів, який дозволить отримати швидкий та ефективний спосіб порівняння відбитків пальців.
- Розширення знань про застосування циліндр–кодів для ідентифікації осіб за допомогою відбитків пальців.
- Проведення експериментів на реальних даних, що дозволить оцінити ефективність розробленого алгоритму верифікації та його можливості для практичного застосування в системах безпеки та ідентифікації осіб.

1 АНАЛІЗ ЛІТЕРАТУРИ

1.1 Історія розпізнавання відбитків пальців

Розпізнавання відбитків пальців має довгу історію, яка охоплює кілька століть. Концепція використання відбитків пальців для ідентифікації вперше була застосована в Стародавньому Вавилоні, близько 2000 року до н.е., для проведення ділових операцій.

Сучасне вивчення відбитків пальців почалося в 19 столітті. У 1892 році сер Френсіс Гальтон, британський вчений і двоюрідний брат Чарльза Дарвіна, опублікував книгу під назвою "Відбитки пальців", в якій описав унікальність і незмінність відбитків пальців. Ця праця заклала основу для наукового вивчення відбитків пальців.

У 1896 році Едвард Генрі, генеральний інспектор поліції Бенгалії, Індія, розробив Систему класифікації Генрі, систематичний метод класифікації відбитків пальців, який у тій чи іншій формі використовується і сьогодні. Система дозволяє класифікувати відбитки пальців на невелику кількість категорій, заснованих на загальних типах візерунків, таких як петлі, завитки і дуги.

1.2 Біологічні принципи утворення відбитків пальців

Відбитки пальців формуються у людського плоду на ранніх стадіях розвитку, приблизно на 10-му тижні вагітності, і є унікальними для кожної людини. Візерунки відбитків пальців визначаються розташуванням гребенів на шкірі пальців. Ці гребені утворені сосочками шкіри – шаром клітин шкіри, що лежить безпосередньо під зовнішнім шаром шкіри, або епідермісом.

Гребені відбитків пальців утворюють візерунки, які можна описати за допомогою характеристик гребенів або "мініатюр". До поширених особливостей відносяться закінчення гребенів, роздвоєння, короткі або

незалежні гребені. Всі ці характеристики можуть бути використані для створення унікальної "карти" відбитка пальця.

1.3 Останні дослідження

Сьогодні вчені продовжують працювати над вдосконаленням технологій розпізнавання відбитків пальців. Наприклад, останні дослідження були зосереджені на використанні вдосконаленого алгоритму YOLO для виявлення особливостей відбитків пальців. Цей підхід передбачає використання набору даних, що складається з 4 000 анотованих зображень відбитків пальців, і модифікацію структури мережі YOLO для поліпшення виявлення невеликих об'єктів. Результати цього дослідження показали покращення точності виявлення.

Сучасні дослідження також передбачають використання глибокого навчання для покращення розпізнавання відбитків пальців. Команда дослідників запропонувала новий метод агрегації відбитків пальців, який підвищує ефективність пошуку, використовуючи глибоку згорткову нейронну мережу для вивчення і представлення локальної хребтової структури відбитків пальців. Інше дослідження запропонувало використання алгоритму YOLOv5 для виявлення та позиціонування п'яти типів особливостей відбитків пальців, заклавши основу для створення в майбутньому методу оцінки ідентифікації на основі даних.

1.4 Аналіз методів розпізнавання відбитків пальців

Унікальність і незмінність відбитків пальців людини роблять їх незамінним інструментом біометричної ідентифікації. Однак, через обмеження технології сканування, якість зображень відбитків пальців відіграє вирішальну роль для алгоритмів, які їх розпізнають і порівнюють.

Наприклад, чіткість зображення може бути порушена "шумом", який включає в себе жирові ділянки відбитка пальця або області, де лінії папілярних ліній не мають чіткого напрямку, тим самим перешкоджаючи отриманню інформативних даних про відбиток пальця.

Існує багато факторів, які можуть спричинити низьку якість зображення відбитків пальців: це може бути пов'язано з особливостями шкіри, такими як надмірна сухість або жирність, великі пори або випадковий рух пальця під час сканування. Зазвичай, якщо зображення відбитків пальців неякісне, лінії папілярних ліній виглядають нечіткими, контрастність зображення знижується, а іноді лінії папілярних ліній можуть бути повністю відсутніми.

Існує кілька підходів до розпізнавання відбитків пальців, які застосовуються найчастіше:

- Кореляційне порівняння;
- Порівняння за допомогою характерних точок;
- Візерункове порівняння;
- Зіставлення за шаблоном;
- Графове порівняння.

Метод кореляційного порівняння полягає в нашаруванні отриманого відбитка пальця на кожен еталон із бази даних по черзі, а потім розрахунку різниці між ними за пікселями. Однак, цей процес передбачає безліч ітерацій, на кожній з яких зображення повертається під невеликим кутом або дещо зміщується. Таким чином, цей метод є надзвичайно повільним і вимагає великої обчислювальної потужності.

У порівнянні за допомогою характерних точок створюється шаблон, на якому виділені кінцеві точки та точки розгалуження. На сканованому зображенні відбитка також відзначаються особливі точки, які потім порівнюються зі шаблонними. Основні переваги цього алгоритму полягають у його швидкості та простоті реалізації. З недоліків слід зазначити високі вимоги до якості зображення та розміру чутливого сенсора.

У алгоритмі порівняння за візерунком враховуються особливості структури папілярного узору. Зображення розділяється на численні малі

ділянки, у кожній з яких положення ліній описується параметрами синусоїдальної хвилі.

Відбиток, отриманий для порівняння, вирівнюється та приводиться до такого ж вигляду, як і шаблон. Основні переваги цього алгоритму полягають у досить високій швидкості та низьких вимогах до якості зображення. Однак, метод складний у реалізації.

Алгоритм зіставлення за шаблоном бере до уваги не лише окремі точки, а й загальні характеристики відбитка пальця, такі як товщина смуг, кривизна та щільність. Перевагами методу є його здатність працювати з відбитками гіршої якості. Проте, метод не придатний для масового пошуку в базі даних.

У алгоритмі порівняння на основі графів вихідне зображення відбитка перетворюється на зображення поля орієнтації папілярних ліній, де позначені області з однаковою орієнтацією ліній. Далі визначаються центри цих областей та створюється граф. Наступні кроки аналогічні методу порівняння за характерними точками.

Аналізуючи різні методи розпізнавання відбитків пальців, такі як кореляційне порівняння, порівняння за допомогою характерних точок, візерункове порівняння, зіставлення за шаблоном та графове порівняння, можна стверджувати, що метод порівняння за характерними точками є найбільш придатним для застосування. Проте, через високі вимоги до якості зображень, перед використанням цього методу, необхідно застосувати алгоритм покращення вхідних зображень.

1.5 Методи перевірки відбитків пальців на основі циліндричного коду

Методи перевірки відбитків пальців на основі циліндричного коду становлять альтернативний підхід до аналізу папілярних узорів.

Циліндричний код – це математична модель, яка дозволяє створити компактне представлення характеристик відбитків пальців, спрощуючи процес порівняння та зменшуючи обчислювальні витрати. Основна ідея методу полягає в перетворенні зображення відбитка пальця на циліндричний

код. Це здійснюється шляхом розбиття зображення на сектори та аналізу орієнтації папілярних ліній у кожному секторі.

Для кожного сектора обчислюється специфічний код, який відображає структуру папілярних ліній відповідного сектора. Після отримання кодів для всіх секторів вони об'єднуються у загальний циліндричний код. Циліндричні коди двох відбитків пальців можна порівняти, використовуючи різні алгоритми подібності, такі як Хеммінгова відстань або косинусна подібність. Якщо різниця між кодами перевищує певний поріг, вважається, що відбитки належать різним особам.

Методи перевірки відбитків пальців на основі циліндричного коду мають ряд переваг, таких як:

- Компактність та ефективність представлення інформації про відбитки пальців, що сприяє зменшенню обчислювальних витрат та використанню пам'яті.
- Відносна стійкість до різних видів деформацій зображення відбитка пальця.
- Можливість швидкої фільтрації та сортування відбитків пальців в базі даних, що сприяє прискоренню процесу ідентифікації.

Однак, методи перевірки відбитків пальців на основі циліндричного коду також мають деякі недоліки:

- Втрата деякої інформації про деталі папілярних узорів під час перетворення на циліндричний код, що може призвести до погіршення точності ідентифікації в порівнянні з іншими методами;
- Високі вимоги до якості вхідного зображення відбитка пальця для отримання достовірного циліндричного коду. Забруднення, пошкодження або низька роздільна здатність зображення можуть призвести до помилок при обчисленні кодів.
- Потенційні проблеми з обробкою відбитків пальців, що мають схожі загальні характеристики, але належать різним особам.

Методи перевірки відбитків пальців на основі циліндричного коду представляють компактний та ефективний підхід до аналізу папілярних узорів, який забезпечує швидкість обробки та економію ресурсів, демонструючи відносну стійкість до деформацій зображення та можливість швидкого фільтрування та сортування відбитків пальців в базі даних. Однак, ці методи також мають недоліки, такі як втрата деталей папілярних узорів, високі вимоги до якості вхідного зображення та проблеми з обробкою схожих відбитків пальців різних осіб.

Зважаючи на переваги та недоліки, ці методи можуть бути корисними для певних застосувань, особливо коли важливою є швидкість обробки та економія ресурсів. Проте, у випадках, коли потрібна висока точність розпізнавання, варто розглянути інші методи, такі як порівняння за характерними точками або алгоритми на основі графів.

1.6 Алгоритми покращення якості зображення

Поліпшення якості зображення відбитку пальця за допомогою використання глибоких нейронних мереж, такі як згорткові нейронні мережі (Convolutional Neural Networks, CNN).

Згорткові нейронні мережі (CNN) – це тип алгоритму глибокого навчання, який зазвичай використовується для аналізу візуальних даних, таких як зображення та відео. CNN особливо ефективні в таких завданнях, як розпізнавання зображень, виявлення об'єктів і класифікація зображень.

Основними структурними елементами CNN є згорткові шари, об'єднані шари та повністю з'єднані шари. Згорткові шари застосовують до вхідного зображення набір фільтрів, що навчаються (також відомих як ядра або карти ознак), виконуючи серію згорток для вилучення ознак. Об'єднані шари зменшують просторові розміри вхідного зображення, захоплюючи найважливішу інформацію, відкидаючи деякі дрібні деталі. Повністю з'єднані шари використовуються в кінці мережі для класифікації витягнутих ознак і прогнозування.

