Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Пояснювальна записка
до магістерської кваліфікаційної роботи
«Магістр»
(освітньо-кваліфікаційний рівень)

на тему: **Підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення**

Виконав студент ІІ курсу, групи УБ – 20м
Спеціальність 125 – «Кібербезпека»
Освітня програма – «Управління інформаційною безпекою»
Богачук Вікторія Володимирівна

Керівник: к.т.н., проф. каф. МБІС Азарова А. О.
«__»_____ 2021р.

Опонент: к.т.н., доц. каф. ОТ Савицька Л.А.
«__»_____ 2021р.

Допущено до захисту
Голова секції УБ кафедри МБІС
д.т.н., проф. Яремчук Ю.Є.
“_____”_____2021 р.

Вінниця – 2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

АНОТАЦІЯ

У ході виконання роботи було проаналізовано теоретичні засади біометричної ідентифікації в системах, досліджено переваги та недоліки існуючих підходів, зокрема за образом обличчя та відбитками пальців.

У магістерській дипломній роботі вдосконалено метод фільтрування зображення засобами технології Face ID на основі фільтра Габора, що уможливило підвищення достовірності ідентифікації користувачів.

На основі запропонованого автором удосконаленого методу фільтрування зображення було розроблено відповідний програмний засіб, що уможливив комплексну ідентифікацію осіб за їх біометричними даними – рисами обличчя. Його реалізація здійснювалася мовою програмування C# у середовищі Visual Studio.

Ключові слова: біометрична ідентифікація, образ обличчя, розпізнавання, фільтр Габора, бібліотека Open CV.

SUMMARY

In the course of the work the theoretical bases of biometric identification in systems were analyzed, the advantages and disadvantages of the existing approaches were investigated, in particular by the image of the face and fingerprints.

In the master's thesis, the method of image filtering by means of Face ID technology based on the Gabor filter was improved, which made it possible to increase the reliability of user identification.

On the basis of the improved method of image filtering proposed by the author, an appropriate software tool was developed, which enabled the complex identification of persons by their biometric data - facial features. It was implemented in the C # programming language in Visual Studio.

Keywords: biometric identification, facial image, recognition, Gabor filter, Open CV library.

ЗМІСТ

ВСТУП

Актуальність. Розпізнавання мови, друкарського і рукописного тексту, різних зображень значно спрощує взаємодію людини з комп'ютером, створює передумови для застосування різних систем штучного інтелекту. Багато операцій пов’язаних з процесами автентифікації та ідентифікації можливо прискорити за допомогою використання комп’ютерних систем розпізнавання образів.

В останні роки розпізнавання образів знаходить все більшого застосування в таких напрямках, як охоронні системи, криміналістика, комп'ютерна графіка та ін. Розпізнавання образів вирішує задачу виділення істотних ознак та їх віднесення до певного класу, що характеризують даний образ, із загальної маси даних.

На сучасному етапі зображення стали невід’ємною частиною людського життя, а також важливою частиною багатьох галузей техніки. Область їх використання безперервно розширюється. Комп’ютеризовані процедури використовуються для полегшення сприйняття рентгенівських та інших зображень у промисловості, медицині та біології, для вивчення картини засмічення навколишнього середовища у географії. У фізиці та суміжних областях комп’ютерно оброблення є звичайним способом покращення якості зображень, що отримані в ході експериментів. Аналогічні приклади успішного використання технологій оброблення зображень можна знайти в астрономії, медичній радіології, промисловості, в оборонній та правоохоронній сферах [1].

Серед провідних закордонних та вітчизняних науковців, які займалися досліджуваною проблематикою, слід відзначити: Додонова О. Г., Торокіна О. О., Ланде Д. В., Марсіа Дж. Уілсона та ін. [1–5]

Останнім часом велика кількість робіт присвячена технічному, криптографічному та стеганографічному захисту інформації, серед яких варто відзначити праці Хорошка В. О., Яремчука Ю. Є., Карпінця В. В., Шелеста М. Є., Хорєва О. О. та ін. [6 – 10].

Не зважаючи на значний доробок у цій царині знань, наявні методи розпізнаванню образів не позбавлені недоліків, серед яких слід зазначити особливу складністю та вартісті технічної реалізації.

Одним із можливих продуктивних шляхів усунення недоліків в існуючих підходах розпізнавання зображень є застосування фільтра Габора та медіанної фільтрації для вдосконалення процедури фільтрування з метою розпізнавання облич.

Актуальність обговорюваного питання зумовлюється недоліком наведених способів, тобто неможливістю контролю користувачем величини згладжування зображення та фіксована кількість напрямків анізотропної фільтрації, що не дозволяє у достатній мірі покращити якість зображення [68].

Отже, дослідження в напрямку удосконалення технології ідентифікації образів, а саме розпізнавання облич на основі комп'ютерної системи архівування зображень на основі вдосконаленого методу фільтрування зображення засобами фільтра Габора є актуальним.

Мета і задачі дослідження. Метою роботи є розроблення та реалізація програмного засобу для підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення.

Задачами дослідження є:

– проаналізувати недоліки та переваги існуючих методів і засобів автентифікації та особливості їх застосування;

– дослідити специфіку використання та реалізації процедури ідентифікації користувачів засобами технології Face ID;

– удосконалити метод фільтрування зображення засобами технології Face ID на основі фільтра Габора;

– розробити алгоритм роботи програмного додатку для ідентифікації користувачів на основі удосконаленого методу фільтрації зображень;

– обґрунтувати вибір мови та середовища програмування для додатку;

– спроектувати та розробити інтерфейс користувача додатку;

– розробити програмний засіб для підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення;

– обґрунтувати економічну доцільність програмного додатку.

Об’єкт дослідження – процес ідентифікації користувачів засобами технології Face ID.

Предмет дослідження – удосконалений метод фільтрування зображення засобами технології Face ID на основі фільтра Габора.

Наукова новизна дослідження полягає в удосконаленні методу фільтрування зображення, що, на відміну від існуючих підходів, дозволяє підвищити точність автентифікації користувачів засобами технології Face ID на основі фільтра Габора.

Практична цінність дослідження полягає у створенні програмного засобу, що уможливлює більш точну автентифікацію користувачів на основі удосконаленої технології Face ID.

Апробація результатів дослідження відбулася на XLVII Науково-технічній конференції підрозділів Вінницького національного технічного університету (м. Вінниця, 2018р.) [11 – 13], XLIX Науково-технічній конференції підрозділів Вінницького національного технічного університету (м. Вінниця, 2020р.) [14] та Всеукраїнській науково-практичній Інтернет-конференції студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2022) [15 – 16]. Результати роботи представлені у 1 турі Всеукраїнського конкурсу студентських наукових робіт зі спеціальності "Кібербезпека" у 2021-2022рр.

Публікації. Було опубліковано 6 тез доповідей.

1 ТЕОРЕТИЧНІ ЗАСАДИ БІОМЕТРИЧНОЇ ІДЕНТИФІКАЦІЇ КОРИСТУВАЧІВ

1.1 Роль та значення біометричної ідентифікації в системах автентифікації користувачів

Біометричні системи захисту інформації – системи контролю доступу, засновані на ідентифікації та аутентифікації людини за біологічними ознаками, такими як структура ДНК, малюнок райдужної оболонки ока, сітківка ока, геометрія та температурна карта обличчя, відбиток пальця, геометрія долоні. Також ці методи аутентифікації людини називають статистичними методами, оскільки засновані на фізіологічних характеристиках людини, присутніх від народження і до смерті, що перебувають при ньому протягом усього життя, і які не можуть бути втрачені або вкрадені. Часто використовуються ще й унікальні динамічні методи біометричної автентифікації – підпис, клавіатурний почерк, голос та хода, що ґрунтуються на поведінкових характеристиках людей.

Поняття «біометрія» з'явилося наприкінці XIX століття. Розробкою технологій для розпізнавання образів з різних біометричних характеристик почали займатися вже досить давно, початок було покладено в 60-ті роки минулого століття. Значних успіхів у розробці теоретичних засад цих технологій досягли наші співвітчизники. Проте практичні результати отримано здебільшого на заході й зовсім недавно. Наприкінці двадцятого століття інтерес до біометрії значно зріс завдяки тому, що потужність сучасних комп'ютерів та вдосконалені алгоритми дозволили створити продукти, які за своїми характеристиками та співвідношенням стали доступними та цікавими широкому колу користувачів. Галузь науки знайшла своє застосування у розробках нових безпекових технологій. Наприклад, біометрична система може контролювати доступ до інформації та сховищ у банках, її можна використовувати на підприємствах, зайнятих обробкою цінної інформації, для захисту ЕОМ, засобів зв'язку тощо [17].

Суть біометричних систем зводиться до використання комп'ютерних систем розпізнавання особи за унікальним генетичним кодом людини. Біометричні системи безпеки дозволяють автоматично розпізнавати людину за її фізіологічними або поведінковими характеристиками.

Усі біометричні системи працюють за однаковою схемою. Спочатку відбувається процес запису, в результаті якого система запам'ятовує зразок біометричної характеристики. Деякі біометричні системи роблять кілька зразків для докладнішого відображення біометричної характеристики. Отримана інформація обробляється та перетворюється на математичний код. Біометричні системи інформаційної безпеки використовують біометричні методи ідентифікації та аутентифікації користувачів. Ідентифікація по біометричній системі проходить у чотири стадії [18]:

Реєстрація ідентифікатора – відомості про фізіологічну або поведінкову характеристику перетворюється на форму, доступну комп'ютерним технологіям, і вносяться в пам'ять біометричної системи;

Виділення – із знову пред'явленого ідентифікатора виділяються унікальні ознаки, аналізовані системою;

Порівняння – зіставляються відомості про знову пред'явлений і раніше зареєстрований ідентифікатор;

Рішення – виноситься висновок у тому, збігаються чи збігаються знову пред'явлений ідентифікатор.

Висновок про збіг/несупад ідентифікаторів може потім транслюватися іншим системам (контролю доступу, захисту інформації тощо), які далі діють на основі отриманої інформації [19].

Говорячи про методи біометричної ідентифікації, їх можна поділити на дві великі групи: статичні та динамічні (рис. 1.1). Більш детально розглянемо їх у наступному підрозділі.

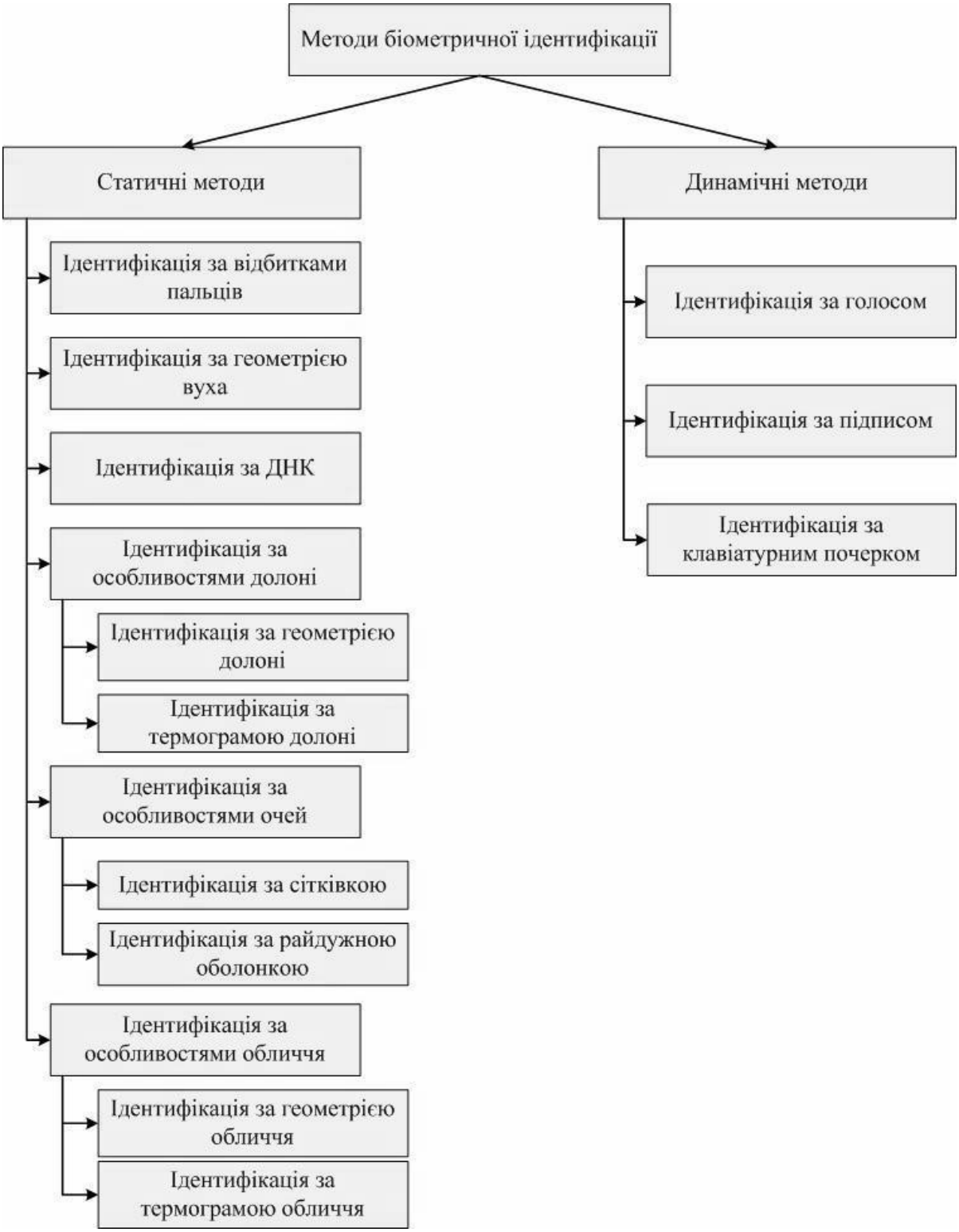


Рисунок 1.1 – Класифікація методів біометричної ідентифікації (за [20])

Розглянемо більш детальніше переваги та недоліки застосування системи біометричної ідентифікації особистості з огляду на цифрову безпеку.

Переваги. Ідентифікація за відбитками пальців і по обличчю вже набули широкого поширення завдяки смартфонам, які дозволяють її використовувати.

Треба визнати, що у неї є деякі суттєві переваги перед звичайними методами ідентифікації і аутентифікації [21]:

– найочевиднішою перевагою є унікальність основи ідентифікації. Якщо відбиток пальця успішно лічений, це гарантує, що мова про те самій людині. Крім того, більшість біометричних показників неймовірно важко підробити;

– біометрична ідентифікація зручна для користувачів. Замість запам'ятовування численних паролів або носіння з собою засвідчують особу досить пред'явити очей, вухо, палець або посміхнутися в камеру. Налаштування біометрії теж досить проста;

– висока точність. Новітнє дослідження показало, що точність розпізнавання відбитку одного пальця становить 98,6%, двох - 99,6%, чотирьох і більше - 99,9%;

– невисока загальна вартість володіння. Біометрична система може бути дорогою, але витрати на управління нею нижче, ніж у звичайних систем.

Недоліки. У біометрії є такі слабкі сторони [22]:

– обмежені можливості пристроїв. В даний момент найбільш зручним і портативним пристроєм з можливостями біометрії є смартфон, але йому притаманні обмеження. У нього невелика поверхню сканування пальця, що дозволяє вивчити лише частину відбитка.

– модифікації. Біометрія передбачає незмінність фізичних особливостей людини.

– перезавантаження. Одна з переваг традиційних заходів цифровий безпеки полягає в тому, що їх можна здійснювати віддалено.

– системні обмеження. Біометрія використовує бази даних, а бази даних уразливі.

Варто зауважити, що однією з найважливіших характеристик систем захисту інформації, заснованих на біометричних технологіях, є висока надійність, тобто здатність системи достовірно розрізняти біометричні характеристики, що належать різним людям, та надійно знаходити збіги. У біометрії ці параметри називаються помилкою першого роду (False Reject Rate,

FRR) та помилкою другого роду (False Accept Rate, FAR). Перше число характеризує можливість відмови доступу людині, що має доступ, друге – можливість помилкового збігу біометричних параметрів двох людей. Підробити папілярний візерунок пальця людини або райдужну оболонку ока дуже складно [23].

Технологія	Переваги	Недоліки
Пароль, PIN - код, ключові слова і так далі	1. Низька вартість 2.Немає необхідності носити аутентифікатор	1. Користувачі часто застосовують короткі паролі, що легко підбираються 2. Існує можливість перехоплення паролів 3. Необхідність регулярно змінювати паролі 4. Складно запам’ятати велику кількість паролів
Відбиток пальця, особливості райдужної оболонки ока, форми кисті рук і т.д.	1. Унеможливлюється несанкціоноване використання даних автентифікації 2. Забезпечується високий ступінь захисту від імітації 3. Майже виключена можливість втрати даних для ідентифікації 4. Відсутні витрати на виготовлення, купівлю даних ідентифікації	1. Неможливо отримати статичний ключ 2. Можливість помилок першого і другого року (пропуск або помилкова тривога)

Рисунок 1.2 – Переваги біометричної ідентифікації порівняно з парольною

Отже, виникнення "помилок другого роду" (тобто надання доступу людині, яка не має на це право) практично виключена. Проте, під впливом деяких чинників біологічні особливості, якими виробляється ідентифікація особистості, можуть змінюватися. Наприклад, людина може застудитися, внаслідок чого її голос зміниться до невпізнання. Тому частота появ "помилок першого роду" (відмова у доступі людині, що має на це право) у біометричних системах досить велика. Система тим краще, що менше значення FRR при однакових значеннях FAR. Іноді використовується і порівняльна характеристика EER (Equal Error Rate), що визначає точку, в якій графіки FRR та FAR перетинаються. Але вона далеко не завжди репрезентативна. При використанні біометричних систем, особливо системи розпізнавання по

обличчю, навіть при введенні коректних біометричних характеристик який завжди рішення про аутентифікації правильно. Це пов'язано з низкою особливостей і, в першу чергу, з тим, що багато біометричних показників можуть змінюватися. Існує певний ступінь ймовірності помилки системи. Причому при використанні різних технологій помилка може суттєво відрізнятись. Для систем контролю доступу під час використання біометричних технологій необхідно визначити, що важливіше не пропустити "чужого" чи пропустити всіх «своїх» [24].

Не лише FAR та FRR визначають якість біометричної системи. Якби це було тільки так, то провідною технологією було б розпізнавання людей ДНК, для якої FAR і FRR прагнуть нуля. Але очевидно, що ця технологія не застосовна на сьогоднішньому етапі розвитку людства. Тому важливою характеристикою є стійкість до муляжу, швидкість роботи та вартість системи. Не варто забувати і те, що біометрична характеристика людини може змінюватися з часом, тому якщо вона нестійка - це істотний мінус. Також важливим фактором для користувачів біометричних технологій у системах безпеки є простота використання. Людина, характеристики якої скануються, не має при цьому відчувати жодних незручностей. У цьому плані найцікавішим методом є, безумовно, технологія розпізнавання обличчя. Щоправда, у разі виникають інші проблеми, пов'язані насамперед, з точністю роботи системи.

Зазвичай біометрична система складається з двох модулів: модуль реєстрації та модуль ідентифікації [25].

Модуль реєстрації «навчає» систему ідентифікувати конкретну людину. На етапі реєстрації відеокамера або інші датчики сканують людину для того, щоб створити цифрове представлення її вигляду. Через війну сканування чого формуються кілька зображень. В ідеальному випадку, ці зображення матимуть трохи різні ракурси та вирази обличчя, що дозволить отримати більш точні дані. Спеціальний програмний модуль обробляє це уявлення і визначає характерні риси особистості, потім створює шаблон. Існують деякі частини особи, які практично не змінюються з плином часу, це, наприклад, верхні

обриси очниць, області навколишні вилиці, та краї рота. Більшість алгоритмів, розроблених для біометричних технологій, дозволяють враховувати можливі зміни в зачісці людини, оскільки вони не використовують для аналізу області обличчя вище за межі зростання волосся. Шаблон зображення кожного користувача зберігається у базі даних біометричної системи.

Модуль ідентифікації отримує від відеокамери зображення людини і перетворює його на той же цифровий формат, в якому зберігається шаблон. Отримані дані порівнюються зі шаблоном, що зберігається в базі даних, для того, щоб визначити, чи відповідають ці зображення один одному. Ступінь подібності, необхідна для перевірки, є деяким порігом, який може бути відрегульований для різного типу персоналу, потужності РС, часу доби та низки інших факторів [26].