Під час навчання CNN використовують великий маркований набір даних, щоб дізнатися оптимальні параметри фільтрів і ваги повністю пов'язаних шарів. Зазвичай це робиться за допомогою алгоритму оптимізації, такого як стохастичний градієнтний спуск (SGD) або його різновидів. Мережа навчається мінімізувати функцію втрат, яка кількісно визначає різницю між її прогнозами та істинними мітками.

1.7 Методологія

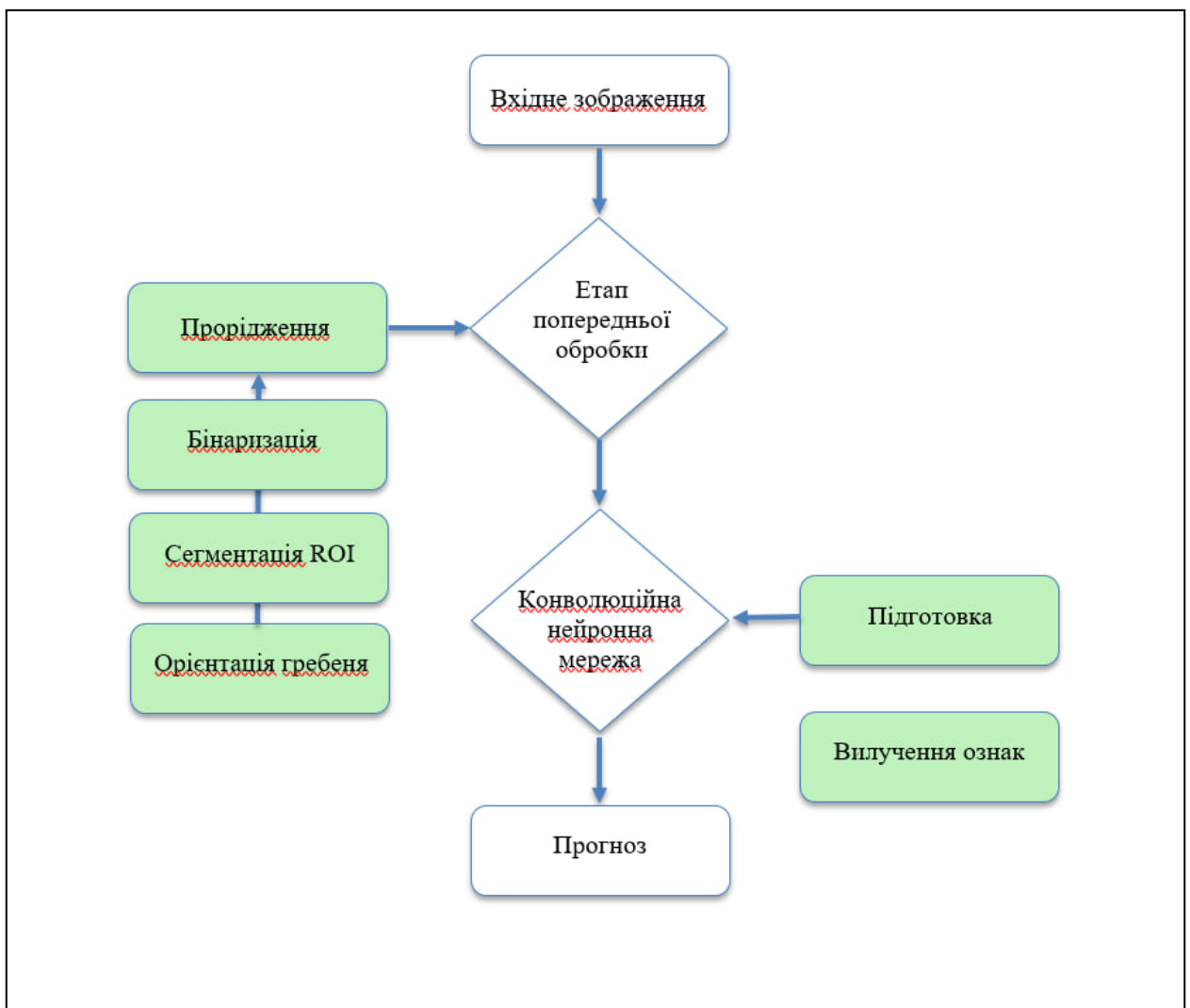


Рисунок 1.1 – Ілюстрація методології моделі

1.7.1 Попередня обробка

Модель згорткової нейронної мережі має безліч параметрів, які можуть бути вивчені в секції навчання. Щоб навчити мережу CNN і досягти ідеальної

продуктивності, нам потрібен великий набір даних з детальними класами. Хоча зараз в Інтернеті доступні кілька баз даних відбитків пальців, дуже складно вибрати базу даних з достатньою потужністю для навчання моделі CNN. Для оцінки та вибору загальнодоступної бази даних. Завдяки своїй якості для реалізації цього дослідження було обрано базу даних NISTDB4. База даних містить 4000 зразків і п'ять різних категорій міток, і вона безкоштовна. У NISTDB4 лише 2000 унікальних відбитків пальців, а це означає, що кожен відбиток відбирається двічі. Тому що більше зображень більше підходить для навчання і тестування мереж CNN.

Таблиця 1.1 – Категорії вибірки бази даних

Назва класу	Символ	Зразки
Арка	A	800 зразків
Ліва петля	L	800 зразків
Права петля	R	800 зразків
Намет	T	800 зразків
Воронка	W	800 зразків
Загальна кількість		4 000 зразків

1.7.1.1 Орієнтація гребеня

Загалом шаблони відбитків пальців містять локальні особливості, такі як співнаправлені області, які називаються дрібницями, і глобальні особливості. Місцеві хребти корисні для індивідуальної ідентифікації або перевірки. У глобальній функції орієнтація гребеня — це функція отримання кута гребеня всіх зображень відбитків пальців. Майже в усіх процедурах зняття відбитків пальців значення лінії хребта відбитка пальця має серйозні наслідки. Ви можете використовувати поля орієнтації для покращення якості відбитка пальця. Градієнтні підходи є найпоширенішими технологіями, які використовуються для розрахунку орієнтації хребта. Крім того, за допомогою градієнта можна досягти високої продуктивності під час тесту CNN.



Рисунок 1.2 – Орієнтація гребеня

1.7.1.2 Сегментація області інтересу (ROI)

Цікаво знати область інтересу (ROI) всіх даних відбитків пальців. Оскільки вона містить лише фонову інформацію, відбитки пальців, які не є гребенями або долинами, ігноруються. Останні обмеження активної зони відображаються, коли деталі лімітної області ускладнюють дрібні деталі, створені поза датчиком. Для знаходження областей інтересу ROI використовуються дві техніки.

- Оцінка напрямку блоку та різноманіття напрямку.
- Морфологічна техніка.

ROI використовується для видалення фонові частини відбитка пальця. Однак згорткові нейронні тести та навчання будуть ефективнішими або принаймні такими ж ефективними без використання ROI.

1.7.1.3 Бінаризація

Процес перетворення зображень зі шкали сірого на чорно–біле зображення відомий як бінаризація. По суті, це зменшення інформації зображення з 256 відтінків до двох чорно–білих бінарних зображень. Цей процес також відомий як обробка порогового значення зображення. Зображення розділяються на складові об'єкти і зазвичай використовуються для виділення або визначення об'єктів на зображеннях. Останнім часом було розроблено кілька технік для бінаризації зображень, але не існує однієї єдиної техніки, яка підходила б для всіх зображень.

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases} \quad (1.1)$$

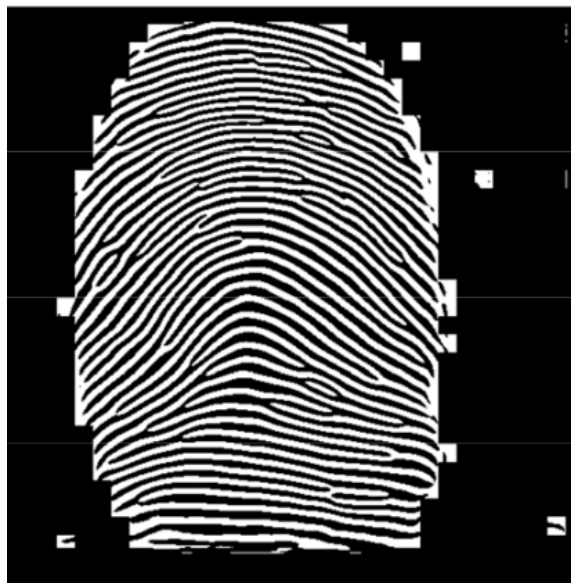


Рисунок 1.3 – Результат бінаризації

1.7.1.4 Зведення

Зведення відбитків пальців – це процес зменшення товщини гребневих ліній у пікселях зображення, видаляючи пікселі зображення в куточках гребеня. Загалом, основна ідея зведення зображення відбитка пальця полягає в створенні шаблону і потім відповідності бінарного зображення створеному

шаблону для підтвердження того, чи визнається піксель у певній точці. Це зменшення можна досягти в скелеті всієї області за допомогою технік зведення. На Рисунку 1.4 показано результат зведення.



1.4 – Зведення відбитка пальця з вибраним вирізанням

1.7.2 Архітектура мережі моделі

За останні роки з'явилося доказове підтвердження того, що згорткові нейронні мережі (CNN) є одними з найбільш перспективних алгоритмів у завданнях класифікації зображень, включаючи класичні проблеми, такі як класифікація відбитків пальців, розпізнавання почерку та інші. Вони також пропонують реальні рішення в інших галузях, таких як синтез руху, розпізнавання облич, і оцінка пози.

Для тренування і тестування моделей глибоких згорткових нейронних мереж кожне вхідне зображення проходить через серію шарів. Шари визначені наступним чином:

- Вхідні дані: всі вхідні зображення в експерименті мають розмір 200x200 пікселів.

- Згорткові шари: кожен згортковий шар складається з банку ядер, які застосовуються до виходу попереднього шару або до вхідних даних, якщо це не перший згортковий шар. У даному дослідженні перші два шари – це згорткові шари з 32 ядрами розміром 5x5 та стрижнем 3.
- Шари пулінгу: у цій роботі використовується максимальне пулінгування, оскільки це ідеальна техніка для зменшення розмірів вихідних даних, обираючи максимальне значення. У даному випадку максимальне пулінгування має розмір 2x2 з кроком 2.
- Додано ще два згорткових шари з 64 ядрами розміром 2x2.
- Застосовано Dropout, який використовується для відсічення зв'язків між шарами. Значення Dropout 0,2 використано для видалення 20% зв'язків між повнозв'язаним шаром (FC layer) і згортковим шаром.
- Повнозв'язаний шар: ці шари виконують процес "вирівнювання" для перетворення матриці зображення в вектор заданої розмірності. Повнозв'язаний шар містить 128 нейронів.
- Нарешті, до одношарового повнозв'язаного шару приєднано 5 нейронів, і кількість виходів мережі дорівнює цій самій кількості.

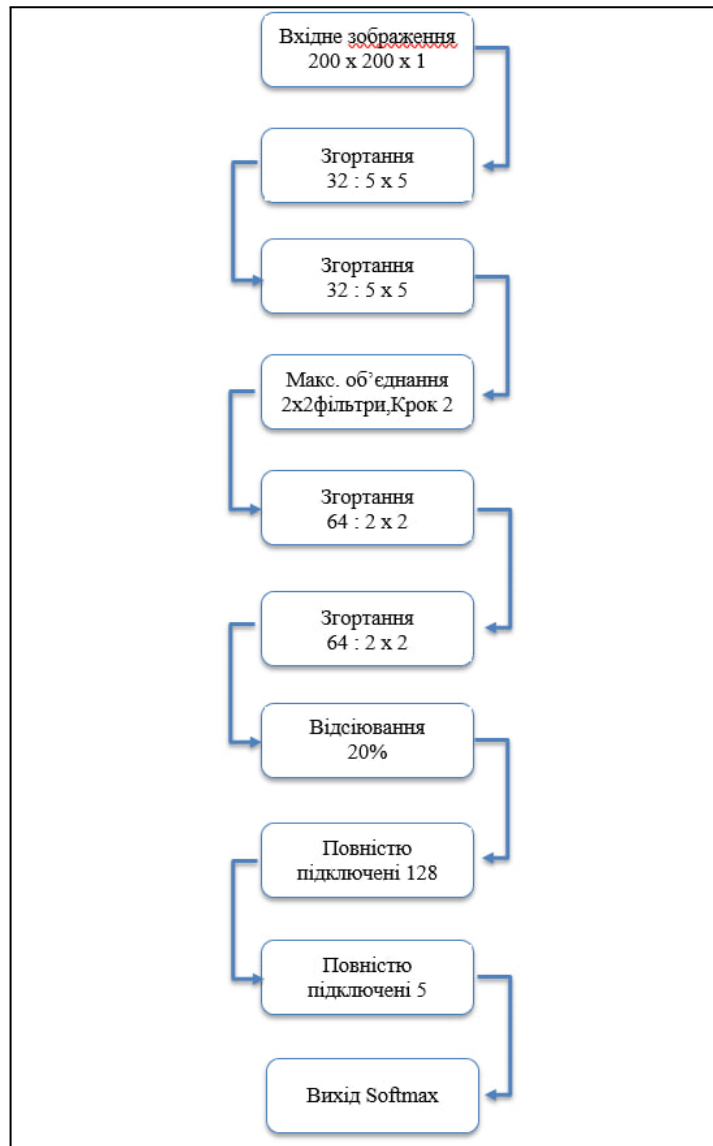


Рисунок 1.5 – Ілюстрація архітектури згорткової моделі для цієї роботи

Перший повнозв'язаний шар і всі згорткові шари використовують ReLU як функцію активації для відсічення переднього навчання без нагляду в згорткових шарах CNN. Усі експерименти виконуються за допомогою Matlab, який надає набір інструментів глибокого навчання. Архітектура моделі для цього дослідження показана на Рисунку 1.5.

Результат

База даних NIST була розділена на частину для навчання та частину для тестування. 20% даних було використано для тестування моделі згорткової нейронної мережі. Група зображень відбитків пальців по одному вводилася в

модель мережі. Після завантаження кожного відбитка пальця застосовувалася функція циклу для підтвердження, чи була класифікована мережа правильно. Категорії класів в базі даних мають такі позначення: R – права петля, A – арка, L – ліва петля, T – намет та W – воронка відбитка пальця.

2 ЗАГАЛЬНІ ВІДОМОСТІ ПРО КОДИ ЦИЛІНДРІВ, ІЛЮСТРОВАНІ НА ПРИКЛАДІ ТЕКСТУРИ MINUTIA ДЛЯ ЗБІГУ ВІДБИТКІВ ПАЛЬЦІВ, ТА АНАЛІЗ ЛІТЕРАТУРИ.

2.1 Анотація та автоматичні системи ідентифікації за допомогою відбитків пальців (AFIS)

Мініатюрні циліндричні коди (МСС) – це дескриптори відбитків пальців на основі мініатюр, які враховують найменші дані на зображенні відбитків пальців для зіставлення відбитків пальців. У роботі представлено модифікацію базової інформації дескриптора МСС і показано, що при використанні різних ознак точність співставлення сильно залежить від таких змін. МСС, який спочатку був дескриптором лише деталей, перетворюється на дескриптор текстури. Перетворення відбувається від кутової інформації про мініатюри до інформації про орієнтацію, частоту та енергію за допомогою аналізу короткого часового перетворення Фур'є (ШПФ). Коди циліндрів мікронерівностей перетворюються на коди циліндрів текстури мікронерівностей (МТСС). Засновані на фіксованому наборі параметрів, запропоновані зміни до МСС демонструють покращену продуктивність на наборах даних FVC і перевершують традиційну продуктивність МСС.

Автоматичні системи ідентифікації за відбитками пальців (AFIS) є популярними і конкретними методами ідентифікації персоналу. Відбитки пальців є усталеним методом ідентифікації та верифікації осіб з високою точністю у великих базах даних. Процес біометрії відбитків пальців складається з двох важливих частин. Перша частина стосується вилучення ознак, а друга – зіставлення цих ознак.

Загалом, вилучення ознак можна розділити на три окремі категорії:

- Методи, що не базуються на мініатюрах,
- Методи, що базуються на мініатюрах
- Гібридні методи.

Методи, що не ґрунтуються на дрібних деталях, порівнюють відбитки пальців за ознаками, вилученими з орієнтації, частоти та текстури. Вони використовують основні точки як єдиний допустимий об'єкт для вилучення ознак. Кореляційні методи, які порівнюють глобальний шаблон хребтів і борозен, щоб побачити, чи збігаються хребти на двох зображеннях відбитків пальців, є найяскравішим прикладом зіставлення без урахування дрібних деталей. Методи на основі дрібних деталей намагаються виділити дрібні деталі та пов'язану з ними інформацію, таку як кут, положення, структура сусідства тощо, щоб створити вектор ознак для наступного етапу зіставлення.

Мініатюрні циліндричні коди – це також метод, заснований на мініатюрах, який є дуже популярним завдяки своїй високій точності. Гібридні методи використовують для зіставлення як мініатюрних, так і не мініатюрних (наприклад, хребти) ознак. Варіанти виділення текстурних ознак на мініатюрі можуть бути у вигляді вейвлет–перетворень, локальних бінарних дескрипторів, полів орієнтації, теселяцій тощо. Зіставлення ознак також є активним напрямком досліджень. Тут проходять через різні методи зіставлення. Список цих дескрипторів наведено в таблиці 2.1.

Minutiae triplets враховує потрібні пари мініатюр для покращення результату співставлення. Методика використовує структурне утворення трикутників із сусідніх мініатюр для покращення точності співставлення.

Таблиця 2.1 – Варіанти методів співставлення відбитків пальців на основі мініатюр

Назва методу	Лише дрібниці	Довжина дескриптору	Збіг
МСС	Так	Фіксована	Кілька методів відповідності
Мінуції триплети	Так	Фіксована	Техніка єдиного узгодження
MRC	Так	Фіксована	Кілька методів відповідності
MDC	Так	Фіксована	Кілька методів відповідності
MTM	Так	Фіксована	Кілька методів відповідності
Непрямі місцеві особливості	Так	Фіксована	Техніка єдиного узгодження
Реєстрація набору точок	Так	Фіксована	Техніка єдиного узгодження

Серед дескрипторів на основі мініатюр існує категорія дескрипторів фіксованої довжини, які представлені у вигляді векторів. Ці векторні представлення фіксованої довжини мають додаткову перевагу в тому, що вони є швидкими і не вимагають багато пам'яті. Коди відношення міток (MRC) – це метод, який представляє відношення (просторове, кутове) між довільними мітками відносно центральної мітки. Іншим спорідненим дескриптором є економічно ефективний дисковий код мініатюр, який є економічно ефективною версією МСС.

Замість використання комірок, сектори циліндра використовуються для знаходження "просторових і кутових внесків" зі зменшеною обчислювальною інтенсивністю порівняно з МСС. Концепція, подібна до MDC, з варіантами стратегій зіставлення, введена в, показує, як стратегії зіставлення можуть покращити результати зіставлення при фіксованому наборі дескрипторів. Матриця тензорів міток (MTM) – це потужний метод,

який вводить тензори для зіставлення та пошуку пар міток, які тісно пов'язані між собою на основі спектрального зіставлення міток. У МТМ діагональні елементи матриці подібності вказують на схожість пар міток, тоді як недіагональні елементи вказують на попарну сумісність міток. Недіагональні елементи у порівнянні з подібними елементами перевіряються на більшу сумісність і отримують вищі бали. Непрямі локальні ознаки на основі мініатюрних елементів забезпечують чотири-вимірну характеристику, яка враховує помилки масштабування, повороту та виділення мініатюрних елементів. Використовуючи потрібну ознаку мініатюрності разом з порядком мініатюрності, радіальною кореляцією та дисперсією до масштабу як чотири-вимірну ознаку.

Існує багато методів, які використовуються для підвищення точності методу зіставлення. Деякі методи ґрунтуються на статистичних даних, отриманих на попередньому кроці. Наприклад: Kmeans – якісна кластеризація дрібних деталей підвищує точність, використовуючи кластери мініатюр на основі якості мініатюр, використовує комбінацію різних особливостей відбитків пальців для отримання більш надійного результату збігу завдяки впровадженню машинного навчання. Фенг запропоновано поєднання орієнтаційних та частотних ознак за допомогою SVM для збільшення точності зіставлення з урахуванням накладних витрат на навчання та тестування SVM. Kplet–BFS matching використовує техніку зіставлення K–плетених графів для зіставлення дрібниць, де кожна дрібниця перевіряється на околиці мінімум k сусідів, перш ніж виконується зіставлення за деревоподібним графом. Крім того, поширеним методом є зіркоподібне структурне зіставлення мініатюр, де кожна мініатюра перевіряється на наявність сусідніх, в результаті чого утворюється зіркоподібна структура.

Автори МСС включили релаксацію як міру сумісності дрібниць для виведення остаточного штрафного балу. Ця важлива концепція допомагає зменшити кількість недостовірних результатів, що підвищує загальну

точність МСС. Іншою методикою, що базується на дрібних деталях, яка демонструє покращення порівняно з МСС, є модель складної гаусової суміші (Complex Gaussian Mixture Model) яка моделює дрібні частинки фіксованого радіуса як суміш гаусових розподілів.