Ідентифікація може виконуватися як верифікації, аутентифікації чи розпізнавання. При верифікації підтверджується ідентичність отриманих даних та шаблону, що зберігається у базі даних [27]. Аутентифікація – підтверджує відповідність зображення, одержуваного від відеокамери одному з шаблонів, що зберігаються у базі даних. При розпізнаванні, якщо отримані характеристики і один із шаблонів, що зберігаються, виявляються однаковими, то система ідентифікує людину з відповідним шаблоном.

1.2 Аналіз методів біометричної ідентифікації користувачів

Біометричні системи ідентифікації користувачів, як вже зазначалось, поділяються на статичні та динамічні. У даному підрозділі розглянемо їх детальніше.

Статичні системи. До даного типу відносяться ті методи і інструменти, які оцінюють біометричні параметри в статиці – без розвитку в часі [28].

У таблиці 1.1 наведено порівняльну характеристику розглянутих статичних методів відповідно їх надійності, відсоток виникнення ситуацій, коли система дозволяє доступ користувачу, незареєстрованим в системі (FAR), відмова в доступі справжньому користувачеві системи (FRR) [29].

Таблиця 1.1 – Порівняння статичних методів біометричної ідентифікації особистості [17 – 18]

Біометричний метод	FAR %	FRR %	Фальсифікація	Незмінність характеристик	Чутливість до зовн. факторів	Швидкість автентифікації	Вартість
Відбиток пальця	0,001	0,6	Можлива	Низька	Висока	Висока	Низька
Розпізнавання форми лиця 2D	0,1	2,5	Можлива	Низька	Висока	Середня	Середня
Розпізнавання форми лиця 3D	0,0005	0,1	Проблематична	Висока	Низька	Низька	Висока
Райдужна оболонка ока	0,00001	0,016	Безуспішна	Висока	Середня	Висока	Висока
Сітківка ока	0,0001	0,4	Неможлива	Середня	Висока	Низька	Висока
Візерунок вен	0,0008	0,01	Неможлива	Середня	Середня	Висока	Середня
Розпізнавання за формою губ	0,2	3,8	Можлива	Низька	Середня	Середня	Середня
Розпізнавання за формою носа	0,8	4,9	Можлива	Середня	Низька	Середня	Середня
Розпізнавання за формою уха (резонування)	0,002	0,1	Проблематична	Висока	Висока	Висока	Низька
Розпізнавання за кодом ДНК	0,00001	0,0012	Проблематична	Висока	Середня	Низька	Висока
Розпізнавання за формою кисті	0,1	2,5	Можлива	Низька	Середня	Середня	Середня

Ідентифікація відбитків пальців [30]. Один із видів, ідентифікація відбитків пальців, є найстарішою з усіх. Відбиток пальців є надійним методом аутентифікації людини, оскільки всі відбитки пальців унікальні. Оскільки, навіть, однояйцеві близнюки мають різні відбитки пальців. Ідентифікація відбитків пальців заснована на розпізнаванні візерунків, де дуги, петлі та витки виступів відбитків пальців порівнюються з раніше збереженою інформацією. Оскільки, ця технологія досягла значного прогресу протягом останніх років, ідентифікація відбитків пальців зараз надзвичайно швидка.

Розпізнавання відбитків долоні [31]. Система розпізнавання відбитків долоні виявляє людину за допомогою функцій відбитків долоні, які можуть бути або не бути помітними неозброєним оком. Відбитки долонь багаті фізичними атрибутами візерунків шкіри, такими як лінії, точки та текстури. Ця технологія використовується разом з ідентифікацією відбитків пальців для підвищення точності ідентифікації.

Голосова ідентифікація (або розпізнавання голосу) [32 – 33]. Розпізнавання голосу є доцільною формою біометричної аутентифікації, яка виявляє особисті голосові моделі для автентифікації особистості. Ця технологія використовує вроджені біологічні характеристики голосу людини, щоб

створити унікальний для цієї людини відбиток голосу. Розпізнавання голосу важко підробити завдяки його біометричним властивостям. З іншого боку, він більш зручний для користувачів, оскільки користувачам не потрібно запам’ятовувати паролі та відповіді на таємні запитання. Голосова ідентифікація не тільки для колл-центрів. Нині додатки мобільного банкінгу також використовують цей метод для підвищення безпеки мобільного банкінгу.

Розпізнавання райдужної оболонки [34]. Ця форма біометричної технології відноситься до тієї ж категорії, що й розпізнавання облич. Біометрія сканування райдужної оболонки вимірює унікальні візерунки в кольоровому колі вашого ока, щоб ідентифікувати та підтвердити особу людини. Технології ідентифікації на основі діафрагми зазвичай використовуються як спосіб контролю фізичного доступу, який ідеально підходить для високопродуктивних середовищ, які вимагають високої точності та швидкості. Сканери райдужної оболонки ока збирають близько 240 біометричних ознак, комбінація яких унікальна для кожного ока. Цікаво, що зчитувачі для розпізнавання райдужної оболонки ока встановлені в Європейській організації ядерних досліджень (CERN) у Женеві, Швейцарія.

Форма вушної раковини [35]. На відміну від багатьох інших біометричних методів, для яких потрібні спеціальні камери, ці біометричні системи вимірюють акустику вуха за допомогою спеціальних навушників і нечутні до звукових хвиль. Мікрофон всередині кожного навушника вимірює те, яким чином звукові хвилі відбиваються від вушної раковини і розходяться в різних напрямках в залежності від вигинів слухового проходу. Цифрова копія форми вуха перетворюється в біометричний шаблон для подальшого використання.

Голос [36]. Технологія розпізнавання голосу потрапляє в сфери і фізіологічних, і поведінкових біометричних даних. З фізіологічної точки зору такі системи розпізнають форму голосового тракту людини, включаючи ніс, рот і гортань, визначають вироблений звук. З поведінкової точки зору вони фіксують те, як людина щось каже – варіації рухів, тон, темп, акцент і т. д., Що

також є унікальним для кожної людини. Об'єднання даних фізичної і поведінкової біометрії створює точний голосовий підпис, хоча можуть виникати деякі невідповідності (наприклад, в разі хвороби або дії інших факторів).

Термограма [37]. Термограма – це уявлення інфрачервоної енергії у вигляді зображення розподілу температури. Біометрична термографія особи фіксує теплові візерунки, викликані рухом крові під шкірою. Оскільки кровоносні судини кожної людини неповторні, відповідні термограми також унікальні навіть серед спільнояйцевих близнюків, що робить цей метод біометричної верифікації навіть більш точним, ніж традиційне розпізнавання облич.

ДНК [38]. ДНК здавна використовувалася як метод ідентифікації. Крім того, це єдина форма біометрії, яка може відстежувати родинні зв'язки. Зіставлення ДНК є особливо цінним під час роботи зі зниклими без звісток, виявленні жертв катастроф і потенційної торгівлі людьми. До того ж, ДНК – єдиний біометричний об'єкт (крім відбитків пальців), який неможливо ненавмисно «забути».

Поведінкові системи [38]. Принципи поведінкової біометрії засновані на особливостях руху людини і його поведінкових характеристиках. Нижче перераховано основні види поведінкової біометрії (рис. 1.3) [28].

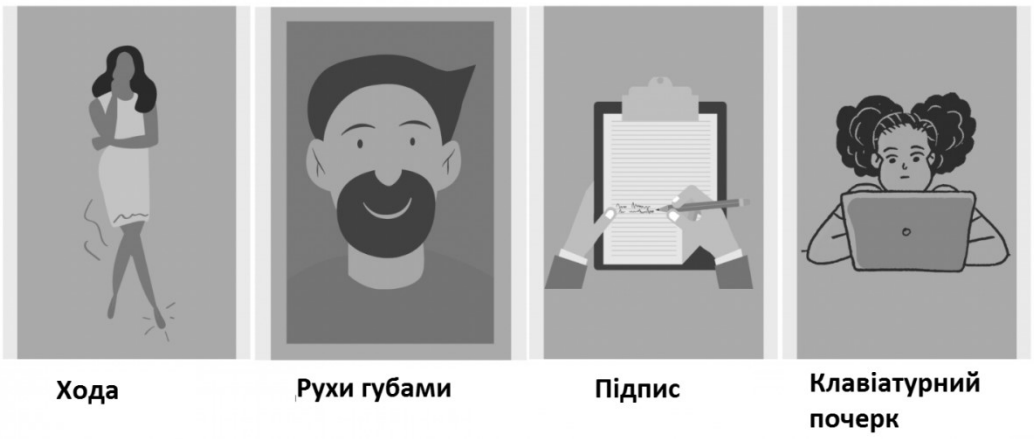


Рисунок 1.3 – Типи поведінкових біометричних систем

Таблиця 1.2 – Порівняння динамічних методів біометричної ідентифікації особистості [37 – 38]

Біометричний метод	FAR %	FRR %	Фальсифікація	Незмінність характеристик	Чутливість до зовн. факторів	Швидкість автентифікації	Вартість
Розпізнавання за ходою людини	0,001	0,6	Можлива	Низька	Висока	Висока	Низька
Розпізнавання за запахом тіла	0,1	1,5	Проблематична	Низька	Висока	Середня	Середня
Розпізнавання за складом поту	0,0005	0,1	Проблематична	Висока	Низька	Низька	Середня
Розпізнавання за мікровібрацією пальців	0,00001	0,016	Проблематична	Висока	Середня	Висока	Висока
Розпізнавання за почерком (клавіатура)	0,012	0,4	Можлива	Середня	Низька	Низька	Низька
Розпізнавання за електроміограмою	0,0008	0,023	Неможлива	Середня	Середня	Висока	Середня
Розпізнавання за фотоплетизмограмою	0,0007	0,01	Проблематична	Середня	Середня	Висока	Низька
Розпізнавання за геометрією серця	0,003	0,012	Неможлива	Середня	Середня	Висока	Середня
Розпізнавання за електрокардіограмою	0,0005	0,01	Неможлива	Середня	Середня	Висока	Середня

Хода [39]. Біометрія ходи фіксує шаблони кроків за допомогою відеозображення, а потім перетворює зіставлені дані в математичне рівняння.

Цей тип біометричних даних є ненав'язливим і непомітним, що робить його ідеальним для масового спостереження за натовпом. Також перевагою є те, що ці системи можуть швидко ідентифікувати людей здалеку. Технологія може бути дуже корисною для використання в магазинах, банках та інших організаціях – наприклад, для виявлення можливих злочинців.

Рух губ [39]. Це – одна з новітніх форм біометричної верифікації. Подібно до того, як глуха людина може відстежувати рух губ, щоб визначити сказане, біометричні системи фіксують активність м'язів навколо рота, щоб сформувати шаблон їх руху. Біометричні датчики такого роду часто вимагають відтворення користувачем пароля, щоб визначити відповідні рухи губ, а потім на основі порівняння з записаним шаблоном надати або заборонити доступ.

Підпис [39]. Розпізнавання підпису – це поведінкова біометрична система, яка вимірює просторові координати, тиск пера, його нахил і хід як в автономних, так і в інтерактивних додатках.

Цифровий планшет записує вимірювання, а потім використовує цю

інформацію в ході автоматичного створення біометричного профілю для майбутньої верифікації. В даний час для введення підпису використовуються планшети, які автоматично фіксують положення ручки в різні моменти часу, кути нахилу і тиск, який чиниться на планшет.

За допомогою кнопок [40]. Динаміка натискання клавіш виводить стандартні паролі на новий рівень, відстежуючи ритм їх введення. Такі датчики можуть реагувати на час, що витрачається на натискання кожної клавіші, затримки між клавішами, кількість символів, що вводяться за хвилину, і так далі.

Мультимодальні біометричні системи [40]. Біометрію розглядають і як мультимодальне технологію. Поєднання декількох типів вимірювань дозволяє підвищити і рівень безпеки, і ефективність роботи систем ідентифікації. Тому останнім часом все більше компаній пропонують мультимодальні біометричні системи, а споживачі орієнтуються на комплексні рішення.

1.3 Аналіз методів реалізації Face ID

Розпізнавання образів – виділення істотних ознак та їх віднесення до певного класу, що характеризують даний образ із загальної маси даних [41]. Розпізнавання облич – один із підрозділів більш широкої категорії розпізнавання образів. Алгоритми та методи розпізнавання мають схожість, проте відмінні за функціями розпізнавання, а точніше за їх параметрами [41].

Список задач, які вирішують розпізнавання образів значно зріс із підвищенням обчислювальних потужностей та вдосконаленням алгоритмів розпізнавання зріс [36]:

– комп'ютерні ігри та віртуальна реальність;

– шифрування даних;

– електронна комерція;

– взаємодія комп'ютер-людина;

– доступ до інформаційних баз;

– персоналізація побутових пристроїв;

– соціальні сервіси.

На сьогодні вже інтегровано алгоритми автоматичного розпізнавання облич на фотографіях в популярних соціальних сервісах Facebook та Google Picassa.

На відміну від розглянутих вище алгоритмів ідентифікації, реальна система розпізнавання – це функціонально взаємопов'язана сукупність методів і технічних засобів, що здійснює процес синтезу і аналізу образів [42].

Перш ніж приступати до побудови системи розпізнавання, необхідно проаналізувати всю доступну інформацію про об'єкти дослідження і вирішити такі питання:

– якими загальними характеристиками і властивостями володіють об'єкти дослідження і чим вони відрізняються;

– якщо необхідні характеристики можуть бути отримані в результаті вимірювань, яка точність цих вимірів;

– чи існує відповідна модель (моделі) для формального опису та аналізу даних характеристик.

На підставі проведених досліджень визначається тип і структура системи розпізнавання. Систему розпізнавання образів можна поділити на прості і складні. Прикладом простої системи розпізнавання є класифікація пікселів на зображенні та виділенні предметів на фото. Як тільки в процес класифікації залучаються інші типи даних, система стає складною. Складні системи розпізнавання можуть бути однорівневими і багаторівневими. У однорівневих системах розпізнавання здійснюється одним алгоритмом розпізнавання. У багаторівневих системах результати розпізнавання, отримані на одному етапі, використовуються в якості вихідних даних для наступного [42].

При вирішенні прикладних задач розпізнавання облич необхідно застосовувати багаторівневі системи. Розбиття схеми виконання завдання на рівні називається декомпозицією. Важливою функцією для багаторівневих

систем є аналітика даних, а саме оцінка якості розпізнавання на поточному етапі, оскільки це може спричинити до втрат інформації на наступному етапі.

При вирішенні задачі розпізнавання осіб з’являються дві проблеми. Першою проблемо є те, що будь-яке зображення це лише масив пікселів. Проте один піксель не має істотного значення, адже зміна його кольору не вплине на результат розпізнавання. Тому подання зображень у вигляді масиву пікселів є надлишковим і неекономічним [42].

Зображення, що вводяться в комп'ютер, часто є мало контрастними. Зазвичай слабкий контраст є результатом широкого діапазону яскравостей, що відтворюються в поєднанні з нелінійною характеристикою передачі рівнів. Також на якість зображення впливає характер зміни яскравості пікселів від мінімального значення до максимального, адже лінійна функція зміни інтенсивності пікселів є оптимальною. Якщо характеристика зображення буде увігнутою, тоді зображення більш темне, а при опуклій характеристиці буде більш світлим. В обох випадках ознаки об'єктів можуть бути спотворенні і недостатньо добре ідентифіковані. Корекція гістограми яскравості істотно покращує якість зображення.

Мала контрастність може бути обумовлена й тим, що варіації функції яскравості пікселів на зображенні набагато менше допустимого діапазону шкали яскравостей. В даному випадку контрастність зображення підвищується шляхом розширення динамічного діапазону яскравостей на всю шкалу за допомогою лінійного по елементного перетворення [42].

Інверсія вхідного зображення є іншим способом корекції яскравості палітри. Оскільки відрізняти слабкі сигнали на темному фоні непросто, то інверсна форма подання зображення має більш задовільну форму для спостереження і візуальної ідентифікації.

Деякі завдання обробки зображення пов'язані з перетворенням напівтонового зображення (різні градації яскравості) в бінарне. Щоб скоротити інформаційну надмірність зображення здійснюється перетворення зображення в бінарне. В результаті залишається лише інформація, яка потрібна для

вирішення задачі розпізнавання. У бінарному зображенні повинні бути усунені несуттєві ознаки, наприклад, фон та збережені певні деталі, наприклад, обриси зображених об'єктів.

Порогове оброблення напівтонового зображення полягає в розбитті елементів зображення на два класи A_1 і A_2 за яскравістю властивостей з кордоном Агр,. Після розділення необхідно виконати порогову фільтрацію із заміною пікселів зображення на встановлену яскравість класів [42].

В силу недостатньої експозиції діапазон яскравості зображення в цифровому вигляді може мати відмінності від діапазону яскравостей вихідного зображення. Виконати корекцію яскравості можливо за допомогою двох способів. Перший спосіб передбачає, що зображення лінійно відображається в діапазоні яскравостей вихідного. При застосуванні другого способу необхідно виконати обмеження яскравості пікселів між максимальним і мінімальним пороговими рівнями. Присутність у зображенні найсвітліших і темних тонів створює враження хорошої контрастності. Надмірна контрастність спричиняє, що максимальні градації впливають на середні тони, адже більшість елементів зображення знаходяться в середніх тонах, а зайва контрастність може спричинити втрати цих деталей або ускладнити їх виділення.

Основини інструментом, щоб виконати оцінку інтенсивності пікселів є гістограма. Гістограма – графічне відображення кількісної характеристики імовірнісного розподілу інтенсивності (яскравості) пікселів в виділеній ділянці зображення.

Максимально можливе значення інтенсивності пікселів 255 і відповідає білому кольору, а 0 відповідає чорному кольору. Інтенсивності в діапазоні від 0 до 255 мають лінійну шкалу зміни, або установлюється у відповідності з прийнятою функцією зміни, наприклад, посилюючої слабкі сигнали (градації сірого) і послаблюючи сильні сигнали (в області білого кольору). На рис. 1.4 наведено приклад побудови гістограм яскравостей.

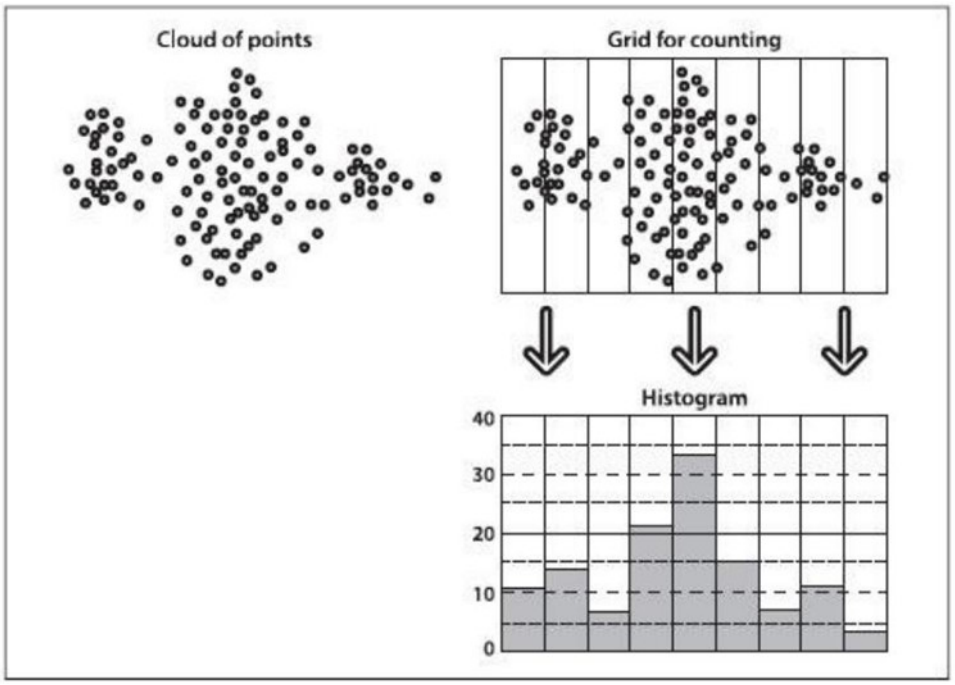


Рисунок 1.4 – Приклад побудови гістограм яскравостей [42]

Велика кількість зображень мають гістограму з високою концентрацією ліній в певних зонах розподілу інтенсивності. В більшості випадків гістограма розподілу яскравостей зображення має перекіс у бік малих рівнів (яскравість більшості елементів нижче середньої). Вирівнювання гістограми є одним з методів поліпшення якості таких зображень. Робота із перетворенням яскравостей цифрового зображення має назву гістограмне перетворення. Оскільки змінюється не лише гістограма зображення, а також вид функції перетворення, яка визначається адаптивно на основі раніше побудованої гістограми вихідного зображення. У випадку дискретних цифрових зображень функція відображення яскравості (1.1) набуває такого вигляду:

$$lm[i,j] \;=\; LUT[lm[i,j]], \tag{1.1}$$

де lm[i, j] – піксель зображення з координатами ⟨i, j⟩⟨i, j⟩;

LUT – цілочисельний масив розміру 256256 елементів для 8-бітного зображення або довжиною 65536 елементів для 16-бітного зображення. Далі розглянемо гістограмне оброблення на прикладі 8-бітних напівтонових зображень.

При цьому LUT може задаватися:

– довільної таблицею відображення, що формується користувачем безпосередньо;

– деякої математичної функцією, обраної з заданого набору функцій;

– адаптивно до гістограми вхідного аналізованого зображення.

У більшості випадків певні області на зображенні бувають занадто темними, щоб на них можна було щось розібрати. Просте підвищення яскравості для всього зображення, може призвести, що світлі ділянки можуть стати засвіченими. Щоб покращити вигляд зображення в таких випадках, застосовується метод вирівнювання освітленості.

Яскравість можна вважати низькочастотною складовою, оскільки освітленість змінюється у просторі повільно, а зображення – високочастотним сигналом, оскільки зображення може містити дрібні деталі, що призводить до швидких змін у його текстурі та конфігурації. Добуток цих складових можна представити у вигляді результуючого сигналу на первинному зображенні: за допомогою залежності (1.2):

$$(x+a)^n = \sum_{k=0}^{n} \binom{n}{k} x^k a^{n-k} \tag{1.2}$$

де $f(uiv)$ – інтенсивність зображення;

мм $fi(uiv)$ – функція освітленості;

$f\tau(uiv)$ – функція відображуваної здатності;

uiv – дискретні просторові зміни.

Із метою звуження динамічного діапазону обробленню підлягає складова освітленості, а для підвищення контрастності – складова відображувальної здатності [43].

У результаті отримуються мультиплікативні складові зображення, які є розділеними та надалі можуть оброблюватися незалежно один від одного. До отриманого сигналу використовується лінійний фільтр високих частот.

Основним завданням гомморфного оброблення зображень є нормалізація рівнів яскравостей, а саме звуженні їх динамічного діапазону та одночасному підвищенні його конрастності.

Під час оброблення зображень важливого значення набувають алгоритми рангової фільтрації, адже їх функціонал дозволяє позбавлятися розмитих деталей. Поліпшення фокусу зображення виконується за рахунок вибору двовимірної n × n маски пікселів. Ранжирування значень інтенсивності виконується в межах маски, а центральному пікселю присвоюється значення, яке рівне максимальному значенню по рангу.

Лінійна фільтрація надає можливості для корекції спотворень різного типу, що виникають внаслідок недосконалості приладів, що створюють зображення. Окрім того, за допомогою лінійної фільтрації виконується зменшення впливу флуктуаційних шумів та інших дефектів на зображеннях в режимі підвищення контрасту мало контрастних деталей при збільшенні масштабу зон інтересу.

Корекція спотворень зображень в разі відсутності шумової складової сигналу здійснюється шляхом інверсної фільтрації (деконволюції). Проте необхідно пам’ятати, що фільтри деконволюції мають коефіцієнт посилення дисперсії шумів, більший 1, і натомість може збільшитися зашумленість зображення [44 – 45].

Перспективними вважаються нелінійні методи фільтрації на основі частотних масок, які дозволяють зменшити вплив низькочастотних компонент сигналу зображення і посилити вплив високочастотних компонент, що підвищує просторову роздільну здатність деталей, описаних у спектрі зображення більш високими просторовими частотами.

1.4 Аналіз існуючих методів фільтрації зображення

У тому випадку, якщо зображення або відео послідовність були отримані за допомогою оцифрування, на них, як правило, присутній шум. Шум вносить спотворення в цифрові зображення.

Тому оброблення та покращення якості зображень як задля покращення їх візуального сприйняття людиною, так і для вирішення завдань, пов'язаних з машинним сприйняттям зображень, є важливою областю сучасної роботи та потребує постійного покращення та удосконалення методів, за допомогою яких дані завдання можуть бути виконані [46].

Методи фільтрації зображень.

Зображення, яке спостерігається отримане за допомогою приладів математично можна записати так:

$$v(i) = u(i) = n(i) \tag{1.3}$$

де $v(i)$ – величина, яка спостерігається;

$u(i)$ – дійсна величина, яку необхідно відновити з $v(i)$;

$n(i)$ – шумове відхилення.

Метод очищення від шуму зображення u може бути визначений як D_h:

$$u = D_h u = n(D_h, u) \tag{1.4}$$

де h – параметр фільтрації;

D_h, u – очищене від шуму зображення;

$n(D_h, u)$ – шум, визначений цим методом.

У даний час недостатньо просто згладити u та отримати очищене зображення. Більш сучасні методи не тільки згладжують зображення, але й намагаються відновити втрачену інформацію $n(D_h, u)$, якщо це необхідно, тобто в зображенні отриманому за допомогою цифрової дзеркальної камери

нам часто необхідно зберегти чіткість та деталі, в той час як шум необхідно розмити.

Існує велика кількість методів покращення якості зображень. Для аналізу обрано такі методи [47 – 48]:

1. Локальні методи фільтрації, в тому числі [49]:

– гаусівська модель згладжування [49], де згладжування *и* вимірюється інтегралом Дірихле $\int |Du|^2$.

– фільтр околів Ярославського [4] та Susan фільтр [50];

– білатеральний фільтр [51].

2. Методи, які ґрунтуються на диференційних рівняннях з частковими похідними (ДРЧП), включаючи [52]:

– модель анізотропної фільтрації [52];

– модель повної варіації Рудіна-Ошера-Фатемі [53];

3. Фільтри в частотній області, що включають [54]:

– локальні адаптивні фільтри в областях перетворення [54];

– жорсткий та м’який трешолдинг [55 – 56];

– zhou-Wang вейвлет повна варіація [57].

4. Алгоритм нелокального усереднення (NL-means) [58].

Розглянемо детальніше локальні методи фільтрації.

Оригінальне (півтонове) зображення u визначається в обмеженій області $\Omega \subset R2$ та позначається як $u(x)$ для $x \in R^2$. Відповідно до звичайної практики візуалізації на екрані або на принтері ми не інтерполюємо дискретні значення шуму n_i як функцію з обмеженим спектром, але як кусочно-постійну функцію, постійну в кожному пікселі i та рівній n_i.

Ми запишемо $|x|$ як L^2 норму та $x \cdot y$ як скалярний добуток.

1. Фільтр Гауса. Це найбільш частий у використанні фільтр розмиття. Це насправді згортка зображення лінійним симетричним ядром. Необхідність згладжування зазвичай виражається з допомогою позитивності ядра. Формула такого ядра також називається ядром Гауса.

$$x \to G_h(x) = \frac{1}{(4\pi h^2)^e}^{-\frac{|x^2|}{4h^2}} \quad (1.5)$$

G_h має середнє квадратичне відхилення. Обчислення правильне якщо h достатньо мале. З іншого боку, властивості зменшення шуму залежать від того факту, що околи, які приймають участь у згладжуванні достатньо великі, таким чином шум зменшується шляхом усереднення.

У подальшому, якщо ми припустимо, що $h \in k\epsilon$, де k – кількість відліків функції u та шуму в інтервалі довжиною h, ϵ^2 – розмір локального вікна, k повинно бути більше за 1.

2. Фільтри околів. Фільтри околів приймають до уваги значення рівнів сірого для визначення сусідніх пікселів. В такому випадку очищене від шуму значення пікселя *i* є (зваженим) середнім значень у пікселях, які мають значення рівня сірого близьке до *u(i)*. Можна визначити окіл рівня сірого, як:

$$U(i,h) = \{j \in I | u(i) - h < u(j) < u(i) + h \quad (1.6)$$

Таким чином, це також локальна схема, тобто локальна в області інтенсивності (яскравості). Але вона нелокальна в просторовій області, оскільки пікселі, які належать всьому зображенню використовуються для обчислення в пікселі *i*.

3. Білатеральні фільтри [59]. Білатеральний фільтр вперше був запропонований C. Tomasi та R. Manduchi [59] у 1998 році. Він застосовує просторове зважене усереднення без згладжування країв. Це досягається шляхом комбінування двох гаусівських фільтрів: один фільтр працює в просторовій області, а інший в області інтенсивності (яскравості). Таким чином, не тільки просторова відстань, а й відстань інтенсивності також важлива для визначення вагів.

Більшість методів для фільтрації зображень зосереджені на півтонових зображеннях з додаванням штучного шуму. І лише мала частина призначається для природних кольорових фотографій, отриманих цифровою камерою з реальним шумом. Але ці методи є або надто складними, або не можуть вирішити проблему зменшення шуму в достатній мірі.

Тому необхідно продовжити дослідження в напрямку визначення чинників, які впливають на зменшення шуму та підвищення якості зображень з реальним шумом [59 – 60].

1.5 Висновки та постановка задач

У даному розділі було проведено огляд теоретичних засад біометричної ідентифікації. Вивчено основні поняття систем біометричної ідентифікації, розглянуто методи попереднього оброблення зображень та алгоритми розпізнавання, досліджено методи фільтрації зображень, що дозволяють знижувати дію шумів і перешкод.

Для досягнення поставленої в роботі мети, було поставлено такі задачі дослідження:

– удосконалити обраний метод фільтрації зображення на основі фільтра Габора;

– розробити алгоритм роботи програмного додатку ідентифікації користувачів;

– обґрунтувати вибір мови та середовища програмування додатку;

– спроектувати та розробити інтерфейс користувача додатку;

– розробити програмний засіб;

– обґрунтувати економічну доцільність розробки.

Отже, результатом виконання поставлених завдань є розроблення та реалізація програмного засобу для підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення на основі фільтра Габора.

2 РОЗРОБКА СИСТЕМИ ІДЕНТИФІКАЦІЇ КОРИСТУВАЧІВ ЗАСОБАМИ ТЕХНОЛОГІЇ FACE ID

2.1 Удосконалення методу фільтрування зображення для технології Face ID

Одним із головних етапів обробки біометричної інформації є попереднє опрацювання зображень.

У більшості випадків інформативнішими є геометричні характеристики межі об’єктів цих зображень – границі, які служать важливими ознаками для класифікації зображених об'єктів і сприйняття зображення в цілому [61 – 62].

Фільтрація зображень є актуальною задачею, оскільки стосується реконструкції зображення шляхом зміни інтенсивності пікселів. Знаходження зміни інтенсивності таким чином, щоб границі зображення були б чіткими, з виділенням інформаційних ознак [63 – 64].

Відомо, що для зменшення спотворень границь застосовують різні способи фільтрації [65 – 66]. Перший з них передбачає виділення матриці пікселів зображення з наступним порівнянням параметрів кожного пікселя матриці з параметрами сусідніх пікселів. При цьому використовуються кілька компараторів за результатами роботи яких, коректують параметри даного пікселя. При використанні другого способу, фільтрація автоматично застосовується до пікселів чорно-білого динамічного зображення. При цьому кутова орієнтація матриці коефіцієнтів анізотропного фільтра [67] визначається як вагова функція пікселів.

Актуальність обговорюваного питання зумовлюється недоліком наведених способів, тобто неможливістю контролю користувачем величини згладжування зображення та фіксована кількість напрямків анізотропної фільтрації, що не дозволяє у достатній мірі покращити якість зображення [68].

Для виділення границь об’єктів на зображенні розроблено велику кількість різних методів, серед яких слід зазначити такі їх класифікаційні

групування:

– як методи високочастотної фільтрації;

– методи просторового диференціювання;

– методи функціональної апроксимації [69].

Разом із цим, завдяки цим методам границі виділяються недостатньо ефективно. А саме, трапляються розмиті перепади границь зображень під час дискретизації елементів. Причому ступінь розфокусування визначається якістю системи реєстрації, кроком дискретизації та умовами освітлення, за яких зображення було отримане.

Алгоритму швидкого перетворення Фур’є для комп’ютерних систем надає можливість перенести оброблення зображень від маніпуляцій з числовими значеннями яскравості окремих пікселів до оброблення сигналу, що можна отримати після проведення швидкого перетворення Фур’є над зображенням.

Представлення зображення у вигляді Фур’є-образу надає можливість застосувати фільтри аналогічні тим, що застосовуються при обробленні сигналів. Дослідження ефективності частотних фільтрів в обробленні зображень надає важливу інформацію про особливості їх застосування.

Складність перетворення Фур’є полягає в тому, що значення коефіцієнта Фур’є залежать від цілого зображення. Для обчислення значення Фур'є-образу окремого вектора, використовуються усі пікселі зображення. Такий спосіб обробки зображення є не зовсім зручним, оскільки втрачається частина інформації.

Якщо проводити аналіз в термінах просторових частот, що визначаються тільки локально, то дане явище можна вважати зміною вмістимого зображення під час пересування по ньому. Саме тому, в деякому вікні навколо точки вузькі смуги виглядають з високою просторовою частотою, а широкі смуги – навпаки, з низькою просторовою частотою [70]. Таких недоліків не мають фільтри Габора. Оскільки, їх ядра виглядають як елементи базиса Фур’є, що помножені на гауссіани.

Таким чином, фільтри Габора дають сильну реакцію у тих точках зображення, де є компонент із локальними особливостями частоти в просторі та орієнтації.

Для побудови одновимірного фільтра Габора використовується залежність:

$$g(z) = e^{-\frac{z^2}{2a^2}}\cos(2\pi\theta z) \tag{2.1}$$

де σ – стандартне відхилення гаусівського ядра, яке визначає амплітуду функції; ω – частота коливань, яка визначається як $\omega = {}^{1}$, де T – період функції $\cos(2\pi\omega z)$.

Чи більше σ – тим форма функції стає пологішою, чим менше σ – тим форма функції буде матиме гострий пік. Одновимірні фільтри Габора з різними значеннями параметрів наведено на рис. 2.1–2.2 (за [71]).

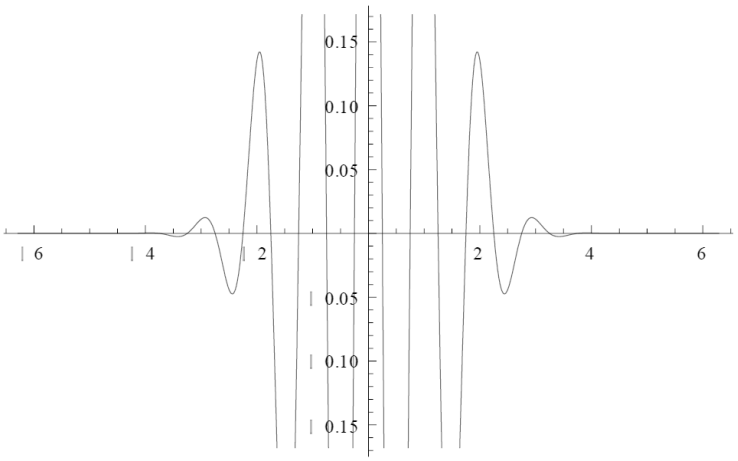


Рисунок 2.1 – Одновимірний фільтр Габора з параметрами σ = 1, θ = 1

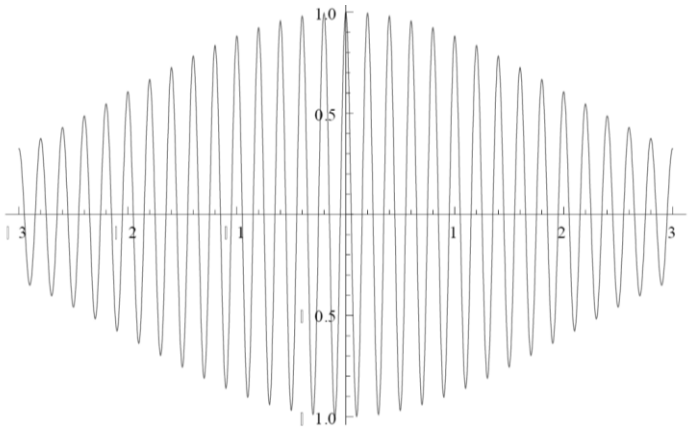


Рисунок 2.2 – Одновимірний фільтр Габора з параметрами σ = 2, θ = 5

На базі залежності (2.1) видно, що фільтр залежить від частоти і напрямків квазіперіодичної структури зображення. Зазвичай, для спрощення завдання розраховується середня частота зображення. Вона є незмінною в кожній точці.

Для побудови поля напрямків може застосовуватися кілька способів, найбільш швидким з яких є диференціальний метод.

Таким чином, маючи частоту і чотири напрямки, попередньо будуються 4 фільтра Габора по одному на кожен з напрямків. Після чого в кожній точці зображення відбувається згортка фільтра із зображенням по певній градації, що дає вихідне значення нового зображення [70].

Отже, описаний вище фільтр Габора належить до групи лінійних фільтрів з імпульсною перехідною характеристикою, що являє собою добуток Гаусіана та гармонійної функції. У випадку цифрової обробки зображень даний фільтр застосовується для розпізнавання меж об'єктів.

Таким чином, метою етапу обробки зображення фільтром Габора є виділення контрольної області. До кожної точки обраної області застосовують даний фільтр, для того, щоб отримати інформацію про фазову структуру. Перевагою фазової складової зображення є те, що вона не залежить від контрасту зображення і освітлення.

В результаті такої обробки, отримуємо образ обличчя, який далі може бути оброблений системою та використовуватись для порівняння з іншими зразками, що зберігаються в базі даних такої системи.

Для підвищення достовірності ідентифікації користувачів, здійснимо наступним етапом обробку зображення із використанням медіанної фільтрації.

Медіанна фільтрація є ефективним способом придушення імпульсних шумів, які, зокрема, неминуче з'являються в цифрових камерах в умовах малого освітлення сцени. Алгоритм медіанної фільтрації є масковим:

Для кожної точки вхідного зображення береться деяка область (наприклад, 3×3).

Точки даної області сортуються за зростанням яскравості.

Серединна (медіанна) точка (5-я для фільтра 3×3) відсортованої множини записується в підсумкове зображення.

На наступному кроці вікно пересувається на один відлік і обчислення повторюються. Крайні значення масиву дублюються стільки раз, щоб можна було застосувати вікно до першого і до останнього значенням.

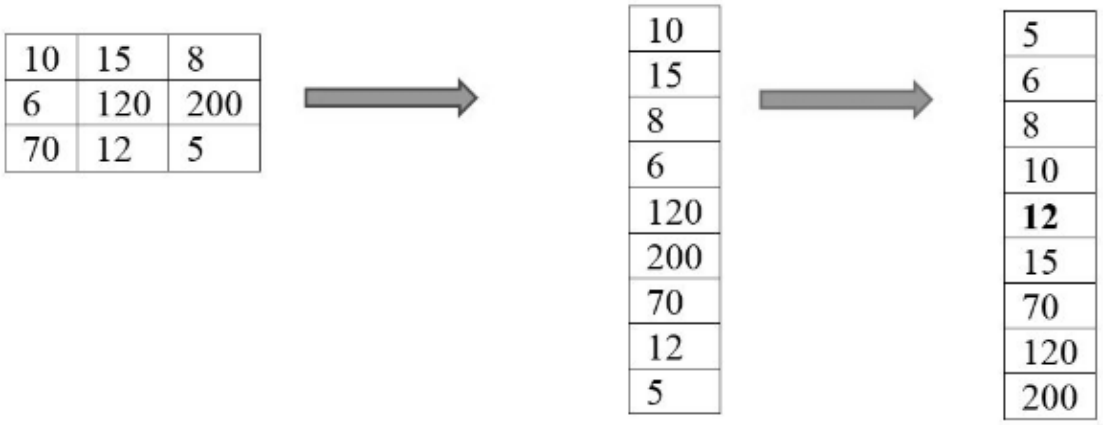


Рисунок 2.3 – Алгоритм здійснення медіанної фільтрації (за [70])

Даний алгоритм досить ресурсоємкий: так, наприклад, при обробці зображення в градаціях сірого медіанним фільтром 3×3 потрібно близько 50 операцій на одну точку зображення. Але в той же час він оперує тільки з 8-бітними числами і йому для роботи потрібно порівняно небагато вхідних даних, що робить алгоритм досить простим. Зауважимо, що, медіанна фільтрація згладжує зображення.

Таким чином, для підвищення достовірності ідентифікації користувачів засобами технології Face ID розробимо програмний додаток у якому для обробки зображення буде застосовано два етапи. На етапі фільтрації методом Габора окреслюватиметься контрольна область зображення, на етапі медіанної фільтрації у такій області будуть оброблятись та зменшуватись зайві шуми та прибиратись вади цифрового зображення, які є цілком характерними для растрових зображень здійснених на звичайну веб-камеру.