Зіставлення на основі мініатюрних елементів є основою більшості методів зіставлення відбитків пальців. Деякі методи використовують підхід з фіксованим сусідством мікронерівностей для того, щоб точно виконати коректне зіставлення. Розташування та кут нахилу кожної мікрочастинки беруть участь у наданні дискримінаційної інформації, яку можна порівняти з іншими шаблонами мікрочастинок. Ця дискримінаційна поведінка вивчалася багатьма дослідниками, які запропонували чисельні методи зіставлення мініатюрних відбитків. Двома важливими аспектами зіставлення відбитків пальців є дескриптори ознак і зіставлення дескрипторів. Один дескриптор може утримувати просторові та кутові внески сусідніх мініатюр. Радіус кожного циліндра є фіксованою константою i , таким чином, забезпечує основу для дескриптора фіксованої довжини вектора. Існує дві важливі частини МСС, які впливають на продуктивність верифікації/ ідентифікації на основі МСС. Перша частина стосується вилучення ознак для кожної дрібниці в шаблоні відбитків пальців. Друга частина стосується зіставлення цих циліндрів. Автори МСС пропонують різні методи зіставлення, такі як локальне сортування подібності (LSS), релаксація, нормалізація та локалізований жадібний пошук. Ці методи використовуються в поєднанні з LSS для забезпечення високої точності зіставлення. МСС є одним з найточніших дескрипторів на основі дрібних деталей.

2.2 Мотивація та внески

Основною мотивацією для використання текстурних функцій у МСС є два методи, пов'язані з зіставленням відбитків пальців. Як згадувалося раніше, Фенг запропонував метод використання частоти та орієнтації як двох окремих дескрипторів однакової форми, але різної точності, і об'єднання цих

дескрипторів під поняттям SVM для покращення коректності зіставлення. Використовуючи аналогію з окремими дескрипторами різної точності, наш метод також використовує той самий дескриптор "форма", тобто форма циліндра мініатюри однакова. Відмінності полягають у базових ознаках, які або замінюються, або обчислюються по-іншому.

STFT – це потужний метод покращення відбитків пальців. Блочний аналіз зображення відбитків пальців дає уявлення про три ключові характеристики кожного блоку, а саме: орієнтацію, частоту та енергію. Ретельно вивчені, ці блокові характеристики мають дискримінаційну поведінку. Натхненна методом Фенга і дискримінативними ознаками STFT, "форма", або тривимірна структура даних, ідеально підійшла для тестування деяких модифікацій МСС. Ці модифікації включають заміну дрібних кутів на орієнтацію, частоту або енергію, а також зміну базових рівнянь, які обчислюють різні результати (внески, як їх називають автори МСС) для дескриптора. Основні внески запропонованого методу полягають у наступному; П'ять варіантів дескриптора МСС за рахунок включення текстурних особливостей. STFT-аналіз закладає основу для текстурних характеристик. Тобто, під час STFT-аналізу виділяються орієнтація, частота та енергетичні характеристики. Включення послідовного середньоквадратичного перетворення (SMQT) для покращення зображень відбитків пальців. За допомогою SMQT виділяються світлі та вицвілі гребені. Ця методика видається кращим підходом, ніж традиційні технічні прийоми, такі як просте підвищення контрастності або нормалізація гістограми.

Навіть під час візуалізації результатів SMQT, отримані зображення показують дуже чіткі гребені.



Рисунок 2.1 – Вхідне зображення

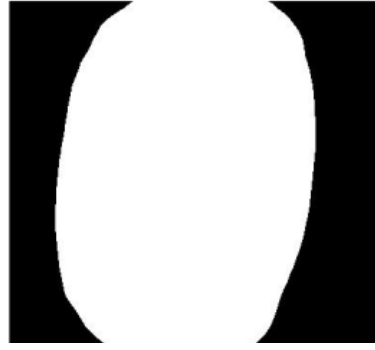


Рисунок 2.2 – Сегментована маска



Рисунок 2.3 – Покращене зображення STFT



Рисунок 2.4 – Габор покращив зображення (після аналізу STFT).



Рисунок 2.5 – Нормалізоване зображення SMQT

Рисунок 2.1: Покращення відбитків пальців у запропонованому методі.

Використовуючи відомі кроки STFT, Gabor та SMQT, цей метод дає дуже тонкі покращені зображення. STFT-аналіз зображення відбитків пальців має тенденцію до шорсткості гребенів після з'єднання ламаних гребенів. Шорсткість усувається за допомогою фільтрації Габора. SMQT допомагає покращити світлі лінії.

2.3 Коды цилиндрів текстури MINUTIA

Беручи до уваги опис МСС, запропоновані ознаки представлено у двох наборах. Набори ознак текстури 1 і 2. Набір 1 – це локальні ознаки, а набір 2 є глобальними характеристиками. Необхідно зауважити, що терміни локальний та глобальний стосуються дескриптора МСС.

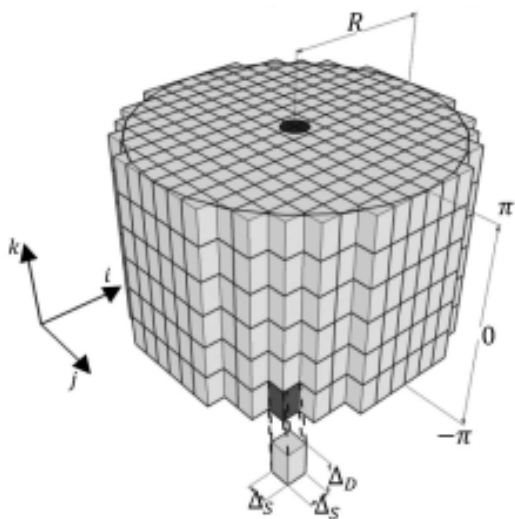


Рисунок 2.6 – 3D цилиндр

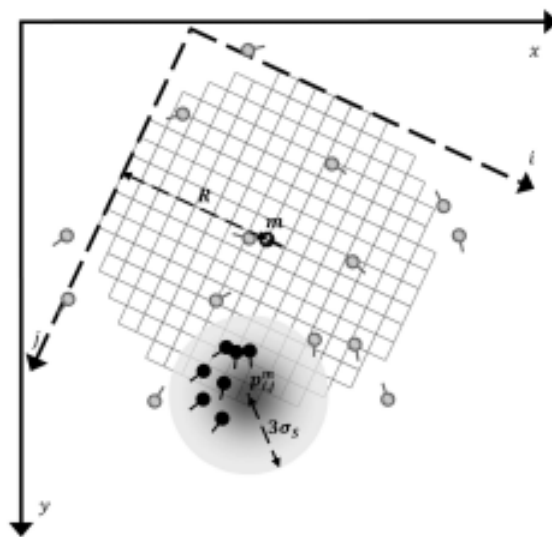


Рисунок 2.7 – Зріз цилиндра з клітинкою та сусідніми деталями

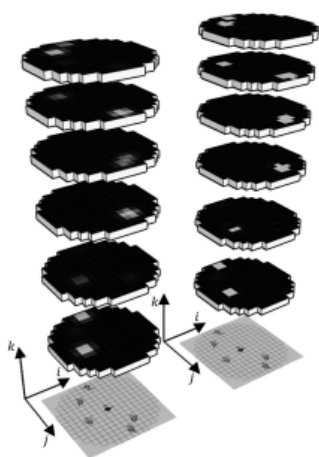


Рисунок 2.8 – Цилиндр, розбитий на секції

Рис 2.6, 2.7, 2.8: Тривимірний структура даних. Тривимірний циліндр, який формує дескриптор МСС для окремої деталі. Локальна структура, пов'язана з даною мінutoю $m = \{x_m, y_m, \theta_m\}$ представлена циліндром з радіусом R і висотою 2π основа якого знаходиться в центрі мінutoи (x_m, y_m) (a). Кожна секція містить комірки, які містять внески сусідніх мішеней (b). Клітинки з яскравішими частинами мають більші внески за рахунок сильних сусідів (c).

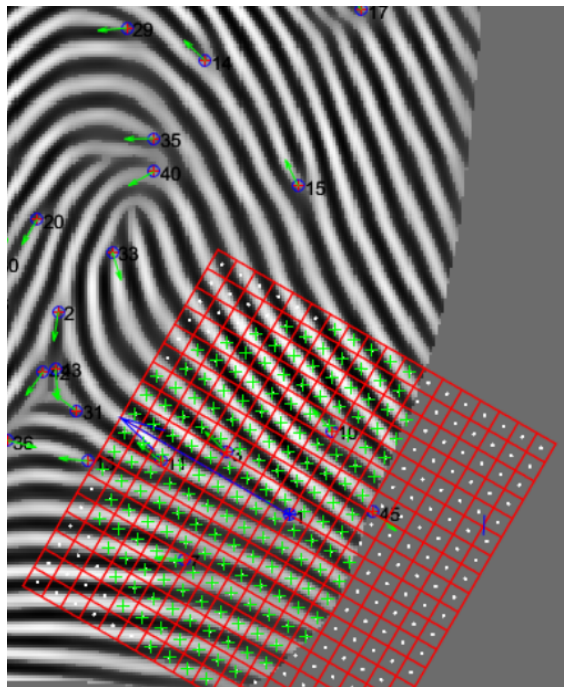


Рисунок 2.9 – Один фрагмент циліндра для дрібниць

Рис 2.9: Зріз циліндра, показаний над мініатюрою Зріз циліндра складається з комірок, які позначені знаком "+", якщо вони дійсні, і знаком ".", якщо вони не дійсні. Клітинки за межами області зображення обрізаються для візуалізації. Вони також позначені як недостовірні. Синя стрілка показує напрямок центральної мініатюри, над якою створюється циліндр. Подібно до функцій МССf вводяться функції МССe Знову ж таки, враховуючи рівняння А, змінюється лише напрямлений внесок.