2.2 Розробка алгоритму роботи додатку для ідентифікації користувачів

Типова система автентифікації користувачів складається із таких ключових елементів елементи:

– користувач, за даними якого здійснюється процедура автентифікації;

– особлива відмінна риса даного суб'єкта – його певна біометрична характеристика;

– адміністратор системи автентифікації, що керує роботою системи та несе відповідальність за усі дії, що у ній відбуваються;

– алгоритм (принцип) роботи системи автентифікації;

– механізм управління системою автентифікації (управління доступом).

Враховуючи темпи розвитку сучасних інформаційних технологій, у системах контролю доступу використовується комплексна ідентифікація користувачів, яка базується на поєднанні та використанні декількох факторів ідентифікації, які значно впливають на захищеність та надійність системи. Проте, під час розробки та організації системи доступу з використанням певних методів ідентифікації користувачів, необхідно брати до уваги умови, що висуваються до ступеню захищеності програми, а також враховувати її економічну та кадрову ефективність. вибору для системи певного фактору чи способу автентифікації, потрібно враховувати необхідний ступінь захищеності системи, вартість її розроблення та реалізації та можливості забезпечити ефективну роботу з нею. Проілюструємо це на прикладі порівняльної табл. 2.1 [61 – 62].

Таблиця 2.1 – Порівняння способів автентифікації залежно від рівня ризику

Рівень ризику	Вимоги до системи	Спосіб автентифікації
Низький	Вимога: здійснити автентифікацію в системі.	Багаторазові Паролі

	Передумова: злом, крадіжка, розголошення конфіденційних даних не передбачають значних негативних наслідків.	
Середній	Вимога: здійснити автентифікацію в системі. Передумова: злом, крадіжка, розголошення конфіденційних даних можуть завдати певних збитків.	Одноразові паролі
Високий	Вимога: здійснити автентифікацію в системі. Передумова: злом, крадіжка, розголошення конфіденційних даних можуть завдати значних збитків.	Багатофакторна автентифікація

У магістерській роботі запропоновано застосування двох методів біометричної ідентифікації, яка порівняно з парольною (використання одноразових та багаторазових паролів) та майновою (використання апаратно-програмних систем ідентифікації і автентифікації або пристроїв введення ідентифікаційних ознак) має низку переваг, а саме:

– надійність і швидкість здійснення автентифікації; пристрої розпізнають людину протягом 1 – 2 с;

– високий рівень безпеки даних автентифікації: біометричні ознаки людини є неповторними, що зводить до мінімуму кількість можливих помилок;

– дані використовуваних біометричних характеристик не можна втратити або забути;

– пристрої для біометричної автентифікації зручні в експлуатації.

Таким чином, враховуючи усі переваги автентифікації на основі біометричної ідентифікації, оберемо такі біометричні унікальні дані, як обличчя користувача та його відбиток пальця. В дані роботі увага приділяється саме ідентифікації користувача за його обличчям, що використовується на основі вдосконаленого методу, розглянемо більш детальніше алгоритм роботи розроблюваного додатку. Даний алгоритм представлено схематично на рис. 2.1.

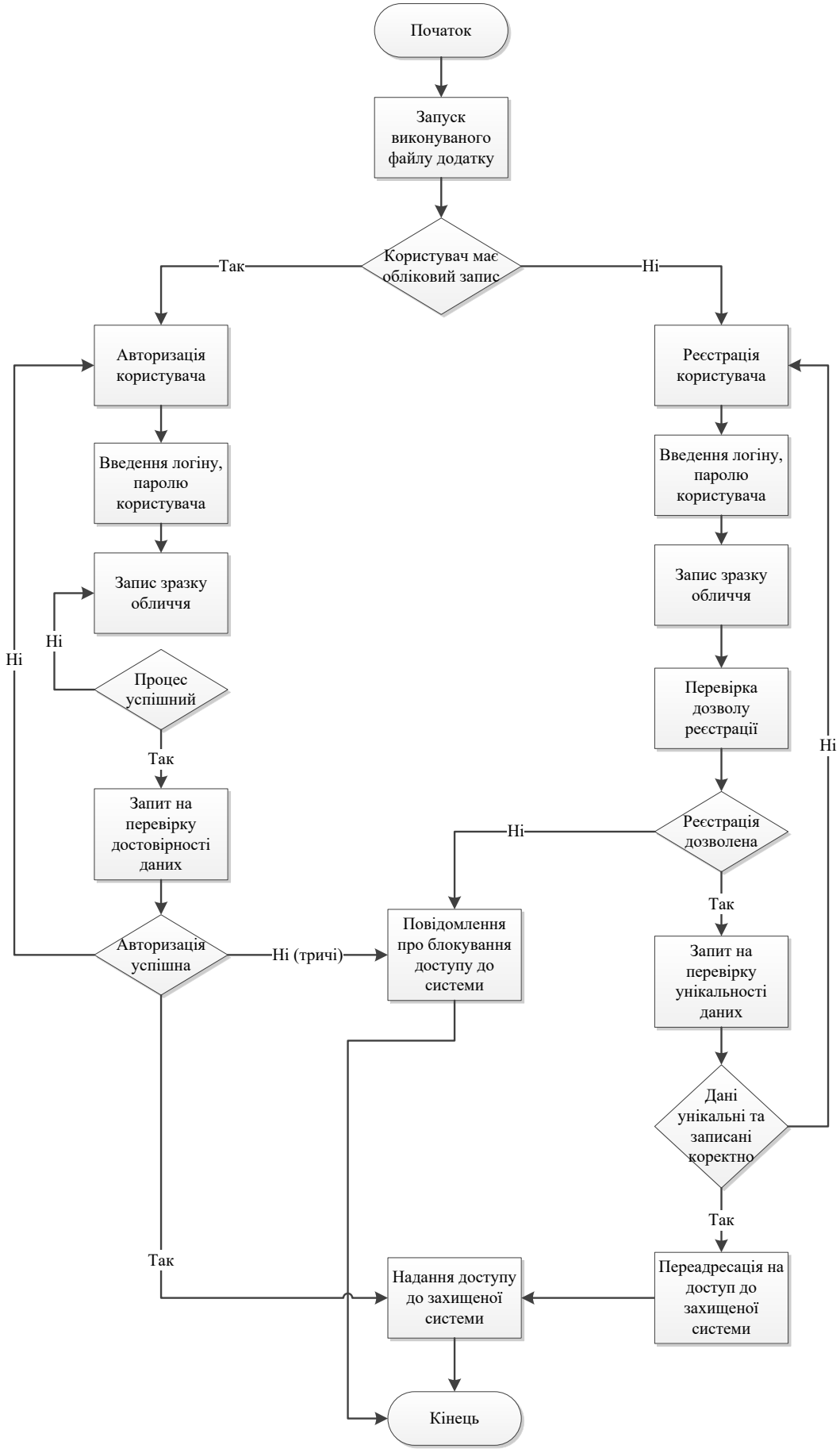


Рисунок 2.1 – Алгоритм роботи додатку

Крок 1. Запуску файлу додатку.

Крок 2. Якщо користувач вперше використовує додаток, йому необхідно пройти етап реєстрації. Для реєстрації нового користувача в системі, суб'єкту потрібно ввести логін (має бути унікальний та не повторюватись в межах згаданої системи) та зафіксувати з допомогою веб-камери зразок (образ) свого обличчя.

Крок 2.1. Для здійснення процесу реєстрації користувач повинен натиснути кнопку «Реєстрація» у верхній частині головного діалогового вікна, ввести у відповідне поле свій логін, парол та показати у веб-камеру своє обличчя. Після цього натиснути кнопку «Зареєструватися».

Крок 2.2 Система автоматично перевіряє чи доступний вказаний логін для реєстрації в системі. Якщо логін погоджений системою, отже користувач має повноваження отримати обліковий запис у такій системі. Далі перехід до кроку 2.3.

У випадку, якщо система не розпізнає вказаний логін, користувачу подальші дії для роботи з системою заблоковані.

Крок 2.3 Система автоматично перевіряє унікальність отриманих даних із даними у базі даних та вносить нового користувача.

Крок 2.4. Користувачеві у новому діалоговому вікні відкривається повідомлення про успішну реєстрацію.

Крок 2.5. Доступ до захищуваних в системі даних дозволений.

Крок 3. Авторизація в системі. Функція, якою можуть користуватись зареєстровані (вже відомі системі користувачі).

Крок 3.1. Для запуску процесу авторизації користувачеві необхідно скористатись кнопкою «Війти», що розташована у верхній частині головного діалогового вікна та ввести свої користувацькі дані (логін) та зчитати образ обличчя (з використанням веб-камери).

Крок 3.2. Система автоматично перевіряє отримані дані від користувача із вже наявними в базі даних. Якщо отриманий від користувача логін вже відомий системі, а зчитаний образ обличчя відповідає образу, що зафіксований в

системі, то користувач переходить до наступного кроку.

Важливо, що система перевіряє не лише наявність образу обличчя та логіну у базі даних, але і те, що саме конкретному логіну відповідає конкретний образ обличчя.

Крок. 3.3 Система повідомляє користувача про успішну авторизацію.

Крок 3.4. Доступ до захищуваних в системі даних дозволений.

Крок 4. У випадку невдалої авторизації (логін та/або образ обличчя не існує, отриманий образ обличчя не відповідає вказаному логіну або навпаки, логін нового користувача не відомий системі, некоректнй пароль), користувач отримує повідомлення про те, що авторизація невдала.

Важливо, що користувач має можливість лиш трьох спроб авторизації. Після кожної невдалої спроби, система повідомляє користувачеві про це та вказує доступну кількість можливих спроб. У випадку, якщо користувач використав усі три спроби, але так і не авторизувався в системі – доступ та можливість авторизуватись для нього буде заблокований. Внести зміни та надати можливість подальшого користування може лише адміністратор (керуючий) системою.

Права керування системою належать адміністратору, який має змогу здійснювати редагування існуючих облікових записів та додавати нові.

Варто зазначити, що алгоритмом роботи передбачено додати можливість ідентифікації користувачів за відбитком пальця. Такий додатковий етап ідентифікації користувача підвищить рівень захищеності системи, а дані користувача для авторизації матимуть ще більшу унікальність.

Також, аналізуючи літературні джерела даної галузі, варто зазначити, що саме комбіновані моделі ідентифікації користувачів мають більшу ефективність порівняно з способами ідентифікації некомбінованими чи парольними.

Отже, під час написання даного підрозділу було розроблено алгоритм додатку для комплексної автентифікації в системі захисту з використанням ідентифікації за обличчям.

2.3 Обґрунтування вибору мови та засобу програмування

Виходячи із теми роботи та поставлених завдань, для розробки програми підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення, було обрано мову об’єктно-орієнтованого програмування C# та середовище Visual Studio для програмної реалізації додатку. Розглянемо більш детальніше обрані та застосовані в роботі засоби програмування.

C# [72] – це сучасна мова програмування загального призначення, яка може бути використана для виконання широкого кола завдань та завдань, що охоплюють різні професії. C# в основному використовується на платформі Windows .NET, хоча його можна застосувати до платформи з відкритим кодом. Ця надзвичайно універсальна мова програмування – це об’єктно-орієнтована мова програмування (ООП), яка є вже досить поширеною.

C# – чудовий вибір для розробників із середнім і великим досвідом написання коду. Незважаючи на те, що експерти визнають, що мова є однією з помірних складностей, вони погоджуються, що її досить просто зрозуміти та досягти успіху. Після того, як ви познайомитеся з C# і вкладете час, щоб його зрозуміти, ви можете розраховувати на швидке просування від новачка до експерта [72].

Це пов’язано з тим, що C# – мова високого рівня, а це означає, що її порівняно легко читати та писати, що робить її надійним вибором для початківців та зручним варіантом для експертів. Окрім читабельності, C# може також використовуватися для автоматизації складних завдань, що вимагають багато часу для досягнення незначних результатів.

Ця мова програмування також статистично набрана, що означає, що помилки виявляються до запуску програми. Це значно полегшує виявлення невеликих вад у вашому стеку, які в іншому випадку були б майже непомітними–не кажучи вже про неймовірно обтяжуючі.

Незважаючи на те, що C# може бути ефективно використаний в руках

усіх типів програмістів, значну частину користувацької бази мови складають ті, хто прихильний до платформи Microsoft.

Як і інші мови програмування загального призначення, C# може використовуватися для створення цілого ряду різних програм та додатків: мобільних додатків, настільних програм, хмарних служб, веб-сайтів, корпоративного програмного забезпечення та ігор. Хоча C# надзвичайна універсальна мова, є три сфери, в яких вона найчастіше використовується.

Розробка веб-сайтів. C# часто використовується для розробки професійних, динамічних веб-сайтів на платформі .NET або програмного забезпечення з відкритим кодом. Отже, навіть якщо ви не прихильник архітектури Microsoft, ви все одно можете використовувати C# для створення повнофункціонального веб-сайту. Оскільки ця мова є об’єктно-орієнтованою, її часто використовують для розробки веб-сайтів, які є неймовірно ефективними, легко масштабованими та легко підтримуються.

Програми Windows. C# був створений Microsoft для Microsoft, тому легко зрозуміти, чому він найпопулярніший для розробки настільних додатків Windows. Додатки C# вимагають платформи Windows .NET для найкращого функціонування, тому найефективнішим варіантом використання цієї мови є розробка програм, які специфічні для архітектури платформи Microsoft.

Ігри. C# може бути просто однією з найкращих мов програмування для ігор. Ця мова активно використовується для створення улюблених уболівальниками ігор, таких як Rimworld на Unity Game Engine.

Ось декілька програм та програм, написаних на C#, які демонструють різноманітність та надійність мови: Windows Installer XML; Microsoft Visual Studio; Paint.NET; Open Dental; KeePass; FlashDevelop; Banshee; NMath; Pinta; OpenRA.

Переваги C# [72]:

C# надає як новим, так і досвідченим програмістам безліч різних переваг. Нижче ми описали наші п’ять улюблених переваг.

C# може заощадити ваш час. Мабуть, найбільшою перевагою є те, скільки

часу можна заощадити, використовуючи C# замість іншої мови програмування. Оскільки C# набирається статично і легко читається, користувачі можуть витрачати менше часу на пошук своїх скриптів на наявність крихітних помилок, які порушують роботу програми.

C# також підкреслює простоту та ефективність, тому програмісти можуть витрачати менше часу на написання складних стеків коду, які неодноразово використовуються протягом проекту.

Досить швидке навчання. Окрім часу, який можна заощадити під час розробки проекту, також витрачається менше часу на вивчення C# на відміну від більш складних мов програмування. Завдяки простоті та простим у користуванні функціям, C# пропонує досить низький рівень навчання для початківців.

Ця мова робить чудовий перший крок у цій галузі та надає початківцям розробникам зручний спосіб ознайомитися з програмуванням, не розчаровуючись і не переживаючи.

Це масштабована мова, яку легко підтримувати. C# – це мова програмування, яка надзвичайно масштабована і проста в обслуговуванні. Через сувору природу написання статичних кодів програми C# надійно узгоджується, що робить їх набагато простішими в налаштуванні та обслуговуванні, ніж програми, написані іншими мовами [72].

Якщо доведеться повернутися до старого проекту, написаного на мові C#, можна побачити, що, хоча процеси можуть змінюватися з роками, стек C# залишається незмінним на всьому рівні.

Існує велика спільнота. У світі кодування та програмування важливість корисної спільноти, від якої ви можете залежати, просто не можна переоцінити. Мови програмування – це не платформа чи послуга із спеціальною лінією довідки чи зручною IT-підтримкою. Програмісти повинні розраховувати на підтримку інших у тій же галузі, які зазнали однакових перешкод і розчарувань.

C# є об'єктно-орієнтованим. C# повністю об'єктно-орієнтований, що є рідкісною характеристикою мови програмування. Багато найпоширеніших мов

до певної міри включають об'єктну орієнтацію, але дуже мало хто досягнув величини C#, не втрачаючи прихильності користувачів.

Об’єктно-орієнтоване програмування (або ООП) має багато різних переваг, таких як ефективність та гнучкість, щоб назвати декілька. Деякі розробники, які не знайомі з ООП, можуть почуватись небажано вибирати нову мову з таким серйозним акцентом, але не хвилюйтеся: зрозуміти об’єктно-орієнтоване програмування не все так складно.

Для роботи з налаштуванням та обробкою розпізнавання райдужної оболонки ока, планується застосувати бібліотеку комп’ютерного зору OpenCV.

OpenCV [73] (Open Source Computer Vision) – бібліотека комп'ютерного зору з відкритим вихідним кодом) – бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Реалізована на C / C ++.

Ця бібліотека дуже популярна за рахунок своєї відкритості та можливості безкоштовно використовувати як в навчальних, так і комерційних цілях.

Фактично, OpenCV – це набір типів даних, функцій і класів для обробки зображень алгоритмами комп'ютерного зору.

Основні модулі бібліотеки:

Cxcore – ядро містить базові структури даних і алгоритми:

– базові операції над багатовимірними числовими масивами;

– матрична алгебра, математичні ф-ції, генератори випадкових чисел;

– запис / відновлення структур даних в / з XML;

– базові функції 2D графіки.

CV – модуль обробки зображень і комп'ютерного зору:

– базові операції над зображеннями (фільтрація, геометричні перетворення, перетворення колірних просторів і т. д.);

– аналіз зображень (вибір відмінних ознак, морфологія, пошук контурів, гістограми);

– аналіз руху, спостереження за об'єктами; виявлення об'єктів, зокрема осіб;

– калібрування камер, елементи відновлення просторової структури.

Highgui – модуль для введення / виведення зображень і відео, створення призначеного для користувача інтерфейсу: захоплення відео з камер і з відео файлів, читання / запис статичних зображень; функції для організації простого UI (всі демо додатки використовують HighGUI).

Cvaux – експериментальні і застарілі функції:

– стерео калібрації, саме калібрації;

– пошук стерео-відповідності, кліки в графах;

– знаходження і опис рис обличчя.

CvCam – захоплення відео: дозволяє здійснювати захоплення відео з цифрових відео-камер (підтримка припинена і в останніх версіях цей модуль відсутній).

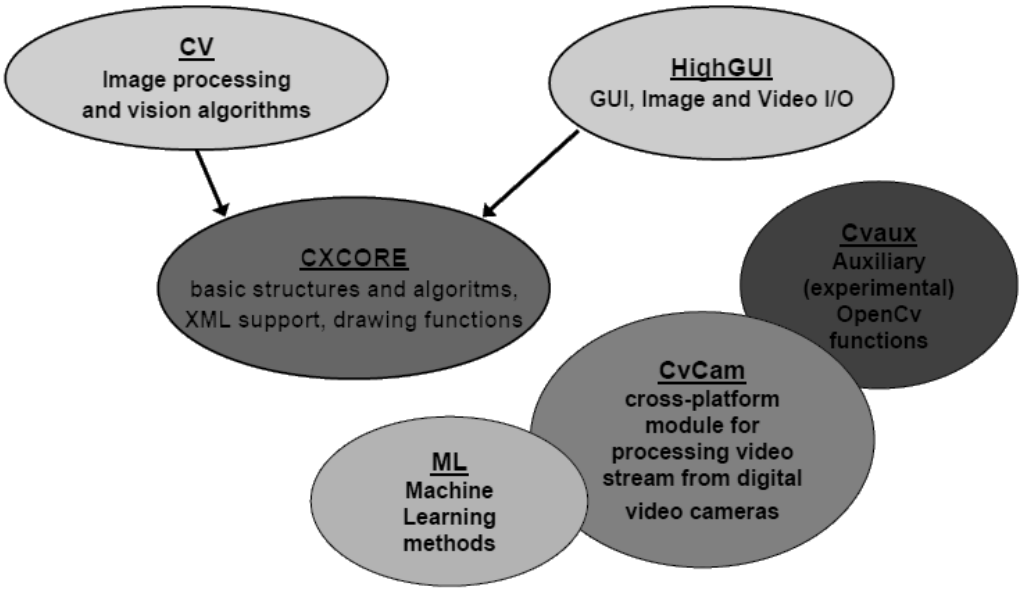


Рисунок 2.8 – Структура бібліотеки комп’ютерного зору OpenCV (за [73])

Інтегроване середовище розробки Visual Studio – це стартовий майданчик для написання, налагодження і складання коду, а також подальшої публікації додатків.

Інтегроване середовище розробки (IDE) являє собою багатофункціональну програму, яку можна використовувати для різних аспектів розробки програмного забезпечення.

Крім стандартного редактора і відладчика, які існують в більшості середовищ IDE, Visual Studio включає в себе компілятори, засоби автозавершения коду, графічні конструктори і багато інших функцій для спрощення процесу розробки [74].

Таким чином, оцінивши можливості та переваги обраних засобів, були обрані відповідні засоби програмної реалізації програми комплексної автентифікації до системи захисту з використанням ідентифікації за обличчям користувача та його логіном.

2.4 Висновки до розділу

У даному розділі здійснювалося розроблення програмного засобу для підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення.

Під час роботи було розглянуто особливості обраної технології, розроблено алгоритм роботи додатка.

Для ефективного оброблення зображень автором роботи запропоновано використовувати фільтрацію на основі методу фільтрів Габора, який дозволяє створити зображення з більш чіткими контурами, та подальшою обробкою на основі медіанної фільтрації, що дозволяє покращити якість зображення шляхом придушення імпульсних шумів, які, зокрема, неминуче з'являються в цифрових камерах в умовах малого освітлення.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ КОМПЛЕКСНОЇ АВТЕНТИФІКАЦІЇ ЗА БІОМЕТРИЧНИМИ ДАНИМИ

3.1 Проектування графічного користувацького інтерфейсу

GUI (Graphical User Interface) – це один із різновидів інтерфейсів користувача, елементи якого виконані у вигляді графічних зображень. Тобто, всі основні об'єкти, присутні в цьому інтерфейсі – іконки, функціональні кнопки, об'єкти меню і т.д. виконані у вигляді зображень.

Якщо порівняти GUI зі звичайним командним рядком, то в першому варіанті перед користувачем відкривається повний доступ до абсолютно всіх елементів, які він бачить на дисплеї. Реалізувати цей доступ можна за допомогою різних пристроїв введення: оптичної миші, трекболу, клавіатури, джойстика та ін.

Зазвичай у GUI кожен графічний об'єкт передає сенс функції за допомогою зрозумілого образу, щоб користувачеві було простіше розібратися з певним програмним забезпеченням та легше взаємодіяти з операційною системою загалом. Але важливо розуміти, що GUI – це лише складова частина графічного інтерфейсу. Функціонує він на рівні візуалізації даних і таким чином взаємодіє з користувачем.

Центральною ланкою при проектуванні графічного інтерфейсу є користувач.

Керівні принципи служать розробникам основою побудови GUI інтерфейсу. При прийнятті будь-яких проектних рішень щодо інтерфейсу GUI вони повинні використовуватися розробниками на підсвідомому рівні.

До таких основних принципів належать:

– контроль на стороні користувача – основний сенс цього принципу полягає в тому, що користувач ініціює дії, і якщо в результаті цього контроль переходить до програми, користувач отримує необхідний зворотний зв'язок;

– узгодженість, безперечно, є другим основним принципом розробки

якісного інтерфейсу. Фактично узгодженість означає дотримання стандартів і дотримання деяких загальноприйнятих правил роботи з графічим інтерфейсом;

– індивідуалізація та налаштування – це два взаємопов'язані принципи розробки графічного інтерфейсу. Індивідуалізація GUI інтерфейсу – це просто його налаштування під персональні потреби, тоді як налаштування – так, як ми розуміємо їх тут – адміністративне завдання пристосування програмного забезпечення до вимог різних груп користувачів;

– естетичність інтерфейсу впливає зорове сприйняття системи, зручність стосується легкості, простоти, ефективності, надійності та продуктивності використання інтерфейсу.

Розглянемо детальніше особливості проектування діалогових вікон розроблюваного додатку на основі біометричної ідентифікації користувачів.

Отже, після запуску виконуваного файлу додатку – відкривається головне діалогове вікно. Розглянемо його структуру.

У верхній частині вікна заплановано розташування двох важливих функціональних кнопок «Вхід» та «Реєстрація». Користувач обирає одне з них, залежно від здійснюваної ним операції.

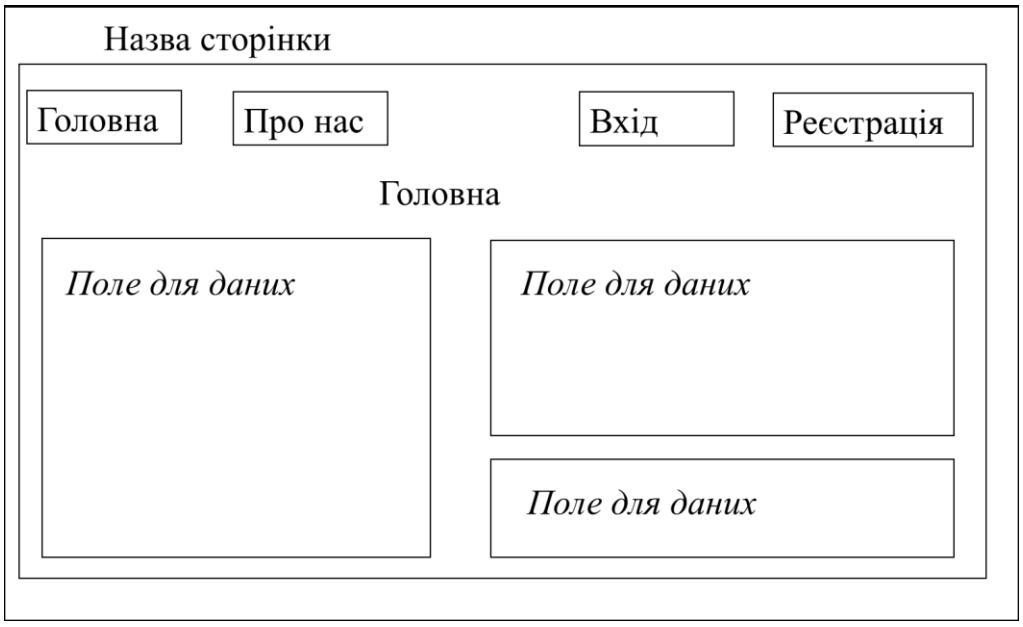


Рисунок 3.1 – Проектування головного вікна додатку

У наступному вікні розміщене поле для вводу логіну з відповідним

позначенням, а також поле «Зразок обличчя», яким потрібно скористатися користувачеві та дати можливість веб-камері записати зразок обличчя, який буде записаний додатком та використаний для подальшої роботи.

У нижній частині вікна розташована кнопка «Зареєструватися».

Її користувач натискає як підтвердження здійснення етапів ідентифікації та подання команди додатку продовжувати роботу далі, тим самим запустивши процес авторизації.

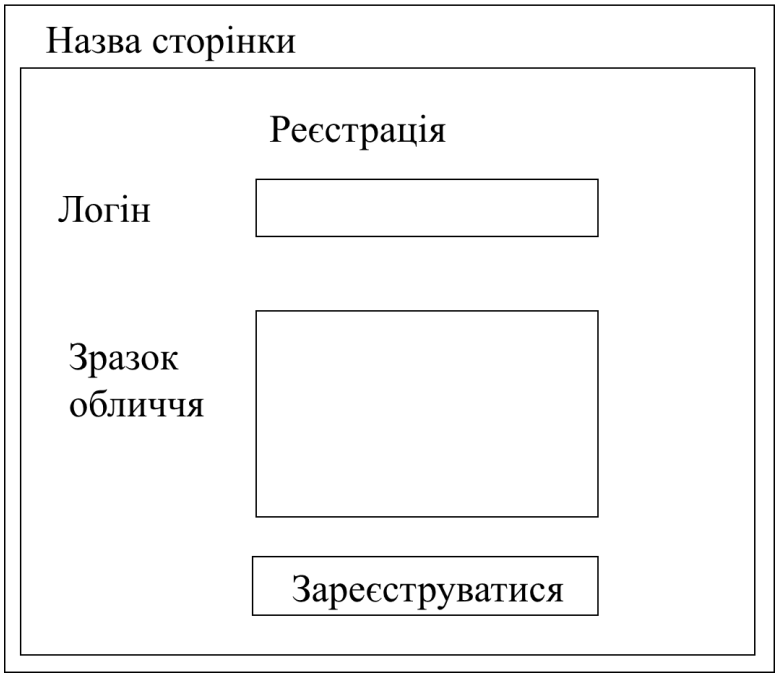


Рисунок 3.2 – Проектування вікна реєстрації (входу) додатку

Після того, як користувачем здійснено успішний процес авторизації або реєстрації в системі, він отримує доступ до неї.

В вікні, що відображає доступ до системи у верхній частині планується розміщення кнопки «Про нас», за натисненням на яку відбувається виведення ще одного діалогового вікна з даними про автора роботи, також може бути вказаний зворотний зв'язок і т. п.

Також в центрі верхньої частини розташоване поле, у якому відображається логін користувача, якщо авторизація здійснена.

Якщо ж користувач скористається сусідньою кнопкою виходу, замість логіну буде вказано, що користувач не є авторизованим – робота в системі буде

заблокована.

Далі у основній частині вікна розташовується певне зображення та текст, який відображатиме приклад самої системи.

Рисунок 3.3 – Вигляд вікна захищуваної системи

Таким чином, у даному підрозділі було викладено основні вимоги до побудови графічного користувацького інтерфейсу, спроектовано ключові діалогові вікна додатку, які дозволяють у повній мірі розмістити та відобразити функціонал, який зумовлений задачами даної роботи.

Далі розглянемо особливості програмної реалізації розробки та створення графічного користувацького інтерфейсу.

3.2 Програмна реалізація додатку

Аналізуючи поставлені задачі, для програмної реалізації додатку автором вирішено використовувати середовище Visual Studio, мову об’єктно-орієнтованого програмування C# та та бібліотеку комп’ютерного зору OpenCV.

Розглянемо деякі моменти програмної реалізації розроблюваного додатку.

Бібліотеки, що використовуються у програмній реалізації додатку:

```
using System.Windows.Controls;
using System.IO;
using System.Runtime.CompilerServices;
using System.Timers;
using System.Windows;
using System.Windows.Media.Imaging;
using System.Windows.Threading;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Timers;
using System.Windows;
using System.Windows.Media.Imaging;
using System.Windows.Threading;
```

Для користування додатком для розпізнавання користувачів, інтерфейсом передбачений ряд кнопок, що дозволяють користувачеві реалізовувати певні команди. Наведемо приклад реалізації деяких з них.

Запуск вихідного файлу додатку:

```
public CommonPage()
    {
      InitializeComponent();
    }
```

Кнопка «Про нас», що відображає дані про систему та розробника.

```
    private void AboutButton_Click(object sender, RoutedEventArgs e)
    {
      WFAbout wfAbout = new WFAbout();
      wfAbout.ShowDialog();
    }
```

Кнопка реєстрації нового користувача:

```
    private void Registration_Click(object sender, RoutedEventArgs e)
    {
      var w = Application.Current.Windows[0];
      w.Hide();

      Registration wfAbout = new Registration();
      wfAbout.ShowDialog();

      w.Close();
    }
```

Кнопка входу для зареєстрованого користувача в системі:

```
private void Login_Click(object sender, RoutedEventArgs e)
{
   var w = Application.Current.Windows[0];
   w.Hide();

   Login wfAbout = new Login();
   wfAbout.ShowDialog();

   w.Close();
}
```

Кнопка виходу з системи авторизованого користувача:

```
private void Exit_Click(object sender, RoutedEventArgs e)
{
   Registration.Visibility = Login.Visibility = Visibility.Visible;
   Exit.Visibility = Visibility.Hidden;

   AuthUserName.Content = "Не авторизовано";
}
```

Звернення до класу реєстрації користувача:

```
public partial class Registration : Window, INotifyPropertyChanged
  {
   public Registration()
   {
      InitializeComponent();
      captureTimer1 = new Timer()
      {
         Interval = Config.TimerResponseValue
      };
      captureTimer1.Elapsed += captureTimer1_Elapsed;
   }
```

Запит на введення даних нового користувача:

```
public string FaceName
   {
      get { return faceName; }
      set
      {
         faceName = value.ToUpper();
         lblFaceName.Dispatcher.Invoke(DispatcherPriority.Normal,
            new Action(() => { lblFaceName.Content = faceName; }));
         NotifyPropertyChanged();
      }
   }
```

Звернення до веб-камери пристрою для зчитування образу обличчя:

```
public Bitmap CameraCapture
    {
      get { return cameraCapture; }
      set
      {
        cameraCapture = value;
        imgCamera.Dispatcher.Invoke(DispatcherPriority.Normal,
       new Action(() => { imgCamera.Source = BitmapToImageSource(cameraCapture); }));
        NotifyPropertyChanged();
      }
    }
```

Реєстрація образу обличчя веб-камерою пристрою:

```
public Bitmap CameraCaptureFace
    {
      get { return cameraCaptureFace; }
      set
      {
        cameraCaptureFace = value;
        // imgDetectFace.Dispatcher.Invoke(DispatcherPriority.Normal,
        //                 new Action(() => { imgDetectFace.Source =
BitmapToImageSource(cameraCaptureFace); }));
        // imgCamera.Source = BitmapToImageSource(cameraCapture);
        NotifyPropertyChanged();
      }
    }
```

Запис даних FaceID у систему:

```
public void OnLoad()
    {
      GetFacesList();
      if (videoCapture == null)
      {
        videoCapture = new VideoCapture(Config.ActiveCameraIndex);
        videoCapture.SetCaptureProperty(CapProp.Fps, 30);
        videoCapture.SetCaptureProperty(CapProp.FrameHeight, 500);
        videoCapture.SetCaptureProperty(CapProp.FrameWidth, 400);
        captureTimer1.Start();
      }
    }
```

Клас класифікації при реєстрації нового користувача:

```
    protected virtual void NotifyPropertyChanged([CallerMemberName] String
propertyName = "")
    {
      var handler = PropertyChanged;
      if (handler != null) handler(this, new PropertyChangedEventArgs(propertyName));
    }
```

Виведення результатів зчитування:

```
public void GetFacesList()
    {
      if (!File.Exists(Config.HaarCascadePath))
      {
        string text = "Cannot find Haar cascade data file:\n\n";
        text += Config.HaarCascadePath;
        MessageBoxResult result = MessageBox.Show(text, "Error",
          MessageBoxButton.OK, MessageBoxImage.Error);
      }
```

Звернення до директорії:

```
      haarCascade = new CascadeClassifier(Config.HaarCascadePath);
      faceList.Clear();
      string line;
      FaceData faceInstance = null;
      if (!Directory.Exists(Config.FacePhotosPath))
      {
        Directory.CreateDirectory(Config.FacePhotosPath);
      }
      if (!File.Exists(Config.FaceListTextFile))
      {
        string text = "Cannot find face data file:\n\n";
        text += Config.FaceListTextFile + "\n\n";
        text += "If this is your first time running the app, an empty file will be created for
you.";
        MessageBoxResult result = MessageBox.Show(text, "Warning",
          MessageBoxButton.OK, MessageBoxImage.Warning);
        switch (result)
        {
          case MessageBoxResult.OK:
            String dirName = Path.GetDirectoryName(Config.FaceListTextFile);
            Directory.CreateDirectory(dirName);
            File.Create(Config.FaceListTextFile).Close();
            break;
        }
      }
```

Оброблення зображення образу обличчя в системі:

```
public void ProcessFrame()
    {
      if (active)
      {
        bgrFrame = videoCapture.QueryFrame().ToImage<Bgr, Byte>();
        if (bgrFrame != null)
        {
          try
```

```
            {
                Image<Gray, byte> grayframe = bgrFrame.Convert<Gray, byte>();
                Rectangle[] faces = haarCascade.DetectMultiScale(grayframe, 1.2, 10,
                    new System.Drawing.Size(50, 50), new System.Drawing.Size(200, 200));
```

Функція розпізнавання образу обличчя зафіксованого в системі:

```
                FaceName = "No face detected";
                foreach (var face in faces)
                {
                    bgrFrame.Draw(face, new Bgr(255, 255, 0), 2);
                    detectedFace = bgrFrame.Copy(face).Convert<Gray, byte>();
                    FaceRecognition();
                    break;
                }
                CameraCapture = bgrFrame.ToBitmap();
            }
            catch (Exception ex)
            {
```

Перевірка доданого зразку образу обличчя:

```
private void FaceRecognition()
        {
            if (imageList.Size != 0)
            {
      FaceRecognizer.PredictionResult result = recognizer.Predict(detectedFace.Resize(100,
100, Inter.Cubic));
                FaceName = nameList[result.Label];
                CameraCaptureFace = detectedFace.ToBitmap();
            }
            else
            {
                FaceName = "Please Add Face";
            }
```

Верифікація даних реєстрації:

```
private bool VerifyData()
        {
            if (String.IsNullOrEmpty(UserName.Text))
            {
                MessageBox.Show("Введіть логін");
                return false;
            }
            if (imgCamera.Source == null)
            {
                MessageBox.Show("Сталася помилка. Зразук обличчя не вдалося отримати");
                return false;
            }
            return true;
```

```
    }
```

Завершення процесу реєстрації:

```
  private void Registration_OnClosed(object sender, EventArgs e)
    {
      active = false;
      if (enableAutoNavigate)
      {
        captureTimer1 = null;
        var w = Application.Current.Windows[0];
        w.Hide();
        CommonPage wfAbout = new CommonPage();
        wfAbout.ShowDialog();
        w.Close();
      }
```

Запит на зчитування образу обличчя користувача:

```
private void OpenVideoFile_Click(object sender, RoutedEventArgs e)
    {
      OpenFileDialog openDialog = new OpenFileDialog();
      if (openDialog.ShowDialog().Value == true)
      {
        captureTimer.Stop();
        videoCapture.Dispose();
        videoCapture = new VideoCapture(openDialog.FileName);
        captureTimer.Start();
        this.Title = openDialog.FileName;
        return;
      }
    }
```

Перевірка коректності зчитаного образу обличчя:

```
public void GetFacesList()
    {
      if (!File.Exists(Config.HaarCascadePath))
      {
        string text = "Cannot find Haar cascade data file:\n\n";
        text += Config.HaarCascadePath;
        MessageBoxResult result = MessageBox.Show(text, "Error",
          MessageBoxButton.OK, MessageBoxImage.Error);
      }
```

Процес розпізнавання зафіксованого зразку:

```
private void ProcessFrame()
    {
      if (active)
      {
```

```
            bgrFrame = videoCapture.QueryFrame().ToImage<Bgr, Byte>();
            if (bgrFrame != null)
            {
              try
              {
                FaceName = "No face detected";
                foreach (var face in faces)
                {
                  bgrFrame.Draw(face, new Bgr(255, 255, 0), 2);
                  detectedFace = bgrFrame.Copy(face).Convert<Gray, byte>();
                  FaceRecognition();
                  break;
                }
                CameraCapture = bgrFrame.ToBitmap();
              }
              catch (Exception ex)
              {
                throw new ArgumentException("Error: " + ex);
              }
```

Перевірка коректності даних зафіксованого зразку:

```
private void FaceRecognition()
    {
      if (imageList.Size != 0)
      {
        FaceRecognizer.PredictionResult result = recognizer.Predict(detectedFace.Resize(100, 100, Inter.Cubic));
        FaceName = nameList[result.Label];
        CameraCaptureFace = detectedFace.ToBitmap();
      }
      else
      {
        FaceName = "Please Add Face";
      }
    }
```

Верифікація даних авторизації:

```
private bool VerifyData()
    {
      if (String.IsNullOrEmpty(UserName.Text))
      {
        MessageBox.Show("Введіть логін");
        return false;
      }
      if (imgCamera.Source == null)
      {
        MessageBox.Show("Сталася помилка. Зразук обличчя не вдалося отримати");
        return false;
```

```
        }

        return true;
    }
```

Завершення процесу авторизації:

```
private void Login_OnClosed(object sender, EventArgs e)
    {
        captureTimer = null;
        active = false;
        if (enableAutoNavigate)
        {
            var w = Application.Current.Windows[0];
            w.Hide();
            CommonPage wfAbout = new CommonPage();
            wfAbout.ShowDialog();
            w.Close();
        }
    }
```

Отже, аналізуючи наведені зразки програмного коду, можна вважати, що обрані засоби програмування дозволяють досягти мети роботи: практично реалізувати додаток для підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення.

3.3 Реалізація користувацького інтерфейсу

Здійснивши аналіз поставлених задачі роботи, необхідний функціонал програмного додатку (системи контролю доступу) та розроблений проект графічного інтерфейсу, далі у підрозділі опишемо практичну реалізацію згадуваного інтерфейсу користувача, що надає останньому зручну та повноцінну роботу з додатком.

Отже, як зазначалося вище, після запуску виконуваного файлу додатку, відкривається головне діалогове вікно програми, у якому реалізована основна робота ідентифікації та автентифікації користувачів.

У верхній частині вікна розташовані кнопки «Реєстрація» та «Вхід», які підсвічуються сірим кольором, залежно від обраної користувачем опції.