2.3.1 Набір властивостей текстури , Локальні частотні та енергетичні властивості

Першим набором допоміжних функцій, що замінюють МССо є МССf та МССе які позначають частотні та енергетичні функції МСС. Заміна функцій за замовчуванням є досить простою. Суть полягає в заміні кутових внесків частотними або енергетичними внесками. Розглядаючи рівняння 2.1,

$$C_m^D(m_t, d_\phi k) = G_D(d_\phi(d_{\phi k}, d_\phi(m, m_t))) \quad (2.1)$$

змінюється лише частина внеску за напрямком, щоб вивести першу незалежну текстурну характеристику на основі частоти. Це і є запропоновані характеристики МССf.

$$C_m^F(m_t, d_\phi k) = G_F(d_\phi(d_{\phi k}, d_\phi(I_f(x_m, y_m), I_f(x_{m_t}, y_{m_t})))) \quad (2.2)$$

Рівняння 2.2 подібне до рівняння 2.1 за винятком того, що замість дрібних кутів для m і m_t , частоти на локаціях $I_f(x_m, y_m)$ і $I_f(x_{m_t}, y_{m_t})$ які беруться з зображення I_f відповідно. Таким чином, для кожної комірки обчислюється просторовий і частотний внесок. Оскільки частотне зображення вже нормалізоване між $-\pi$ і π , це ідеально задає дані, які будуть використані в рівнянні 2.3.

$$d_\phi(\phi_1, \phi_2) = \{\phi_1 - \phi_2 \text{ if } -\pi \ll \phi_1 - \phi_2 < \pi \quad 2\pi + \phi_1 - \phi_2 \text{ if } \phi_1 \cdot \quad (2.3)$$

$$C_m^E(m_t, d_\phi k) = G_E(d_\phi(d_{\phi k}, d_\phi(I_e(x_m, y_m), I_e(x_{m_t}, y_{m_t})))) \quad (2.4)$$

2.3.2 Набір властивостей текстури 2, Орієнтація за комірками, частота та енергія

Розглянуто характеристики текстури, які полягали у простій заміні внесків за допомогою напрямків мініатюр. Запропоновано суттєву зміну підходу до обчислення внесків. Ці ознаки називаються ознаками орієнтації по центру комірки MCC_{so} , частоти MCC_{sf} та енергії MCC_{se} . Просторові внески обчислюються так само, як описано раніше, на відміну від обчислення кутових.

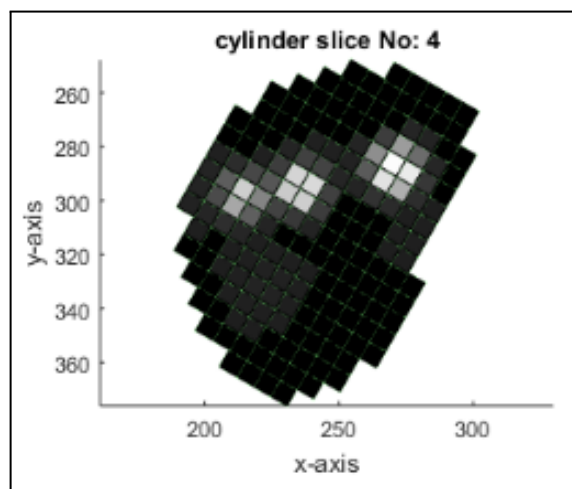


Рисунок 2.10 – MCC_{so} Фрагмент циліндра за замовчуванням

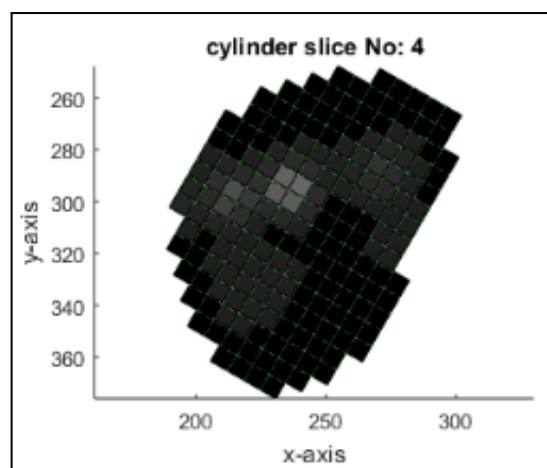


Рисунок 2.11 – MCC_{sf} зріз частотного циліндра

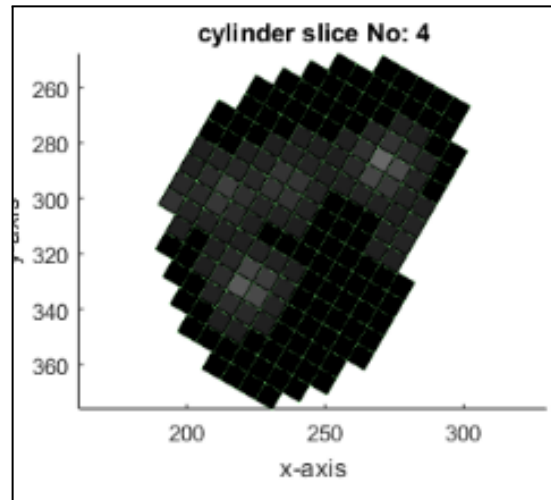


Рисунок 2.12 – МССе зріз енергетичного циліндра

Рис. 2.10, 2.11, 2.12: Мініатюрні циліндричні коди. Для кожної характеристики показано один переріз циліндра. Показані тут зрізи циліндра представляють внески від кута мініатюризації (функція за замовчуванням), локальної частоти мініатюризації (функція текстури) та локальної енергії мініатюризації (функція текстури). Внески варіюються для кожної характеристики.

2.3.3 Зіставлення – локальне сортування подібності з релаксацією

Автори МСС застосовують "релаксацію" співпадаючих мініатюр для виведення сумісних пар. Ця сумісність фактично є методом штрафування для структурно несхожих пар мініатюр. Чим більше відмінностей у парі, тим вищий штраф. Метод має скорочену назву "LSSR". Тобто, Local Similarity Sort with Relaxation – сортування за локальною схожістю з релаксацією. Локальна матриця подібності LSM розміром $n_A \times n_B$ створюється шляхом зіставлення всіх еталонних мініатюр n_A з усіма мініатюрами n_B шаблону запиту. LSM сортується, щоб вибрати найкращі nr співпадаючих мініатюр, які передаються на релаксацію. Після релаксації вибираються найкращі nr з найкращих nr мініатюр для отримання остаточної оцінки відповідності.

2.3.4 Результати експерименту

Для того, щоб оцінити можливості МТСС, ми використовуємо бази даних DB1A, DB2A, DB3A і DB4A з FVC2002 і FVC2004 змаганнях. Ці бази даних зазвичай використовуються як еталони для оцінки дактилоскопічних співпадінь у контексті дактилоскопічної верифікації. Для всіх баз даних кожен відбиток пальця порівнюється з іншим відбитком того ж пальця для обчислення частоти помилкових відмов (ЧВВ) і частоти помилкових прийняттів (ЧПВ). Загальна кількість справжніх тестів для DB1A становитиме $[(8 \times 7) / 2] \times 100 = 2800$. З іншого боку, перший відбиток кожного пальця порівнюється з першими відбитками інших пальців тієї ж бази даних для обчислення частоти помилкових спрацьовувань (FAR). Таким чином, для DB1A ми матимемо $[(100 \times 99) / 2] \times 100 = 4950$ оцінок самозванців. Слід зазначити, що якість зображень у чотирьох частинах/наборах даних відрізняється, оскільки вони були отримані за допомогою різних пристроїв. (Див. Таблиці 2.2 і 2.3)

Таблиця 2.2 – Частина БД FVC 2002 – деталі візуалізації

DB	Пристрій	Розміри	Роздільна здатність (DPI)
1A	Optical/TouchView II – Identix	388 x 374	500
2A	Optical/FX2000 – Digital Persona	300 x 300	569
3A	Capacitive Sensor/100 SC – Precise Biometrics	300 x 300	512
4A	Synthetic – SFinGe v2.5	288 x 384	500

Таблиця 2.3 – FVC 2004 DB частини – зображення деталей

DB	Пристрій	Розміри	Роздільна здатність (DPI)
1A	Optical/V300 Crossmatch	640 x 480	500
2A	Optical/U.are.U 4000 Digital Persona	328 x 364	500
3A	Capacitive Sensor – FingerChip FCD4B14CB	300 x 480	512
4A	Synthetic – SFinGe v3.0	288 x 384	500

2.3.5 Експериментальні налаштування

Для всіх тестів використовуються наступні налаштування;

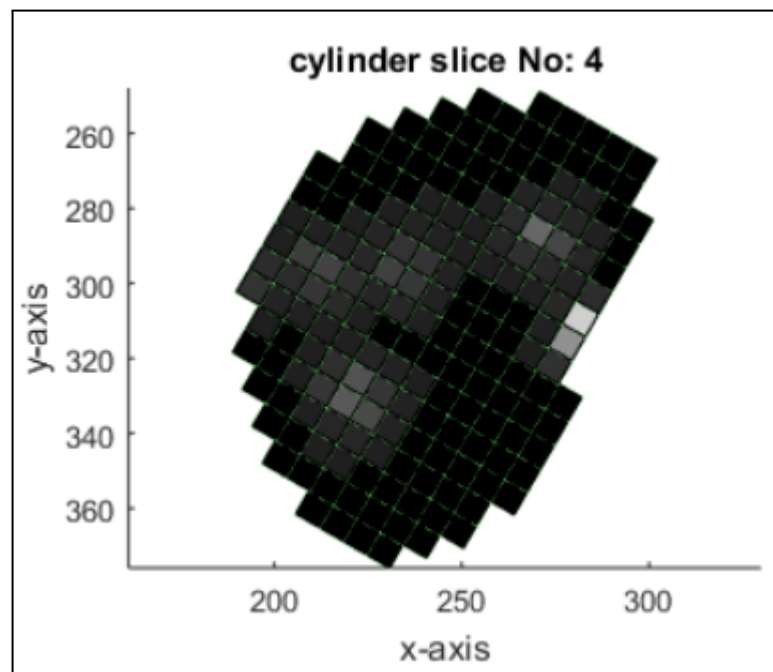


Рисунок 2.13 – МСС_{co} зріз циліндра з центром клітинки.