Рисунок 3.4 – Головне діалогове вікно додатку

Розпочнемо з аналізу здійснення процесу реєстрації.

Після натиснення кнопки «Реєстрація», відкривається відповідне діалогове вікно. У полі логін та пароль, користувач повинен ввести свої облікові дані, які повинні бути унікальними. В наступному рядку реалізоване поле для зчитування зображення обличчя веб-камерою. У накреслених контурах відображається знайдене системою за допомогою веб-камери обличчя користувача.

Нижче розташована кнопка «Зареєструватися». Вигляд даного діалогового вікна наведено на рис. 3.5.

Якщо користувачем обрана кнопка «Вхід», далі за аналогією із функціоналом входу заповнюється поле логіну.

У нижній частині вікна також знаходиться область відображення зображенням з веб-камери. У контурі квадрата відображається знайдене системою обличчя.

Якщо у випадку реєстрації, зчитане зображення веб-камери зберігається у базі даних та використовується для створення нового користувача, то у випадку входу, зчитане веб-камерою зображення порівнюється із зображенням у базі даних, що відповідає заявленому логіну.

Вигляд даного діалогового вікна наведено на рис. 3.6.

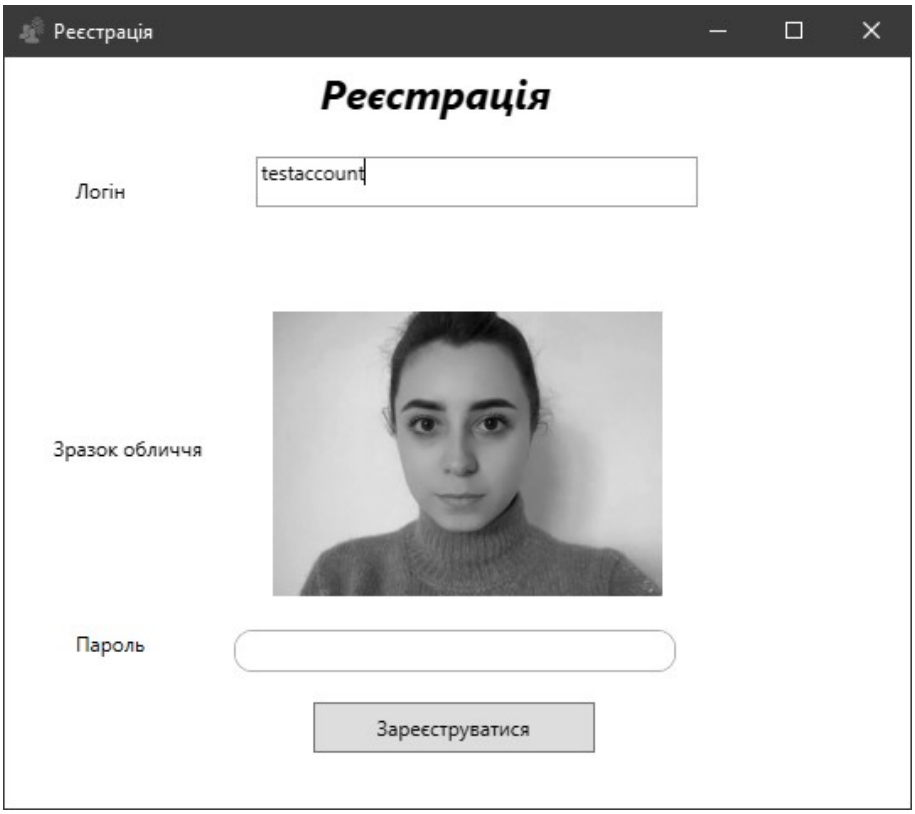


Рисунок 3.5 – Вигляд головного вікна додатку з активною кнопкою «Зареєструватися»

Рисунок 3.6 – Вигляд вікна додатку з активною кнопкою «Вхід»

Після того, як необхідні поля та дані введено, користувачеві потрібно натиснути кнопку «Увійти» для запуску процесу авторизації та продовження роботи з додатком. Якщо під час реєстрації введено унікальний логін та зображення обличчя є достовірним, то відбувається процес реєстрації користувача. Про успішну авторизацію користувач отримує відповідне повідомлення (рис. 3.7).

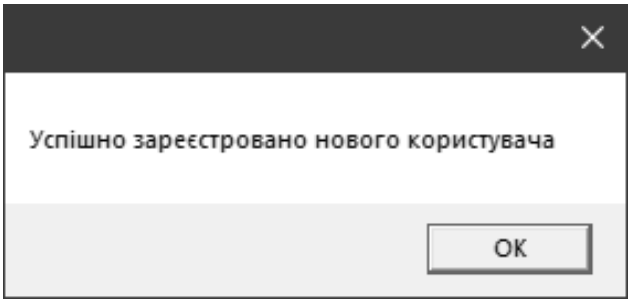


Рисунок 3.7 – Вигляд вікна з повідомленням про успішну реєстрацію

У випадку, якщо логін користувача для реєстрації не зазначений у базі даних системи як можливий до реєстарції, користувач не має можливості продовжити подальшу роботу та отримати доступ.

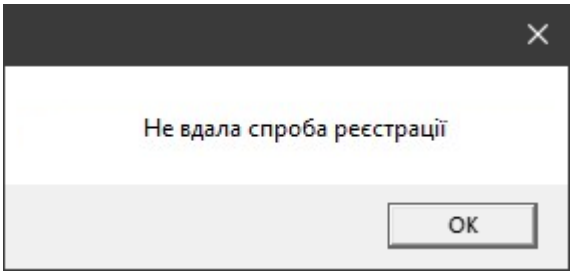


Рисунок 3.8 – Вигляд вікна з повідомленням про успішну реєстрацію

Якщо логін користувача рахується дозволеним до реєстрації, то відбувається подальша робота з системою. Після того, як реєстрація (вхід) здійснена, відкривається вікно захищуваної системи, у якому можна побачити відповідне зображення, текст, зазначення логіну авторизованого користувача (рис. 3.9), а також функціональні кнопки «Про нас» та «Вихід». Головна особливість полягає у відображені логіна користувача у верхній частині екрану, тоді коли у випадку неавторизованого користувача виникає повідомлення про те, що користувач є невідомим системі.

Рисунок 3.9 – Вигляд вікна авторизованого користувача

Якщо реєстрація користувача була невдалою (логін не є унікальним, обличчя не зчитано) користувачеві відкривається відповідне повідомлення.

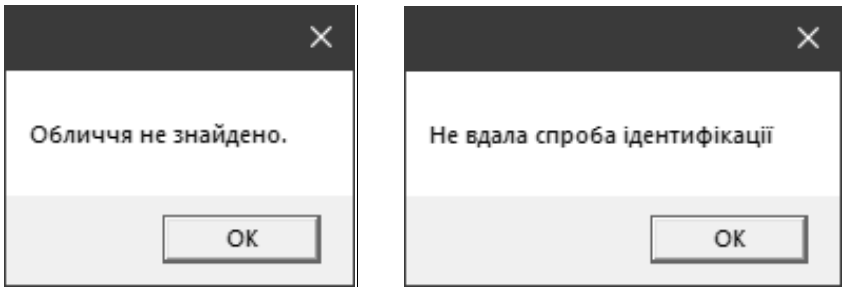


Рисунок 3.10 – Повідомлення про невдалі спроби входу/реєстрації користувача

У випадку, якщо дані користувача отримані системою при авторизації некоректні, невірні, підроблені, не відповідають цілком один одному, користувач отримує повідомлення про те, що користувача з такими параметрами не знайдено. Доступ до захищеної системи залишається закритим.

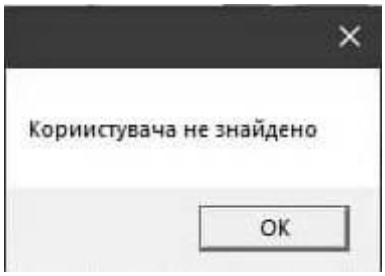


Рисунок 3.11 – Вигляд вікна з повідомленням про не авторизованого користувача

У випадку, якщо введені користувацькі унікальні дані (логін, зразок обличчя, відбиток пальця (як це передбачено повною версією програми) – повністю відповідають зразками збереженим у додатку та є достовірними, користувачеві відкривається відповідне повідомлення про успішну авторизацію (рис. 3.12) та надається доступ до захищеної системи.

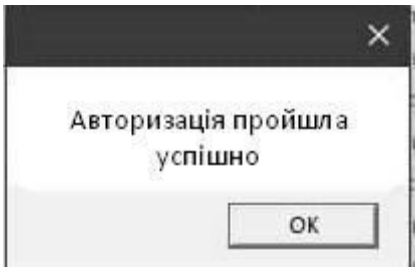
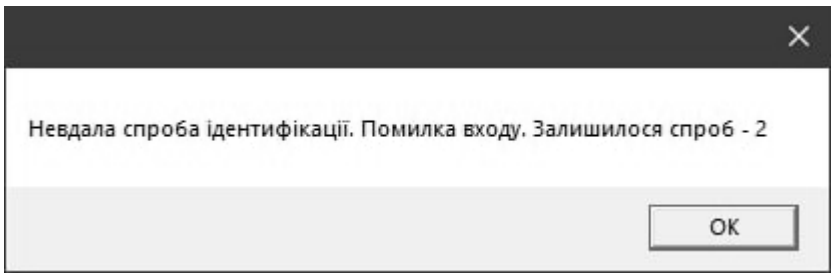
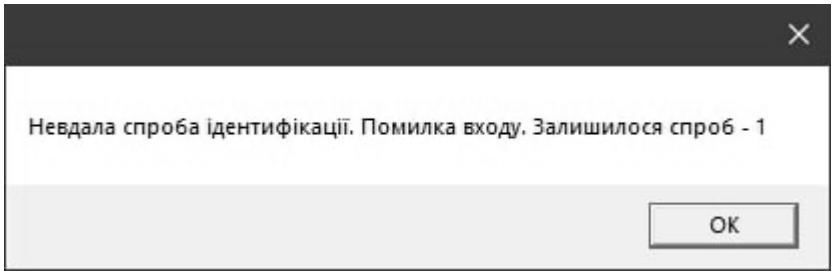


Рисунок 3.12 – Вигляд вікна про вдало авторизованого користувача

Також, як зазначалось при розробці алгоритму роботи програми, користувачеві надається лише три спроби авторизації. У випадку, якщо користувач тричі здійснює три невдалі спроби авторизації, відкривається повідомлення про блокування доступу до системи та автоматичне закриття усіх діалогових вікон системи. Попередньо, після кожної невдалої спроби авторизації користувача система інформує про кількість спроб входу та загрозу блокування доступу.





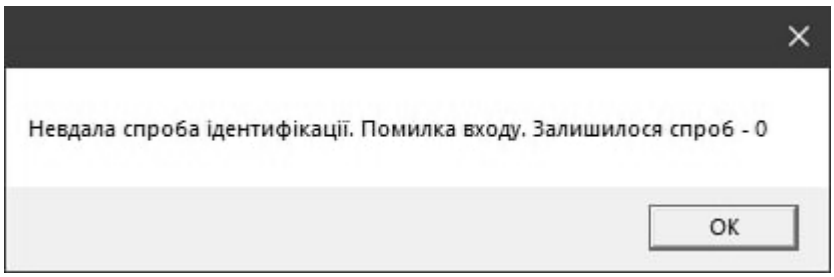


Рисунок 3.13 – Діалогові вікна з повідомленням про обмеження можливих спроб авторизації

Скориставшись кнопкою в системі «Про нас», користувач може побачити дані про розробника. Також в даному вікні може бути розміщена довідка щодо програми, контакти зворотного зв’язку та інші необхідні дані, що могли б передбачатись у випадку розробки власне самої системи.

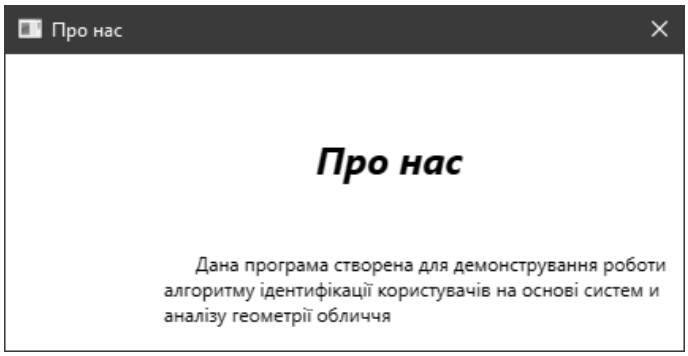


Рисунок 3.14 – Вигляд вікна кнопки «Про нас»

Права доступу до керування системою, реєстрації та видалення нових користувачів належать адміністратору системи, який може заходити в неї за допомогою своїх унікальних адміністраторських даних.

Вигляд вікна авторизованого адміністратора системи зображено на рисунку 3.15.

У верхній частині діалогово вікна відображається логін авторизованого адміністратора, а також активна кнопка «Адміністрування користувачів», де є можливість керувати обліковими записами можливих та існуючих користувачів системи.

Рисунок 3.15 – Вигляд вікна авторизованого адміністратора

Отже, в даному підрозділі було детально описано алгоритм роботи з розробленим додатком для типового користувача системи комплексної ідентифікації за логіном та образом обличчям. Даний користувацький інтерфейс розроблявся згідно поставлених вимог до візуальної та функціональної задачі додатку.

3.4 Представлення результатів обробки зображення на основі вдосконаленого методу

Залежність (2.1) визначає те, що фільтр залежить від частоти і напрямків квазіперіодичної структури зображення. Зазвичай, на практиці для спрощення завдання розраховується середня частота зображення. Вона є незмінною в кожній точці.

Для побудови поля напрямків може застосовуватися кілька способів, найбільш швидким з яких є диференціальний метод. Він дозволяє побудувати чотириградаційне поле напрямків.

Таким чином, маючи частоту і чотири напрямки, попередньо будуються 4 фільтра Габора (по одному на кожен із напрямків). Після чого в кожній точці зображення відбувається згортка фільтра із зображенням по певній галузі, що дає вихідне значення нового зображення.

Отже, на практиці для розпізнавання користувача за образом обличчя, застосуємо для обробки зображення фільтр Габора та медіанну фільтрацію (для визначення контрольної області придушення імпульсних шумів, відповідно).

На рис. 3.16 – 3.18, що є результатом зчитаного обличчя веб-камерою, представлено результати здійснення фільтрації зображення типовим та удосконаленим методом фільтрування.

Рисунок 3.16 – Зображення перед фільтрацією

Рисунок 3.17 – Зображення з фільтрацією на основі типового фільтра Габора

Рисунок 3.18 – Зображення, яке відфільтроване на основі комплесного методу фільтрування

Як можна побачити, з рисунку 3.16, оброблене зображення є більш чітким, з меншою кількістю шумів та неточностей (у порівнянні із рис. 3.15). Таке покращення якості зображення надає системі можливість ефективніше розпізнати надане зораження та надати достовірну інформації ідентифікації користувача.

Аналізуючи отриманий результат, можна вважати, що практична реалізація удосконаленої фільтрації зображення, що здійснена у даній роботі, має практичну доцільність та може застосовуватися для трасування та векторизації біометричних зображень [14].

3.5 Висновки до розділу

Таким чином, в даному розділі було здійснено практичну реалізацію програми комплексної ідентифікації до системи захисту з використанням ідентифікації за обличчям та логіном.

У розділі описано проектування графічного користувацького інтерфейсу, особливості програмної реалізації розробки, реалізацію користувацького інтерфейсу додатку та представлено результати роботи удосконаленого методу фільтрації зображення.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки ПЗ на основі біометричної ідентифікації

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності [70].

Результатом магістерської кваліфікаційної роботи є розроблення та реалізація програмного засобу для підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення.

Для проведення технологічного аудиту залучено трьох незалежних експертів. У нашому випадку такими експертами є викладачі кафедри МБІС: Карпінець В. В. (к.т.н., доцент каф. МБІС ВНТУ) та Дьогтєва І.О. (асис. каф. МБІС ВНТУ), Шиян А. А. (к.ф.-м.н., доцент каф. МБІС ВНТУ).

Оцінювання комерційного потенціалу було здійснене за критеріями, що наведені в таблиці 4.1

Таблиця 4.1 - Критерії оцінювання комерційного потенціалу розробки бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатацій ні витрати значно вищі, ніж в аналогів	Експлуатацій ні витрати дещо вищі, ніж в аналогів	Експлуатацій ні витрати на рівні експл. витрат аналогів	Експлуатацій ні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навч. наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження таблиці 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої к-ті дозвільних документів на вир-во та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки пові-домлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 4.2.

Таблиця 4.2 - Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1 – Карпінець В.В.	2 – Дьогтєва І.О.	3 – Шиян А.А
1	4	4	4
Ринкові переваги (недоліки):			
2	4	3	4
3	3	3	3
4	5	5	3
5	4	4	4
Ринкові перспективи			
6	4	4	4
7	3	4	3
Практична здійсненність			
8	4	4	4
9	4	4	3
10	4	4	4
11	4	4	4
12	3	4	4
Сума балів	$СБ_1 = 46$	$СБ_1 = 47$	$СБ_1 = 44$
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = 45{,}3$		

За даними таблиці 4.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Рівень комерційного потенціалу розробки, становить 46 балів, що відповідає рівню «високий».

Проаналізуємо суть технічної проблеми та розглянемо аналоги. Наукова новизна розробки полягає в удосконаленні методу фільтрування зображення, що, на відміну від існуючих підходів, дозволяє підвищити точність автентифікації користувачів засобами технології Face ID на основі фільтра Габора.

Одним із найважливіших етапів оброблення біометричної інформації є попереднє оброблення зображень. У багатьох випадках інформативнішими є геометричні характеристики межі об’єктів цих зображень – краї, які служать важливими ознаками для класифікації зображених об'єктів і сприйняття зображення в цілому. Фільтрація зображень є актуальним завданням, оскільки стосується реконструкції зображення шляхом зміни інтенсивності пікселів.

У магістерській роботі запропоновано застосування двох методів біометричної ідентифікації, яка порівняно з парольною (використання одноразових та багаторазових паролів) та майновою (використання апаратно-програмних систем ідентифікації і автентифікації або пристроїв введення ідентифікаційних ознак) має низку переваг, а саме: надійність і швидкість здійснення автентифікації; високий рівень безпеки даних автентифікації; дані використовуваних біометричних характеристик не можна втратити або забути;

пристрої для біометричної автентифікації зручні в користуванні та

експлуатації.

Враховуючи такі переваги розробленого методу, можемо порівняти його з основними аналогами.

У таблиці 4.4 наведені основні технічні показники аналога і нового програмного продукту.

Таблиця 4.4 – Основні технічні показники аналога і нового програмного продукту

Показники, %	Аналог	Нова розробка	Відношення параметрів нової розробки до параметрів аналога
Функціональність	95	100	1,05
Надійність	80	100	1,25
Сумісність	80	100	1,25
Супровід	80	100	1,25
Економія ресурсів і часу	90	100	1,11
Простота використання	70	100	1,43

В програмній розробці, порівняно з аналогами, де використовується лише логін та парольт (наприклад, додаток TronLink [61]), в розроблюваному додатку реалізовуватиметься застосування біометричної ідентифікації із розпізнаванням образу обличчя та, за потреби, відбитку пальця.

Враховуючи, сучасні темпи розвитку технологій, контроль доступу до системи лише з використанням логіну та паролю – є малонадійним, оскільки витік та підробка таких авторизаційних даних з часом стає все більш ймовірною. Проте, у поєднанні з використанням біометричного методу – захищеність входу в систему підвищується, оскільки біометричні дані є унікальними та їх неможливо підробити.

На підставі вищевикладеного можна стверджувати, що нове технічне рішення, що пропонується для розробки, буде мати кращі показники, ніж у аналога та більшою мірою задовольнить потреби споживачів. Тому його розробка та впровадження є актуальним та доцільним.

Програмний засіб на сьогодні має перспективу та користь як для пересічних користувачів так і для спецслужб. Продукт, який пропонується є реалізованим засобом, що дозволяє проводити автентифікацію користувачів в системі. Готовий програмний продукт буде реалізовуватись на ринку програмних засобів шляхом щомісячної підписки за певну плату.

Під час встановлення ціни та попиту на новий програмний продукт основна увага повинна акцентуватися на унікальності об'єкта купівлі-продажу, цінах продуктів конкурентів, перевагах порівняно з аналогами, витратах, які зазнає покупець у разі заміни старого продукту новим, ступені терміновості та гостроті потреби.