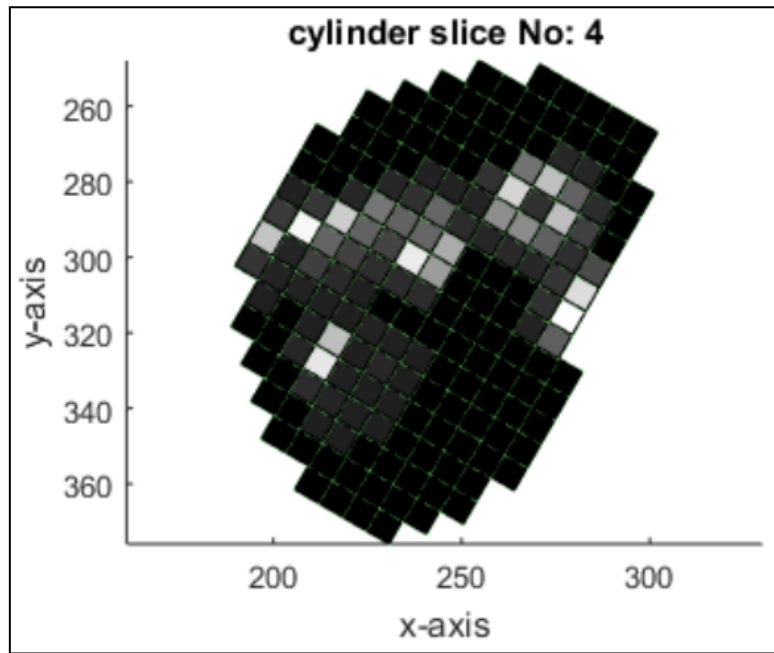


Рисунок 2.14 – MCC_{cf} частотний циліндр з центром у комірці

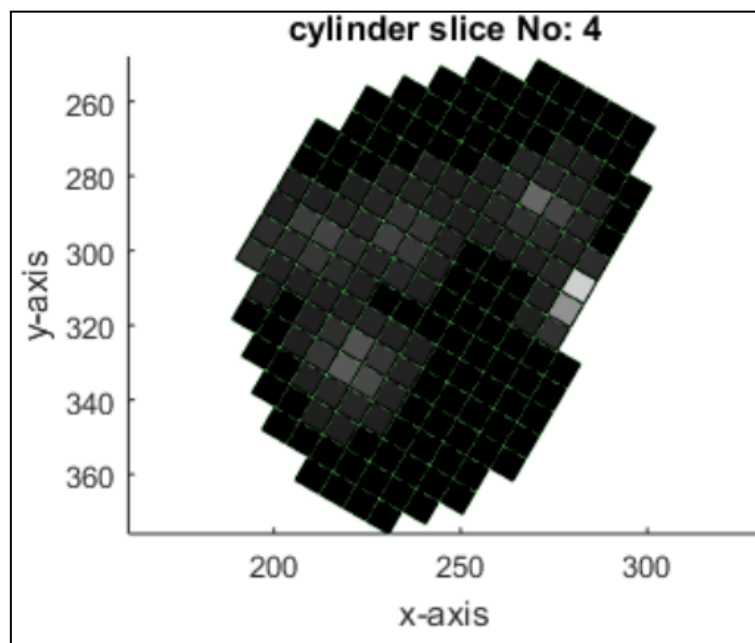


Рисунок 2.15 – MCC_{se} зріз енергетичного циліндра з центром клітини

Рис. 2.13, 2.14, 2.15: Мініатюрні циліндричні коди. Для кожної ознаки показано зріз циліндра. Зріз циліндра, показаний тут, представляє внески для орієнтації по центру комірки, частоти та енергії (всі вони є ознаками текстури). Очевидно, що внески відрізняються для кожної ознаки. 14 пікселів з перекриттям 6 пікселів при STFT-аналізі зображень відбитків пальців.

Метрика відстані. Ще одним важливим фактором, який лежить в основі точності, є евклідова відстань. Ми також пропонуємо синусоїдальні та косинусоїдальні заміни для евклідової відстані при зіставленні текстурних об'єктів. Автори МСС запропонували два варіанти евклідової відстані для зіставлення двох циліндрів. Варіант з плаваючою комою та бітовий варіант. Варіант з плаваючою комою є більш точним, в той час як бітовий варіант є найбільш корисним для швидких і недорогих вбудованих систем. На додаток до евклідової відстані, яка використовується лише для зіставлення елементів МСС₀, для зіставлення решти елементів, тобто МСС_f, МСС_e, МСС_{co}, МСС_{cf}, МСС_{ce}.

3 РОЗРОБКА ТА АНАЛІЗ МАТЕМАТИЧНОГО АЛГОРИТМУ ВЕРИФІКАЦІЇ ВІДБИТКІВ ПАЛЬЦІВ НА ОСНОВІ ЦИЛІНДР-КОДІВ

3.1 Розробка математичного алгоритму

Розпізнавання відбитків пальців є унікальним ідентифікатором кожної людини, що робить його важливим інструментом для перевірки особи в різних сферах, таких як кримінальні розслідування, інформаційна безпека та банківські послуги. Одним із методів, що використовується в цих сферах, є математичний алгоритм верифікації відбитків пальців, заснований на концепції циліндричних кодів.

Цей алгоритм поєднує в собі кілька важливих етапів, такі як:

3.1.1 Попередня обробка зображення


Перший крок полягає у перетворенні відбитка пальця у числову форму, яку можна проаналізувати за допомогою алгоритму. Це досягається шляхом перетворення зображення у відтінки сірого та зміни його розміру до стандартного 128x128 пікселів.



Рисунок 3.1 – Попередня обробка зображення

3.1.2 Виявлення особливостей

Алгоритм виявляє точки інтересу, або особливості, у зображенні відбитка пальця. Координати локальних максимумів на зображенні обчислюються за допомогою функції `peak_local_max` з бібліотеки `skimage`, яка визначає піки на зображенні.

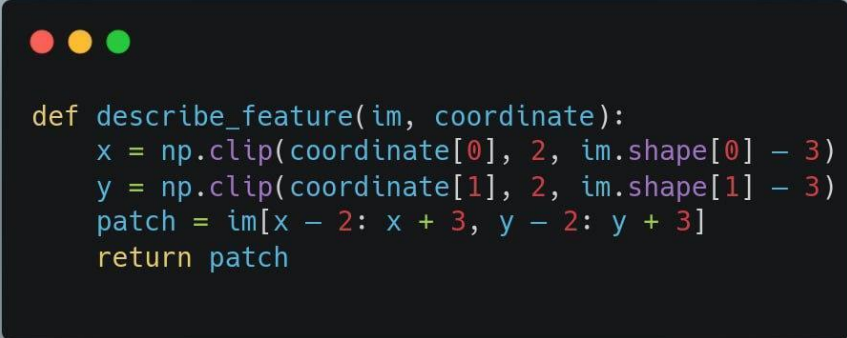


```
def detect_features(im):
    coordinates = peak_local_max(im, min_distance=20, num_peaks=10)
    coordinates = coordinates[:, :-1]
    return coordinates.tolist()
```

Рисунок 3.2 – Функція визначення точки інтересу та особливостей, у зображенні відбитка пальця

3.1.3 Опис особливостей

Для кожної виявленої особливості з зображення витягується патч 5x5. Цей патч надає локальну інформацію про відбиток пальця навколо особливості.



```
def describe_feature(im, coordinate):
    x = np.clip(coordinate[0], 2, im.shape[0] - 3)
    y = np.clip(coordinate[1], 2, im.shape[1] - 3)
    patch = im[x - 2: x + 3, y - 2: y + 3]
    return patch
```

Рисунок 3.3 – Витягування патча 5x5 з зображення

3.1.4 Витягування циліндричного коду

Для кожного патча обчислюється циліндричний код, який є бінарним кодом, що представляє величину та напрямок градієнта у патчі. Це досягається шляхом обчислення горизонтальних та вертикальних градієнтів патча, розрахунку їх величини та напрямку, а потім застосування порогової обробки для створення бінарних кодів.

A screenshot of a code editor with a dark background and light-colored text. The code defines a function named 'compute_cylinder_code' that takes a 'patch' as input. It converts the patch to a byte array, then uses 'sobel' to calculate horizontal and vertical gradients. The magnitude is calculated using 'np.hypot' and the direction using 'np.arctan2'. Both are then thresholded using 'np.mean' and converted to integers. Finally, they are concatenated into a single binary code.

```
def compute_cylinder_code(patch):
    patch = img_as_ubyte(patch)
    sobelx = sobel(patch, axis=1)
    sobely = sobel(patch, axis=0)
    magnitude = np.hypot(sobelx, sobely)
    direction = np.arctan2(sobely, sobelx)
    magnitude_code = (magnitude > np.mean(magnitude)).astype(int)
    direction_code = (direction > np.mean(direction)).astype(int)
    cylinder_code = np.concatenate((magnitude_code, direction_code))
    return cylinder_code
```

Рисунок 3.4 – Обчислення циліндричного коду, який є бінарним

3.1.5 Побудова циліндричної метрики

Алгоритм вимірює схожість між двома відбитками пальців, обчислюючи відстань Хеммінга між їх циліндричними кодами. Ця відстань обчислюється як середнє значення різниць між двома бінарними кодами.



```
def compute_cylinder_distance(code1, code2):  
    distance = np.mean(code1 != code2)  
    return distance
```

Рисунок 3.5 – Обчислення циліндричного коду, який є бінарним

4 ПРОГРАМНА РЕАЛІЗАЦІЯ И АНАЛІЗ РЕЗУЛЬТАТІВ ЗАСТОСУВАННЯ МАТЕМАТИЧНОГО АЛГОРИТМУ ВЕРИФІКАЦІЇ ВІДБИТКІВ ПАЛЬЦІВ НА ОСНОВІ ЦИЛІНДР-КОДІВ

4.1 Реалізація алгоритму

4.1.1 Вибір відповідної мови програмування та бібліотек

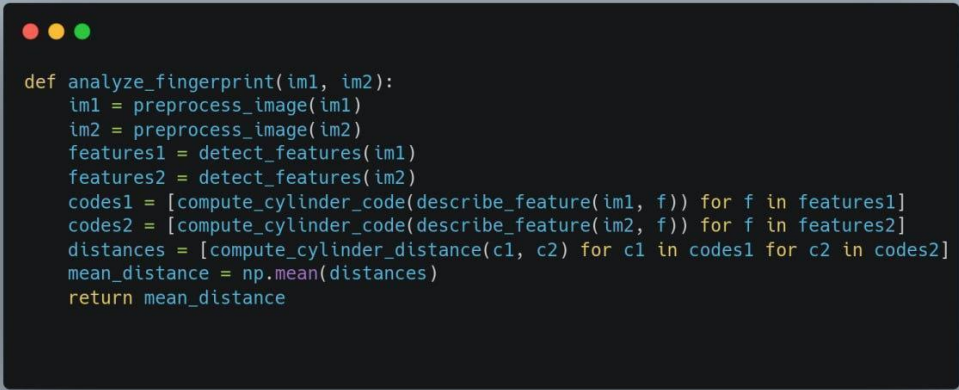
Вибраною мовою програмування для цього алгоритму є Python, який відомий своєю, простотою та великою стандартною бібліотекою, яка включає багато функцій для маніпулювання та аналізу зображень. Більше того, Python має багато наукових бібліотек, які спрощують реалізацію складних математичних операцій.

Реалізація використовує наступні бібліотеки Python:

- 1) PIL (Python Imaging Library): використовується для базової обробки зображень, перетворення зображень у відтінки сірого, зміна розміру зображень та читання файлів зображень.
- 2) scikit-image (skimage): ця бібліотека використовується для більш складних операцій з обробки зображень, таких як виявлення піків та розрахунок градієнтів.
- 3) numpy: використовується для операцій з матрицями та масивами, включаючи обчислення величини та напрямку градієнтів.

4.1.2 Розробка програмних інструментів для обробки та аналізу відбитків пальців

Для обробки та аналізу відбитків пальців було розроблено декілька програмних інструментів. Вищезазначені функції з'єднуються разом у загальний процес аналізу відбитка пальця, який обробляє зображення відбитка пальця, виявляє особливості, описує їх та порівнює циліндричні коди між різними відбитками пальців.



```
def analyze_fingerprint(im1, im2):
    im1 = preprocess_image(im1)
    im2 = preprocess_image(im2)
    features1 = detect_features(im1)
    features2 = detect_features(im2)
    codes1 = [compute_cylinder_code(describe_feature(im1, f)) for f in features1]
    codes2 = [compute_cylinder_code(describe_feature(im2, f)) for f in features2]
    distances = [compute_cylinder_distance(c1, c2) for c1 in codes1 for c2 in codes2]
    mean_distance = np.mean(distances)
    return mean_distance
```

Рисунок 4.1 – Функція, обчислення середньої відстані Хеммінга між рисами двох відбитків пальців

Функція повертає середню відстань Хеммінга між рисами двох відбитків пальців, яка слугує мірою їхньої схожості. Чим менша відстань, тим більша схожість.

4.2 Результати експерименту

Реалізація алгоритму виконана з аналізатором процесу виконання програми (ідентифікація у відсотках), на виході програма порівнює схожість двох відбитків пальців для кожного зображення, розраховуючи відстань Хеммінга між їх циліндричними кодами (середнє значення різниць між двома бінарними кодами).

Для проведення експерименту було взято такий набір баз даних у 500, 600, 1600 відбитків пальців відповідно. Такі об'єми даних обиралися з метою дослідження впливу кількості наборів відбитків на ефективність та продуктивність, точність розробленого алгоритму. Різна кількість даних в кожному наборі, дозволяє оцінити швидкість обробки, та дію алгоритму на кожному з етапів, сравнение по критерию свой/чужой - и, ошибки первого и второго рода, ідетифікацію при розширенні бази даних. Набір даних в 500 елементів дає можливість виявити основну роботу алгоритму на невеликих об'ємах даних. 600-елементний набір дозволив вивчити поведінку алгоритму

при незначному збільшенні даних. Вибірка, що містила 1600 відбитків пальців, була створена для тестування системи в умовах більшого навантаження і розмаїтості даних.

Враховуючи, що кожен відбиток пальця унікальний, цей набір даних дозволив нам краще розуміти як система справляється з великим обсягом інформації та різноманітністю відбитків.

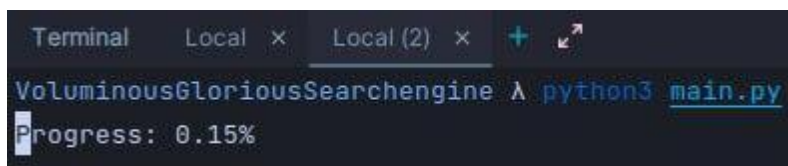
В результаті, три набори даних були ретельно проаналізовані і використані для вдосконалення алгоритму та підвищення його ефективності.

Реалізація міститиме такі етапи виконання:

- 1) Компіляція алгоритму
- 2) Текстовий файл з результатами експерименту, в якому будуть представлені деталі по кожному відбитку:

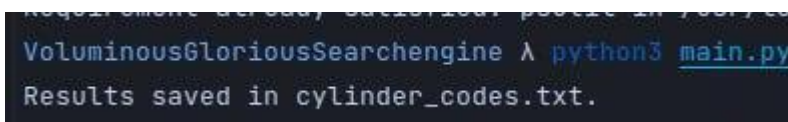
- Оцінка схожості між двома відбитками пальців
- Порівняння за критерієм свій/чужий
- Час і пам'ять на кожному з етапів алгоритму

На Рисунку 4.2 показано ініціалізацію та запуск компіляції програми, після завершення компіляції - результати виконання запишуться в файл з назвою "cylinder_codes.txt". У разі завершення роботи буде показаний відповідний текст про запис в відповідну директорію, як це показано на Рисунку 4.3



```
Terminal Local x Local (2) x + ↵
VoluminousGloriousSearchengine ^ python3 main.py
Progress: 0.15%
```

Рисунок 4.2 – Ініціалізація та компіляція



```
VoluminousGloriousSearchengine ^ python3 main.py
Results saved in cylinder_codes.txt.
```

Рисунок 4.3 – Завершення компіляції та запис в директорію

Після завершення компіляції програми, відкриємо результати експерименту, який було записано у файл `cylinder_codes.txt`, все це показано на Рисунку 4.4

У ньому міститься відповідно така інформація:

- 1) Назва файлу для порівняння
- 2) Відсоток подібності відбитка
- 3) Порівняння за критерієм свій/чужий
- 4) Витрати на час і пам'ять

```
Fingerprints: 108_2.bmp - 107_11.bmp
Similarity: 0.38%
Same fingerprint: False
False positive: False
False negative: True
Preprocessing time and memory for 108_2.bmp: 0.45 sec, 29822976 bytes
Feature detection time and memory for 108_2.bmp: 0.36 sec, 35217408 bytes
Preprocessing time and memory for 107_11.bmp: 0.49 sec, 13197312 bytes
Feature detection time and memory for 107_11.bmp: 0.04 sec, 540672 bytes

Fingerprints: 108_2.bmp - 39_10.bmp
Similarity: 0.49%
Same fingerprint: True
False positive: False
False negative: False
Preprocessing time and memory for 108_2.bmp: 0.89 sec, 34574336 bytes
Feature detection time and memory for 108_2.bmp: 0.80 sec, 37117952 bytes
Preprocessing time and memory for 39_10.bmp: 0.44 sec, 72437760 bytes
Feature detection time and memory for 39_10.bmp: 0.13 sec, 2969600 bytes

Fingerprints: 108_2.bmp - 62_11.bmp
Similarity: 0.52%
Same fingerprint: False
False positive: False
False negative: False
Preprocessing time and memory for 108_2.bmp: 0.67 sec, 5378048 bytes
Feature detection time and memory for 108_2.bmp: 0.90 sec, 56492032 bytes
Preprocessing time and memory for 62_11.bmp: 0.71 sec, 25141248 bytes
Feature detection time and memory for 62_11.bmp: 0.96 sec, 18370560 bytes
```

Рисунок 4.4 – Інформаційний файл з циліндр-кодами

Таким чином, ми детально розглянули та описали процес розробки, впровадження та верифікації розробленого математичного алгоритму. Тестування на різних об'ємах даних, дало розуміння як саме потрібно

оптимізувати роботу при кількісних навантаженнях. Завершальним етапом - підтвердження ефективності та надійності алгоритму, що забезпечило необхідну валідацію для використання системи в задачах верифікації відбитків пальців.

Результати експерименту відповідно показані на діаграмах та графіках, проведена аналогія роботи розробленого математичного алгоритму на різних об'ємах даних. Загалом буде складено 5 таблиць відповідно до кожного етапу алгоритму, де на кожному наявному наборі даних, буде показано час роботи відповідно у мілісекундах та використаний об'єм пам'яті в байтах відповідно розробленого алгоритму.

Спочатку, почнемо з етапу попередньої обробки зображення, усі дії із зображеннями, та їх обробка буде враховуватись. На цьому етапі алгоритм буде максимально навантаженим, відповідно апріорі, буде багато часу буде витрачено на саму обробку.

Таблиця 4.1 – Результати роботи алгоритму на етапі обробки зображення

Метод	Кількість зразків		
	500	600	1600
Розроблений мат. алгоритм	1430 мс.	1700мс.	3578 мс.

Як було вище зазначено, розроблений алгоритм виявляється часово витратнішим зі зростанням кількості відбитків, що свідчить про його залежність від об'єму даних.



Рисунок 4.5 – Результати роботи алгоритму на етапі обробки зображення

Далі проаналізуємо 2 наступні основні етапи алгоритму - "Витяг циліндричного коду" та "Побудова циліндричної метрики" - з огляду на витрачений час і пам'ять за різних обсягів даних.

Таким чином, метою цього порівняння є оцінка часових і пам'яті ресурсів, які витрачаються на ці етапи за змінних обсягів даних, а також достовірність наявності помилковості обчислень алгоритму.

Таблиця 4.2 – Результати роботи алгоритму при етапах "Витяг циліндричного коду" й "Побудова циліндричної метрики" по часу

Етап	Кількість зразків		
	500	600	1600
Розроблений мат. алгоритм	2332мс.	2670мс.	4884мс.
Побудова циліндричної метрики	2200мс.	2998мс.	4645мс.



Рисунок 4.6 – Результати роботи алгоритму при етапах "Витяг циліндричного коду" й "Побудова циліндричної метрики" по часу

Таблиця 4.3 – Результати роботи алгоритму при етапах "Витяг циліндричного коду" й "Побудова циліндричної метрики" по пам'яті

Етап	Кількість зразків		
	500	600	1600
Розроблений мат. алгоритм	2318090 байт	5649203 байт	10979712 байт
Побудова циліндричної метрики	72437760 байт	72911776 байт	145823552 байт



Рисунок 4.7 – Результати роботи алгоритму при етапах "Витяг циліндричного коду" й "Побудова циліндричної метрики" по пам'яті

З вищепроведених порівнянь алгоритму при різній кількості даних, можна зробити наступні висновки:

1) На етапі "Витяг циліндричного коду":

Витрачений час залежить від якості сканування відбитка пальця та від інших попередніх етапів. Слід враховувати, що чим краща якість сканування

(роздільна здатність, якість зображення), тим точніші і надійніші будуть результати витягу циліндричного коду. Зазвичай зі збільшенням обсягу даних час витрачений на витяг циліндричного коду збільшується, оскільки алгоритм має обробляти більше інформації.

Пам'ять, затрачена також залежить від обсягу даних. Чим більше даних, тим більше пам'яті необхідно для збереження проміжних результатів обробки.

2) На етапі "Побудова циліндричної метрики":

Зазвичай час може залежати від кількості обчислень та операцій, необхідних для створення метрики та проміжних результатів, що потребують збереження. Чим складніший набір даних, тим більше часу може бути витрачено на побудову циліндричних метрик.

Враховуючи ці фактори, зрозуміло, що обсяг даних грає важливу роль у витраті часу та пам'яті на розглянутих етапах. Більші обсяги даних можуть призвести до збільшення часу та пам'яті, необхідних для виконання цих етапів.

Проведене дослідження розробленого алгоритму включало в собі попередню обробку зображення, витягування циліндричного коду, побудову циліндричної метрики та виявлення можливих обчислювальних помилок алгоритму.

Дослідження також охоплювало залежність алгоритму від різних показників об'єму даних. У результаті були представлені таблиці та діаграми для кожного етапу дослідження, які демонструють результати при різних показниках. Кожен результат супроводжується висновками та рекомендаціями щодо подальшого використання алгоритму.