Програмний засіб готовий для використання. Фахівці відповідної кваліфікації наявні, трудові та фінансові ресурси теж, обслуговування програми може відбуватись в режимі он-лайн, з будь-якої точки світу, оскільки немає проблем з передачею на нього прав. Комерціалізація розробки знаходиться на початковому етапі. Ведуться пошуки інвесторів та партнерів. Наявні зацікавлені особи, що готові першими випробувати програмний засіб в обмін на акт впровадження та подальшу рекламу від їх імені. Просування на ринок планується шляхом реалізації та продажу через спеціалізовані магазини програмного забезпечення.

4.2 Прогнозування витрат на виконання наукової роботи та впровадження її результатів

Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи складається з таких етапів:

1-й етап: розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;

2-й етап: розрахунок загальних витрат на виконання даної роботи;

3-й етап: прогнозування загальних витрат на виконання та впровадження

результатів даної роботи.

Виконаємо розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи, за такими статтями та формулами, приймаючи до уваги те, що для розробки інформаційної технології було залучено одного розробника програмного забезпечення.

1. Основна заробітна $З_о$:

$$З_о = \frac{М}{Т_р} \cdot t, \text{грн.} \quad (4.1)$$

де М – місячний посадовий оклад – 25 000 грн.;

$Т_р$ – число робочих днів в місяці; приблизно $Т_р = 20$ днів;

t – число робочих днів роботи – 30 днів.

Таким чином:

$$З_о = \frac{25000}{20} \cdot 20 = 25\,000 \text{ (грн.)}$$

Таблиця 4.5 – Витрати по заробітній платі

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату
Розробник	25 000	1250	20	25 000
Всього				25 000

2. Додаткова заробітна плата $З_д$ працівників розраховується як 12% від основної заробітної плати:

$$З_д = 0{,}12 \cdot 25\,000 = 3\,000 \text{(грн.)} - \text{для розробника}$$

3. Нарахування на заробітну плату $Н_{зп}$ розробника становить:

$$Н_{зп} = (З_о + З_д) \cdot \frac{\beta}{100} \quad (4.2)$$

де $З_о$ – основна заробітна плата розробника;

$З_д$ – додаткова заробітна плата розробника;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування – 22%.

$$Н_{зп} = (25\,000 + 4\,500) \cdot 0{,}22 = 6\,160 \text{ (грн.)}$$

4. Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час виконання даного етапу роботи розраховуємо за формулою:

$$А = \frac{Ц \cdot Т}{12 \cdot Т_В} \quad (4.3)$$

де Ц – загальна балансова вартість обладнання, приміщення тощо, грн.;

Т – фактична тривалість використання, міс;

$Т_В$ – термін використання обладнання, приміщень тощо, роки.

Розробка програмного забезпечення ведеться 1 місяць.

Розрахунки зведено до таблиці 4.6:

Таблиця 4.6 – Амортизаційні відрахування

Найменування	Балансова вартість (грн.)	Термін використання (років)	Фактична тривалість використання, (міс.)	Величина амортизаційних відрахувань, (грн..)
Офісне приміщення	30 000	25	1	1000
Комп'ютер	15 000	3	1	420
Монітор	5 000	3	1	140
			Всього	1 560

5. Витрати на комплектуючі К, що були використані під час виконання даного етапу роботи, розраховуються за формулою:

$$К = \sum_{1}^{n} Н_i \cdot Ц_i \cdot К_i \text{ (грн.)} \quad (4.4)$$

де $Н_i$ – кількість комплектуючих *i*-го виду, шт.;

$Ц_i$– ціна комплектуючих *i*-го виду, грн.;

$К_i$ – коефіцієнт транспортних витрат, $К_i$ = (1,1…1,15);

n – кількість видів комплектуючих.

Таблиця 4.7 – Витрати на комплектуючі

Найменування комплектувальних	Кількість	Ціна за штуку, грн.	Сума, грн.	Примітка
Клавіатура (тип 1)	1	400 грн.	500 грн.	
Веб-камер (тип 2)	1	1000 грн.	1000 грн.	
Всього:			$К_i = 1,2$	1 800 грн.

6. Витрати на силову електроенергію $В_е$ розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п \text{ (грн.)} \quad (4.5)$$

Де В – вартість 1 кВт-год.;

П – установлена потужність обладнання – 0,8 кВт;

Ф – фактична кількість годин роботи обладнання – 250 годин,

$К_п$ – коефіцієнт використання потужності.

$$В_е = 3,5 \cdot 0,8 \cdot 200 \cdot 0,14 = 100 \text{ (грн).}$$

7. Інші витрати $В_{ін}$ охоплюють:

– витрати на управління організацією;

– оплату службових відряджень;

– витрати на утримання, ремонт та експлуатацію основних засобів;

– витрати на опалення, освітлення, водопостачання, охорону праці тощо.

Інші витрати $В_{ін}$ можна прийняти як 100% від суми основної заробітної

плати розробника:

$$В_{ін} = 25\,000 \cdot 1 = 25\,000 \text{ (грн)}$$

Послуги Інтернету – 350 грн., канцтовари – 400 грн., інше – 250грн. Загальна вартість становить:

$$350 + 400 + 250 = 1000 \text{ (грн.)}$$

8. Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи – В.

$$В = 25\,000 + 3\,000 + 6\,160 + 1\,560 + 1\,800 + 100 + 25\,000 + 1000 = 63\,620 \text{ (грн.)}$$

9. Проведемо прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи. Прогнозування здійснюється за формулою:

$$ЗВ = \frac{В_{заг}}{\beta}, \text{грн.} \quad (4.6)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Так, якщо розробка знаходиться:

– на стадії науково-дослідних робіт, то $\beta \approx 0{,}1$;

– на стадії технічного проектування, то $\beta \approx 0{,}2$;

– на стадії розробки конструкторської документації, то $\beta \approx 0{,}3$;

– на стадії розробки технологій, то $\beta \approx 0{,}4$;

– на стадії розробки дослідного зразка, то $\beta \approx 0{,}5$;

– на стадії розробки промислового зразка, $\beta \approx 0{,}7$;

– на стадії впровадження, то $\beta \approx 0{,}9$.

$В_{заг}$ – загальна вартість всієї наукової роботи.

$$В = 63\,620 \text{ (грн.)}$$

$$ЗВ = \frac{63\ 620}{0{,}7} = 90\ 885\ (\text{грн.})$$

Отже, прогноз загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи складає орієнтовно 91 тис. (грн.)

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

У даному підрозділі проведемо кількісне прогнозування, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної наукової роботи.

В умовах ринку узагальнюючим позитивним результатом, що його отримує підприємство від впровадження результатів тієї чи іншої розробки, є збільшення чистого прибутку підприємства. Зростання чистого прибутку можна оцінити у теперішній вартості грошей.

Зростання чистого прибутку забезпечить інвестору надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Виконання даної наукової роботи та впровадження її результатів складає приблизно 1 рік. Позитивні результати від впровадження розробки очікуються вже в перші місяці після впровадження.

Проведемо детальне прогнозування позитивних результатів та кількісне їх оцінювання по роках.

Обчислимо збільшення чистого прибутку підприємства $\Delta П_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta П_i = \sum_{1}^{n} (\Delta П_я \cdot N + П_я \cdot \Delta N)_i \qquad (4.7)$$

де $\Delta П_я$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$П_я$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

Припустимо, що внаслідок впровадження результатів наукової розробки чистий прибуток підприємства збільшиться на 100 грн., а кількість одиниць реалізованої послуги збільшиться:

– протягом першого року – на 600 од.,

– протягом другого року – ще на 850 од.,

– протягом третього року – ще на 1000 од.

Орієнтовно: реалізація продукції до впровадження результатів наукової розробки складала 1 шт., а прибуток, що його отримувало підприємство на одиницю продукції до впровадження результатів наукової розробки – 50 грн.

Потрібно спрогнозувати збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства $\Delta П_1$ протягом першого року складе:

$$\Delta П_1 = 50 \cdot 1 + (50 + 100) \cdot 600 = 90\,050 \text{ (грн.)}$$

Обчислимо збільшення чистого прибутку підприємства $\Delta П_2$ протягом другого року:

$$\Delta П_2 = 50 \cdot 1 + (50 + 100) \cdot (600 + 850) = 179\,794 \text{(грн.)}$$

Збільшення чистого прибутку підприємства $\Delta П_3$ протягом третього року становитиме:

$$\Delta П_3 = 50 \cdot 1 + (50 + 100) \cdot (600 + 850 + 1000) = 276\ 146\ (\text{грн.})$$

Отже, розрахунки показують, що відповідно прогнозуванню комерційний ефект від впровадження розробки виражається у значному збільшенні чистого прибутку підприємства.

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Основними показниками, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає:

1-й крок. Розрахунок теперішньої вартості інвестицій PV, що вкладаються в наукову розробку.

Такою вартістю ми можемо вважати прогнозовану величину загальних витрат ЗВ на виконання та впровадження результатів НДДКР, тобто $ЗВ = PV = 90\ 885\ (\text{грн.})$

2-й крок. Розрахуємо очікуване збільшення прибутку$\Delta П_i$, що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку також було розраховане нами раніше та становить:

$$\Delta П_1 = 90\ 050(\text{грн}), \Delta П_2 = 179\ 79(\text{грн}), \Delta П_3 = 276\ 146(\text{грн}).$$

3-й крок. Будуємо вісь часу, на якій відображаємо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Рисунок 4.1 характеризує рух платежів (інвестицій та додаткових прибутків).

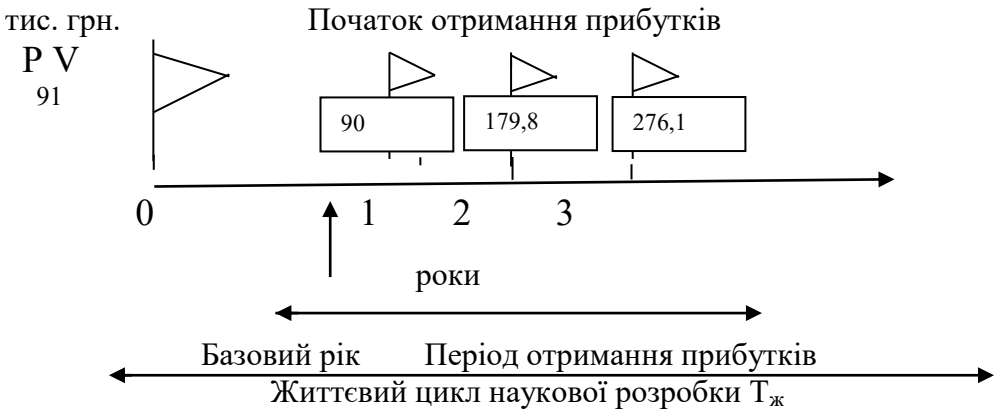


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

4-й крок. Розрахуємо абсолютну ефективність вкладених інвестицій $Е_{абс}$ за формулою:

$$Е_{абс} = (ПП - PV), (грн.) \quad (4.8)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій $PV = ЗВ$, грн.

Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_{1}^{Т} \frac{\Delta П_i}{(1+\tau)^t}, (грн) \quad (4.9)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн.;

Т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні – 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до

точки «0»;

$$ПП = \frac{90\,050}{(1+0{,}1)^1} + \frac{179\,794}{(1+0{,}1)^2} + \frac{276\,146}{(1+0{,}1)^3} = 537\,803\ (грн.)$$

$$Е_{абс} = 537\,803 - 90\,885 = 446\,918\ (грн.)$$

Оскільки $Е_{абс} > 0$, результат від проведення наукових досліджень щодо розробки програмного продукту та їх впровадження принесе прибуток, тобто є доцільним, проте це ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даної програми.

5-й крок. Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $Е_в$ за формулою:

$$Е_в = \sqrt[Т_ж]{1 + \frac{Е_{абс}}{PV} - 1} \qquad (4.10)$$

де $Е_{абс}$ – абсолютна ефективність вкладених інвестицій, грн.;

PV – теперішня вартість інвестицій $PV = ЗВ$, грн.

$Т_ж$ – життєвий цикл наукової розробки, роки.

$$E_в = \sqrt[3]{1 + \frac{446\,918}{90\,885}} - 1 = \sqrt[3]{6} - 1 = 0{,}81 \text{ або } 81\%$$

Порівняємо $E_в$ з мінімальною (бар'єрною) ставкою дисконтування τ_{min}, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть.

Спрогнозуємо величину τ_{min}.

У загальному вигляді мінімальна (бар'єрна) ставка дисконтування τ_{min}визначається за формулою:

$$\tau_{min} = d + f \tag{4.11}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; $d = 0{,}2$;

f – показник, що характеризує ризикованість вкладень; величина $f = 0{,}5$.

$$\tau_{min} = 0{,}2 + 0{,}5 = 0{,}7$$

Оскільки

$$E_{\text{в}} = 81\% > \tau_{min} = 70\%,$$

то у інвестора є потенційна зацікавленість у фінансуванні даної наукової розробки.

6-й крок. Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій $\text{Т}_{\text{ок}}$ за формулою:

$$\text{Т}_{\text{ок}} = \frac{1}{\text{Е}_{\text{в}}}, \text{рік} \tag{4.12}$$

$$\text{Т}_{\text{ок}} = \frac{1}{0{,}81} = 1{,}2 \text{ (року)}$$

Оскільки термін окупності вкладених у реалізацію наукового проекту інвестицій менше трьох років

$$(\text{Т}_{\text{ок}} < 3 \text{ років}),$$

то фінансування нової розробки є доцільним.

4.5 Висновки до розділу

В даному розділі було виконано оцінювання комерційного потенціалу розробки програмного засобу для підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення.

Проведено технологічний аудит з залученням трьох незалежних експертів. Визначено, що рівень комерційного потенціалу розробки вище середнього. Проведено порівняння з аналогом. Згідно з проведеним оцінюванням нова розробка є якісною та конкурентоспроможною.

Рівень комерційного потенціалу розробки, становить 45,3 балів, що відповідає рівню «високий».

Згідно із розрахунками всіх статей витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи загальні витрати на розробку складають 90 885 (грн.). Розрахована абсолютна ефективність вкладених інвестицій в сумі 446 918 (грн.) свідчить про отримання прибутку інвестором від комерціалізації програмного продукту.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 81%, що вище за мінімальну бар'єрну ставку дисконтування, яка складає 70%. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.

Термін окупності вкладених у реалізацію проекту інвестицій становить 1,2 (року), що також свідчить про доцільність фінансування нової розробки.

Отже, проаналізувавши отримані економічні показники, можна вважати, що запропонована розробка програмного засобу має високий комерційний потенціал, а тому є доцільною для подальшого впровадження.

ВИСНОВОК

У магістерській дипломній роботі розроблялася система підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення.

Біометрична ідентифікація – це пред'явлення користувачем свого унікального біометричного параметра і процес порівняння його з усією базою наявних даних. Біометричні системи контролю доступу зручні для користувачів тим, що носії інформації знаходяться завжди при них, не можуть бути загублені або вкрадені.

Було доведено актуальність обраної в роботі теми, розроблено алгоритм та практично реалізовано комплексну систему, що базується на використанні біометричного ідентифікатора у вигляді образу обличчя користувача.

У першому розділі роботи було здійснено аналіз теоретичних засад по темі дослідження: вивчено переваги та недоліки існуючих методів біометричної ідентифікації в інформаційних системах, досліджено особливості ідентифікації користувачів за образом обличчя, а також деякі види фільтрації зображень для розуміння ефективності їх використання у системах, що захищаються.

У другому розділі було розроблено алгоритм роботи комплексної біометричної автентифікації у системах безпеки, зокрема розглянуто та обрано методи для реалізації біометричної ідентифікації за образом обличчя.

Виходячи з теми роботи та поставлених завдань, для розроблення програми комплексної автентифікації, було обрано мову програмування C# та середовище Visual Studio для програмної реалізації додатку.

У третьому розділі роботи було здійснено практичну реалізацію програми комплексної автентифікації до системи захисту з використанням ідентифікації за образом обличчя.

Описано проектування графічного користувацького інтерфейсу, особливості програмної реалізації розробки, реалізацію користувацького інтерфейсу додатку та тестування ефективності та достовірності

удосконаленого методу. За результатом аналізу отриманого результату, можна вважати, що практична реалізація удосконаленої фільтрації зображення, що здійснена у даній роботі, має практичну доцільність та може застосовуватися для трасування та векторизації біометричних зображень

Для ефективного оброблення зображень автором роботи запропоновано використовувати фільтрацію на основі методу фільтрів Габора, який дозволяє створити зображення з більш чіткими контурами, та подальшою обробкою на основі медіанної фільтрації, що дозволяє покращити якість зображення шляхом придушення імпульсних шумів, які, зокрема, неминуче з'являються в цифрових камерах в умовах малого освітлення.

Таким чином, в результаті виконаної роботи, було досягнуто поставленої мети, а саме розроблено програму комплексної автентифікації для підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення.

ПЕРЕЛІК ПОСИЛАНЬ

1. Додонов О.Г., Нестеренко О.В., Бойченко А.В., Бойченко О.А. Формування, інтеграція та використання інформаційних ресурсів органів державної влади // Реєстрація, зберігання і оброб. даних. – 2002. – Т. 4, № 3. – С. 69 –75.

2. Торокин, А.А. Инженерно–техническая защита информации / А. А. Торокин. – Москва: Гелиос АРВ, 2005. – 960 с.

3. Д.В. Ландэ, В.Н. Фурашев. О цифровой идентификации личности [Електронний ресурс] // Режим доступу: http://dwl.kiev.ua/art/iden/iden-art.pdf

4. Аронов А.В. Основы биометрии [Електронний ресурс] // Режим доступу: http://habrahabr.ru/

5. Татарченко Н.В, Тимошенко С.В. Биометрическая идентификация в системах безопасности [Електронний ресурс] // Режим доступу: http://www.vidim.od.ua/

6. Яремчук Ю. Є., Павловський П. В., Катаєв В. С., Сінюгін В. В.. Комплексні системи захисту інформації : навчальний посібник / Вінниця: ВНТУ, 2017. – 120 с.

7. І. І. Данилюк, В. В. Карпінець, А. В. Приймак,Ю. Є. Яремчук, О. І. КостюченкоМетод ідентифікації користувача за клавіатурним почерком на основі нейромереж / Вінниця: ВНТУ, 2017. – 120 с.

8. Азарова А. О., Гудзь В. О., Блонський В. О. Управління інформаційною безпекою в державних установах на основі біометричної аутентифікації відбитків пальців для захисту інформації від несанкціонованого доступу. [Електронний ресурс] // Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fm/all-fm2019/paper/view/7429

9. Хорошко В.А. Методы и средства защиты информации / В.А. Хорошко, А.А. Чекатков. – К.: Изд-во Юниор, 2003. – 504 с.

10 . Гонсалес Р. Цифровая обработка зображений / Вудс Р. – Техносфера, Москва, 2005. – 1072 с.

11. Богачук В.В. Ідентифікація соціальних груп деструктивного впливу соціальних мереж/ Тези доповіді/ Матеріали XLIX Науково- технічній конференції підрозділів Вінницького національного технічного університету 2020 [Електронний ресурс] // Режим доступу: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/28986/9795.pdf?sequence=3&isAllowed=y.

12. Богачук В.В. Корпоративна культура в системі управління персоналом/ Тези доповіді/ Матеріали XLIX Науково-технічній конференції підрозділів Вінницького національного технічного університету 2020 [Електронний ресурс] // Режим доступу: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/29059/9179.pdf?sequence=3&isAllowed=y

13. Богачук В.В. Економічна ефективність виробництва продукції і шляхи її підвищення/ Тези доповіді/ Матеріали XLIX Науково-технічній конференції підрозділів Вінницького національного технічного університету 2020 [Електронний ресурс] // Режим доступу: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/29041/9015.pdf?sequence=3&isAllowed=y

14. Богачук В.В. Розробка модуля захисту програмного забезечення від несанкціонованого копіювання/ Тези доповіді/ Матеріали XLVII Науково-технічній конференції підрозділів Вінницького національного технічного університету 2018 [Електронний ресурс] // Режим доступу: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/21119/5134.pdf?sequence=3&isAllowed=y

15. Азарова А. О, Богачук В. В., Безмощук О. В. Автентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення/ Тези доповіді/ Всеукраїнська науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2022) [Електронний ресурс] //

Режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2022/paper/viewFile/14176/11988

16. Азарова А. О, Безмощук О. В., Богачук В. В. Ідентифікація користувачів на основі удосконаленого методу дактилоскопії/ Тези доповіді/ Всеукраїнська науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи (МН-2022) [Електронний ресурс] // Режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2022/paper/viewFile/14177/11992

17. Компьютерный журнал "КомпьюПресс", [Електронний ресурс] // Режим доступу: http://www.compress.ru

18. Попов М., Задорожный В. Биометрические системы безопасности [Електронний ресурс] // Режим доступу: www.BRE.ru

19. Сафронов В.В. Современные биометрические методы идентификации. [Електронний ресурс] // Режим доступу: http://habrahabr.ru/blogs/

20. Борзенко А. Биометрические системы распознавания внешности: [Електронний ресурс] // Режим доступу: http://www.bytemag.ru/

21. Face Verification using Correlation Filters Marios Savvides, Electrical and Computer Eng. Dept, Carnegie Mellon University Pittsburgh, PA 15213, U.S.A. [Електронний ресурс] // Режим доступу: http://www.ece.cmu.edu

22. On the Recent Use of Local Binary Patterns for Face Authentication S [Електронний ресурс] // Режим доступу: http://www.idiap.ch

23. Болл Р.М. Руководство по биометрии. – М.:Наука: 2011. – 460 с.