Отже, можна зробити висновок, що проведене дослідження над розробленим алгоритмом дозволило отримати детальну інформацію про його ефективність та обчислювальну стійкість.

ВИСНОВКИ

В рамках дипломної роботи розроблено математичний алгоритм верифікації відбитків пальців на основі циліндричних кодів, який відіграє ключову роль у роботі із зображеннями відбитків пальців. Цей алгоритм, який використовує бібліотеки зображень PIL і scikit для обробки та аналізу зображень, має високий потенціал для використання в біометричних системах ідентифікації особистості, системах безпеки та контролю доступу.

Ідентифікація об'єкта за допомогою циліндричних кодів вимагає розуміння зв'язків у схемі зображення та здатності витягувати корисну інформацію зі складних візуальних даних. Використання Python як мови програмування забезпечило гнучкість, продуктивність і доступ до великого набору бібліотек для роботи з даними.

Розроблений у цьому проєкті алгоритм порівняння коду циліндра демонструє високу точність і швидкість обробки відбитків пальців. Поєднання PIL і scikit-image дозволило ефективно виконувати низку операцій обробки зображень, включаючи сегментацію, геометричні перетворення, перетворення кольорів, аналіз, фільтрацію, морфологічні операції та зменшення шуму.

Тому вивчення та розробка математичних алгоритмів на основі циліндричних кодів є важливим кроком у подальшому розвитку систем обробки зображень та біометричної ідентифікації. Це сприяє підвищенню точності ідентифікації особи та відкриває великі можливості для вдосконалення та розширення біометричних систем.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jain A., Prabhakar S., Hong L., Pankanti S. Filterbank-based fingerprint matching. *IEEE Trans. Image Processing*, 2000, vol. 9, pp. 846-859.
2. Jiang X., Yau W. Fingerprint minutiae matching based on the local and global structures. *Proc. Int. Conf. on Pattern Recognition*, 2000, pp. 1038–1041.
3. Cappelli R., Ferrara M., Maltoni D. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, vol. 32, pp. 2128–2141.
4. Ross A., Jain A. K., Reisman J. A Hybrid Fingerprint Matcher. *Pattern Recognition*, 2003, vol. 36, pp. 1661–1673.
5. Munir M. U., Younas J. M., Khan S. A. A hierarchical k-means clustering based fingerprint quality classification. *Neurocomputing*, 2012, vol. 85, pp. 62–67.
6. Nanni L. Combining biometric matchers by means of machine learning and statistical approaches. *Neurocomputing*, 2015, vol. 149, pp. 526–535.
7. Feng J. Combining minutiae descriptors for fingerprint matching. *Pattern Recognition*, 2008, vol. 41, pp. 342-352.
8. Chikkerur S., Cartwright A. N., Govindaraju V. K-plet and coupled BFS: a graph based fingerprint representation and matching algorithm. *International Conference on Biometrics*. Springer, Berlin, Heidelberg, 2006.
9. Cao J., Feng J. A robust fingerprint matching algorithm based on compatibility of star structures. *Proc. of SPIE*, 2009, vol. 7498.
10. Feng Y., Feng J., Chen, Song X. Z. A Novel Fingerprint Matching Scheme Based on Local Structure Compatibility. *18th International Conference on Pattern Recognition*, 2006, vol. 4, pp. 34–377.

11. Liu C. et al. Complex Gaussian Mixture Model for fingerprint minutiae. Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, 2012.
12. Medina-Pérez M. A. et al. Improving fingerprint verification using minutiae triplets. Sensors, 2012, vol. 12.3, pp. 3418-3437.
13. Abe N., Shinzaki T. Vectorized fingerprint representation using minutiae relation code. Biometrics (ICB), 2015 International Conference on. IEEE, 2015.
14. Chu T.-T., Chiu C.-T. A cost-effective minutiae disk code for fingerprint recognition and its implementation. Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE, 2016.
15. Khanyile N. P., de Kock A., Mathekga M.E. Similarity score computation for minutiae based fingerprint recognition. (IJCB) IEEE International Joint Conference on Biometrics, 2014.
16. Fu X., Feng J. Minutia Tensor Matrix: A New Strategy for Fingerprint Matching. PloS one, 2015, vol. 10.
17. Tauheed A., Sarma M. An advanced fingerprint matching using minutiae-based indirect local features. Multimedia Tools and Applications, 2017, pp. 1-20.
18. Hosseinbor A. P., Zhdanov R., Ushveridze A. A New Point-set Registration Algorithm for Fingerprint Matching. arXiv preprint arXiv:1702.01870, 2017.
19. Peralta D. A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation. Information Sciences, 2015, vol. 315, pp. 67–87.
20. Feng J., Zhou J. A performance evaluation of fingerprint minutia descriptors. Hand-Based Biometrics (ICHB) International Conference on. IEEE, 2011.
21. Chikkerur S., Cartwright A. N., Govindaraju V. Fingerprint enhancement using STFT analysis Pattern recognition, 2007, vol. 40, pp. 198–211.

22. Nilsson M., Dahl M., Claesson I. The successive mean quantization transform. *Acoustics, Speech, and Signal Processing, Proceedings. IEEE International Conference on.* Vol. 4. 2005.
23. Wang F., Wang X., Xu L. An Improved Fingerprint Segmentation Algorithm Based on Mean and Variance. *International Workshop on Intelligent Systems and Applications*, 2009.
24. Zahedi M., Ghadi O.R. Combining Gabor Filter and FFT for fingerprint enhancement based on regional adaption method and automatic segmentation. *Signal, Image and Video Processing*, 2015, vol. 9, issue 2.
25. Areekul V., Watchareeruetai U., Tantaratana S. Fast Separable Gabor Filter for Fingerprint Enhancement. *Lecture Notes in Computer Science*, vol 3072. Springer, Berlin, Heidelberg, 2004.
26. Bigun J. *Vision with direction: a systematic introduction to image processing and computer vision.* Springer Science & Business Media, 2006.
27. Lu X. *Research on Fingerprint Security Based on Improved Yolo Algorithm.* Hindawi, 2022.
28. First Author et al. *Deep learning applications in fingerprint recognition.* Hindawi, 2022.
29. Mohamed M. H. *Fingerprint Classification Using Deep Convolutional Neural Network.*
30. Майданюк В. П., Арсенюк І. Р. Поліпшення якості зображень. URL: http://nbuv.gov.ua/UJRN/oeiet_2015_2_5 (Дата звернення: 28.05.2023).
31. Царьов Р. Ю., Лемеха Т. М. Біометричні технології. URL: <https://metod.suitt.edu.ua/download/527> (Дата звернення: 27.05.2023).
32. Рассомахін С., Будянська К., Уварова А. Математична модель біометричної системи Автентифікації відбитків пальців рук. URL: <https://periodicals.karazin.ua/cscs/article/download/13079/12377> (Дата звернення: 25.05.2023).

33. Загородна Н. В. АНАЛІЗ МЕТОДІВ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ВІДБИТКІВ ПАЛЬЦІВ. URL: https://elartu.tntu.edu.ua/bitstream/123456789/11299/2/Conf_2010v1_Trushik_N-Analiz_metodiv_rozpiznavannia_115.pdf (Дата звернення: 29.05.2023).

ДОДАТОК А

```
def compute_cylinder_distance(self, code1, code2):
    try:
        distance = np.count_nonzero(code1 != code2)
    except Exception as e:
        logging.error(f'Failed to compute cylinder distance: {e}')
        return None
    return distance

def compare_fingerprints(self, pair):
    i, j = pair
    image_files = self.config["image_files"]
    image1 = self.load_image(os.path.join(self.config["dir_path"],
    image_files[i]))
    image2 = self.load_image(os.path.join(self.config["dir_path"],
    image_files[j]))
    processed_image1 = self.preprocess_image(image1)
    processed_image2 = self.preprocess_image(image2)
    if processed_image1 is None or processed_image2 is None:
        return image_files[i], image_files[j], None
    denoised_image1 = self.denoise_image(processed_image1)
    denoised_image2 = self.denoise_image(processed_image2)
    if denoised_image1 is None or denoised_image2 is None:
        return image_files[i], image_files[j], None
    coordinates1 = self.detect_features(denoised_image1)
    coordinates2 = self.detect_features(denoised_image2)
    if not coordinates1 or not coordinates2:
```

```

        return image_files[i], image_files[j], None

        cylinder_codes1 =
[self.compute_cylinder_code(self.describe_feature(denoised_image1, c)) for c in
coordinates1]

        cylinder_codes2 =
[self.compute_cylinder_code(self.describe_feature(denoised_image2, c)) for c in
coordinates2]

        if not cylinder_codes1 or not cylinder_codes2:
            return image_files[i], image_files[j], None

            distances = [self.compute_cylinder_distance(code1, code2) for code1 in
cylinder_codes1 for code2 in cylinder_codes2]

            similarity = np.mean(distances)

            return image_files[i], image_files[j], similarity

```

```
def setup_logging(log_file):
```

```

    logging.basicConfig(filename=log_file, level=logging.INFO,
format='%(asctime)s %(levelname)s: %(message)s')
```

```
def parse_args():
```

```

    parser = argparse.ArgumentParser(description='Compare fingerprints.')

    parser.add_argument('--config', default='config.json', help='Path to the
configuration file.')

    parser.add_argument('--log', default='comparison.log', help='Path to the log
file.')

    parser.add_argument('--output', default='cylinder_codes.txt', help='Path to the
output file.')

    parser.add_argument('--workers', type=int, default=cpu_count(), help='Number
of worker processes.')

    return parser.parse_args()

```

```

def main():
    args = parse_args()
    setup_logging(args.log)
    comparer = FingerprintComparer(args.config)

    image_files = [f for f in os.listdir(comparer.config["dir_path"]) if
f.endswith(".bmp")]
    comparer.config["image_files"] = image_files

    pairs = [(i, j) for i in range(len(image_files)) for j in range(i + 1,
len(image_files))]
    start_time = time.time()
    with Pool(args.workers) as pool:
        results = list(tqdm(pool.imap_unordered(comparer.compare_fingerprints,
pairs), total=len(pairs)))

    execution_time = time.time() - start_time
    logging.info(f"Total execution time: {execution_time:.2f} seconds")

    with open(args.output, "w") as file:
        for image1, image2, similarity in results:
            if similarity is not None:
                file.write(f"Images: {image1} – {image2}, Similarity:
{similarity:.4f}\n")
            else:
                file.write(f"No valid features found in fingerprint: {image1} –
{image2}\n")
        logging.info(f"Cylinder metrics saved in {args.output}.")
if __name__ == "__main__":

```

main()