24. Визильтер Ю.В., Желтов С.Ю., Князь В.А. и др. Обработка и анализ цифровых изображений. – М.: ДМК Пресс, 2007. – 464 с.

25. Методы компьютерной обработки изображений / Под ред. В.А. Сойфера. – М.: Физматлит, 2001. – 784 с.

26. Девід Ліон. Товариство спостереження: Моніторинг повсякденного життя / Девід Ліон. – Філадельфія, 2001.

27. Хагхигат М. Дискриминантний кореляційний аналіз: Fusion в режимі реального часу для мультимодального біометричного розпізнавання / М.

Хагхигат, М. Абдель-Мотталеб, В. Аналлабі., 2016.

28. N. K. Ratha. Enhancing security and privacy in biometrics-based authentication systems, IBM systems Journal / N. K. Ratha, J. H. Connell, R. M. Bolle. – №40.

29. Шаров В. Біометричні методи комп'ютерної безпеки/ Владислав Шаров // ByteMag. – 2005 [Електронний ресурс] // Режим доступу: https://www.bytemag.ru/

30. Попов М. Біометричні системи безпеки / Попов М.., 2011.

31. Клімакін С.П. Ера біометрії / Клімакін С.П, Петруненко А.А., Черномордик О.М.., 2006.

32. Біометрія як універсальний спосіб ідентифікації людини [Електронний ресурс] // Режим доступу: http://bablyukh.clan.su/publ/1-1-0-4.

33. Шаньгин В. Ф. Информационная безопасность компьютерных систем и сетей : учеб. пособие. – М. : ИД «ФОРУМ» : ИНФРА-М, 2011 – 416 с.

34. Іванов В.Г., Мазниченко Н.І. Ідентифікація користувачів інформаційних систем: аналіз і прогнозування підходів.

35. Системний аналіз. Інформатика. Управління (САІУ-2012) : матеріали III Міжнар. наук.-практ. конф., м. Запоріжжя, 14-16 березня 2012 р. Запоріжжя : КПУ, 2012. С.127-128.

36. Біометричні системи безпеки [Електронний ресурс] // Режим доступу: http://uadoc.zavantag.com

37. Біометрична автентифікація [Електронний ресурс] // Режим доступу: http://ukrdoc.com.ua/

38. Draper B.A., Baek K., Bartlett M.S., Beveridge J.R. Recognizing faces with PCA and ICA. Comput. Vis. Image Underst. 2003; 91:115–137.

39. Ekman P., Friesen W.V. The repertoire of nonverbal behavior: Categories, origins, usage, and coding. Semiotica. 2010; 1:49 – 98.

40. Петренко С.А. Симонов С.В. Економічно виправдана безпека. Управління інформаційними ризиками. - М.: ДМК, 2003.

41. Вилле Й. Новые пути биометрии. Журнал сетевых решений LAN.

2005. № 1. С. 15–18

42. Есин В. И. Безопасность информационных систем и технологий / В. И. Есин, А. А. Кузнецов, Л. С. Сорока. – Х. : ООО «ЭДЭНА», 2010. – 656 с.

43. Мельников В.В.Захист інформації в комп'ютерних системах. – М.: Фінанси і статистика, 1997.

44. T. F. Chan The Digital TV Filter and Nonlinear Denoising / T. F. Chan, S, Osher, and J. Shen // IEEE Trans, Image Proc. – 2001. – P. 231–241.

45. A. Efros Texture synthesis by non parametric sampling / A. Efros and T. Leung // Proc. Int. Conf. Computer Vision (ICCV 99), Vol. 2, 1991 – P. 1033-1038.

46. M. Mahmoudi Fast Image and Video Denoising via Nonlocal Means of Similar Neighborhoods / M. Mahmoudi and G. Sapiro// IEEE Signal Processing Letters 12. – 2005. – P. 79.

47. Using Geometry and iterated renement for inverse problems: Total variation based image restoration / S. Osher, M. Burger, D. Goldfarb, J. Xu and W. Yin,], CAM-Report 04-13 UCLA. – 2004. – P.47

48. F. Malgouyres A noise selection approach of image restoration, Applications in signal and image processing IX / F. Malgouyres // Vol 4478. – 2001. – P. 34-41.

49. M. Lindenbaum On Gabor Contribution To Image Enhancement / M. Lindenbaum, M. Fischer and A. M. Bruchkstein // Pattern Recognization 27. – 1994. – P. 1-8.

50. L.P. Yaroslavsky Digital Picture Processing. An Introduction / L.P. Yaroslavsky // Berlin-Heidelberg-New York, Springer-Verlag. – 1985. – P. 276.

51. S.M. Smith Susan - a new approach to low level image processing / S.M. Smith and J.M. Brady //International Journal of Computer Vision Vol 23(1). – 1997. – P. 45-78.

52. C. Tomasi Bilateral Filtering for Gray and Color Images / C. Tomasi and R. Manduchi // in Proc. 6th Int. Conf. Computer Vision, New Delhi, India. – 1998.

53. P. Perona Scale space and edge detection using anisotropic diffusion / P. Perona and J. Malik // IEEE Trans. Patt. Anal. Mach. Intell. – 1990. – P. 629-639.

54. L.I. Rudin Nonlinear total variation based noise removal algorithms / L.I. Rudin and E. Fatemi S. Osher // Physica D. – 1992. – P. 259-268.

55. У. К. Прэтт Цифровая обработка изображений / У. К. Прэтт // М.: Мир. – 1982. – Р. 523.

56. D. Donoho Ideal spatial adaptation via wavelet shrinkage / D. Donoho, I. Johnstone // Biometrika, vol. 81. – 1994. – P. 425–455.

57. D. Donoho Denoising by soft-thresholding / D. Donoho // IEEE Transactions on Information Theory, 41. – 1995. – P. 613–627.

58. Y. Wang A Total Variation Wavelet Algorithm for Medical Image Denoising / Y. Wang and H. M. Zhou // The International Journal on Biomedical Imaging, Volume 2006, article ID 89095. – 2006. – P. 6.

59. A. Buades Non-Local Algorithm for Image Denoising / A. Buades, B. Coll, and J.-M. Morel // In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2. – 2005. – P. 605.

60. M. Black Robust Anisotropic Diffusion / M. Black, G. Sapiro, D. Marimont, and D. Heeger // IEEE Trans. Image Process., vol. 7, no. 3. – 1998.

61. Lin, Z., Kahrilas, P. J., Xiao, Y., Nicodème, F., Gonsalves, N., Hirano, I., & Pandolfino, J. E. (2013) 97-107.

62. Tomasi, C., & Manduchi, R. (1998, January). Bilateral filtering for gray and color images. In null (p. 839). IEEE.

63. Xu, Y., Weaver, J. B., Healy, D. M., & Lu, J. (1994). Wavelet transform domain filters: a spatially selective noise filtration technique. IEEE transactions on image processing, 3(6), 747-758.

64. Gritsik, V. V., and Nazarkevich, M. A. (2007). Matematichni modelI algoritmIv I realIzatsIya AtebfunktsIy. Dopovidi NatsIonalnoYi akademIyi nauk Ukrayini, (12), 37-42.

65. Lalitha, Y. S., & Latte, M. V. (2011). A novel approach noise filtration for MRI image sample in medical image processing. International Journal of Computer Science and Communication, 2(2), 359-363.

65. Yamanaka, A., Maeda, Y., & Sasaki, K. (2019). Ensemble Kalman filter-

based data assimilation for threedimensional multi-phase-field model: Estimation of anisotropic grain boundary properties. Materials & Design, 165, 107577.

66. Schilling, A., Knittel, G., & Strasser, W. (1996). Texram: A smart memory for texturing. IEEE Computer Graphics and Applications, 16(3), 32-41.

67. Caselles, V., Kimmel, R., & Sapiro, G. (1995, June). Geodesic active contours. In Proceedings of IEEE international conference on computer vision (pp. 694-699). IEEE.

68. Liu, Y. X., Yang, C. N., Wu, C. M., Sun, Q. D., & Bi, W. (2019). Threshold changeable secret image sharing scheme based on interpolation polynomial. Multimedia Tools and Applications, 1-15.

69. Sherlock, B. G., Monro, D. M., & Millard, K. (1994). Fingerprint enhancement by directional Fourier filtering. IEE Proceedings-Vision, Image and Signal Processing, 141(2), 87-94.

70. Andrysiak, T., & Choraś, M. (2005). Image retrieval based on hierarchical Gabor filters. International Journal of Applied Mathematics and Computer Science, 15, 471-480.

71. Назаркевич М.А., Возний Я.В. Узагальненн фільтрів Габора на основі Ateb-функцій, №2 (6), 2019

72. OpenCV шаг за шагом. Введение. [Електронний ресурс] // Режим доступу: http://robocraft.ru/blog/

73. Начало работы с Visual Studio [Електронний ресурс] // Режим доступу: https://visualstudio.microsoft.com

74. Visual Studio [Електронний ресурс] // Режим доступу: https://docs.microsoft.com/ru-2019

75. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.

ДОДАТКИ

Додаток А. Технічне завдання

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ
Голова секції «Управління інформаційною
безпекою» кафедри МБІС
д.т.н., професор
_____ Ю. Є. Яремчук

«24» вересня 2021 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:

Підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення
08-42.МКР.003.00.00 ТЗ

Керівник магістерської кваліфікаційної роботи
Проф. каф. МБІС Азарова А.О.
_____

Вінниця – 2021 р.

1 Найменування та область застосування

Програмний засіб для для підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення.

Область застосування: ідентифікація користувачів засобами технології Face ID.

2 Підстави для розробки

Розробка виконується на основі наказу ректора ВНТУ 24 вересня 2021 року №277.

3 Мета та призначення розробки

3.1 Мета розробки

Метою роботи є розроблення та реалізація програмного засобу для підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення.

3.2 Призначення

Розроблений програмний продукт забезпечує можливість підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення.

4 Джерела розробки

1. Хорошко В.А. Методы и средства защиты информации / В.А. Хорошко, А.А. Чекатков. – К.: Изд-во Юниор, 2003. – 504 с.

2. Задонский А. Ю. Вопросы аутентификации в информационных системах. – [Електронний ресурс] // Режим доступу: http://www.compserv.ru

3. Чала Л. Е. Методи динамічної ідентифікації користувачів розподілених інформаційних систем. / Л. Е. Чала. – Харків, 2006. – 20 с.

4. Дудатьєв А.В., Каплун В.А., Семеренко В.П. Д 81 Захист програмного забезпечення. Частина 1. Навчальний посібник. – Вінниця: ВНТУ, 2005. – 140 с.

5. Рудаков, О. М. Методы биометрической аутентификации [Електронний ресурс] // Режим доступу:: https://moluch.ru/archive/115/30980/

5 Вимоги до програми

5.1 Вимоги до функціональних характеристик

5.1.1 Програмний додаток призначений для ідентифікації користувачів засобами технології Face ID.

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків

5.1.3 Програмний додаток повинен мати зручний, легкий у розумінні користувача інтерфейс.

5.2 Вимоги до надійності:

5.2.1 Програмний додаток повинен бути працездатним продуктом, функціонувати без помилок.

5.2.2 Програмний додаток повинен працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень.

5.4 Вимоги до складу і параметрів технічних засобів:

– оперативна пам'ять – не менше 512 Mb.

5.5 Вимоги до інформаційної та програмної сумісності – будь-яка операційна система.

6 Вимоги до програмної документації

6.1 Обов’язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.3

7 Вимоги до технічного захисту інформації

6.1 Необхідно забезпечити контроль доступу користувачів до системи засобами технології Face ID.

8 Техніко-економічні показники

7.1 Програмний додаток має бути простим у використанні, легко змінюваним, мати можливість швидкого введення змін.

7.2 Витрати на програмні продукти, що використовуються в ході розробки мають бути мінімальними.

9 Стадії та етапи розробки

№	Назва етапів МКР	Початок	Закінчення
1	Визначення напрямку магістерської роботи, формулювання та затвердження теми	01.09.2021	26.09.2021
2	Аналіз предметної області обраної теми	27.09.2021	05.10.2021
3	Апробація отриманих результатів	06.10.2021	15.10.2021
4	Розробка алгоритму роботи	16.10.2021	31.10.2021
5	Написання магістерської роботи на основі розробленої теми	01.11.2021	14.11.2021
6	Розробка економічної частини	15.11.2020	21.11.2021
7	Передзахист магістерської кваліфікаційної роботи	22.11.2021	25.11.2021
8	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	26.11.2021	19.12.2021
9	Захист магістерської кваліфікаційної роботи	20.12.2021	23.12.2021

10 Порядок контролю та прийому

10.1 До приймання магістерської кваліфікаційної роботи надається:

– ПЗ до магістерської кваліфікаційної роботи;

– демонстрація результату магістерської кваліфікаційної роботи;

– презентація;

– відзив керівника роботи;

– відзив рецензента.

Технічне завдання до виконання прийняв _____ Богачук В.В.

Додаток Б. Лістинг CommonPage

```
Ausing System.Windows;
using System.Windows.Controls;

namespace FaceFinger
{
   public partial class CommonPage : Window
   {
      public CommonPage()
      {
         InitializeComponent();
      }
      private void AboutButton_Click(object sender, RoutedEventArgs e)
      {
         WFAbout wfAbout = new WFAbout();
         wfAbout.ShowDialog();
      }
      private void Registration_Click(object sender, RoutedEventArgs e)
      {
         var w = Application.Current.Windows[0];
         w.Hide();
         Registration wfAbout = new Registration();
         wfAbout.ShowDialog();
         w.Close();
      }
      private void Login_Click(object sender, RoutedEventArgs e)
      {
         var w = Application.Current.Windows[0];
         w.Hide();

         Login wfAbout = new Login();
         wfAbout.ShowDialog();

         w.Close();
      }

      private void Exit_Click(object sender, RoutedEventArgs e)
      {
         Registration.Visibility = Login.Visibility = Visibility.Visible;
         Exit.Visibility = Visibility.Hidden;
         AuthUserName.Content = "Не авторизовано";
      }
      private void RichTextBox_TextChanged(object sender, TextChangedEventArgs e)
      {
      }
   }
}
```

Додаток В. Лістинг Registration

```
using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Face;
using Emgu.CV.Structure;
using Emgu.CV.Util;
using FaceFinger.Context;
using FaceFinger.Models;
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.IO;
using System.Runtime.CompilerServices;
using System.Timers;
using System.Windows;
using System.Windows.Media.Imaging;
using System.Windows.Threading;

namespace FaceFinger
{
   public partial class Registration : Window, INotifyPropertyChanged
   {
      public Registration()
      {
         InitializeComponent();
         captureTimer1 = new Timer()
         {
            Interval = Config.TimerResponseValue
         };
         captureTimer1.Elapsed += captureTimer1_Elapsed;
      }
      private RecognizeContext _context = new RecognizeContext();
      #region Properties
      public event PropertyChangedEventHandler PropertyChanged;
      private VideoCapture videoCapture;
      private CascadeClassifier haarCascade;
      private Image<Bgr, Byte> bgrFrame = null;
      public string FaceName
      {
         get { return faceName; }
         set
         {
            faceName = value.ToUpper();
            lblFaceName.Dispatcher.Invoke(DispatcherPriority.Normal,
               new Action(() => { lblFaceName.Content = faceName; }));
```

```
            NotifyPropertyChanged();
        }
    }
    #endregion
    #region CameraCaptureImage
    private Bitmap cameraCapture;
    public Bitmap CameraCapture
    {
        get { return cameraCapture; }
        set
        {
            cameraCapture = value;
            imgCamera.Dispatcher.Invoke(DispatcherPriority.Normal,
             new Action(() => { imgCamera.Source = BitmapToImageSource(cameraCapture); }));
            NotifyPropertyChanged();
        }
    }
    #endregion
    #region CameraCaptureFaceImage
    private Bitmap cameraCaptureFace;
    public Bitmap CameraCaptureFace
    {
        get { return cameraCaptureFace; }
        set
        {
            cameraCaptureFace = value;
            NotifyPropertyChanged();
        }
    }
    #endregion
    #endregion
    public void OnLoad()
    {
        GetFacesList();
        if (videoCapture == null)
        {
            videoCapture = new VideoCapture(Config.ActiveCameraIndex);
            videoCapture.SetCaptureProperty(CapProp.Fps, 30);
            videoCapture.SetCaptureProperty(CapProp.FrameHeight, 500);
            videoCapture.SetCaptureProperty(CapProp.FrameWidth, 400);
            captureTimer1.Start();
        }
    }
    protected virtual void NotifyPropertyChanged([CallerMemberName] String
propertyName = "")
    {
        var handler = PropertyChanged;
        if (handler != null) handler(this, new PropertyChangedEventArgs(propertyName));
    }
```

```
        private void RegistrFace(string personName)
        {
            if (detectedFace == null)
            {
                MessageBox.Show("Обличчя не знайдено.");
                return;
            }
            detectedFace = detectedFace.Resize(100, 100, Inter.Cubic);
            detectedFace.Save(Config.FacePhotosPath + "face" + (faceList.Count + 1) +
Config.ImageFileExtension);
            StreamWriter writer = new StreamWriter(Config.FaceListTextFile, true);
            MessageBox.Show("Зразок обличчя зареєстровано коректно.");
        }
        #region Method
        public void GetFacesList()
        {
            if (!File.Exists(Config.HaarCascadePath))
            {
                string text = "Cannot find Haar cascade data file:\n\n";
                text += Config.HaarCascadePath;
                MessageBoxResult result = MessageBox.Show(text, "Error",
                    MessageBoxButton.OK, MessageBoxImage.Error);
            }
            haarCascade = new CascadeClassifier(Config.HaarCascadePath);
            faceList.Clear();
            string line;
            FaceData faceInstance = null;
            if (!Directory.Exists(Config.FacePhotosPath))
            {
                Directory.CreateDirectory(Config.FacePhotosPath);
            }
            if (!File.Exists(Config.FaceListTextFile))
            {
                string text = "Cannot find face data file:\n\n";
                    MessageBoxButton.OK, MessageBoxImage.Warning);
                switch (result)
                {
                    case MessageBoxResult.OK:
                        String dirName = Path.GetDirectoryName(Config.FaceListTextFile);
                        Directory.CreateDirectory(dirName);
                        File.Create(Config.FaceListTextFile).Close();
                        break;
                }
            }
            StreamReader reader = new StreamReader(Config.FaceListTextFile);
            int i = 0;
            while ((line = reader.ReadLine()) != null)
            {
                string[] lineParts = line.Split(':');
```

```
                faceInstance = new FaceData();
                faceInstance.FaceImage =
                  new Image<Gray, byte>(Config.FacePhotosPath + lineParts[0] +
Config.ImageFileExtension);
                faceInstance.PersonName = lineParts[1];
                faceList.Add(faceInstance);
              }
              foreach (var face in faceList)
              {
                imageList.Push(face.FaceImage.Mat);
                nameList.Add(face.PersonName);
                labelList.Push(new[] { i++ });
              }
              reader.Close();
              if (imageList.Size > 0)
              {
                recognizer = new EigenFaceRecognizer(imageList.Size);
                recognizer.Train(imageList, labelList);
              }
            }
            private void ProcessFrame()
            {
              if (active)
              {
                bgrFrame = videoCapture.QueryFrame().ToImage<Bgr, Byte>();

                if (bgrFrame != null)
                {
                  try
                  {
                    FaceName = "No face detected";
                    foreach (var face in faces)
                    {
                      bgrFrame.Draw(face, new Bgr(255, 255, 0), 2);
                      detectedFace = bgrFrame.Copy(face).Convert<Gray, byte>();
                      FaceRecognition();
                      break;
                    }
                    CameraCapture = bgrFrame.ToBitmap();
                  }
                  catch (Exception ex)
                  {
                  }
                }
              }
            }
            private void FaceRecognition()
            {
              if (imageList.Size != 0)
```

```
                {
                    FaceRecognizer.PredictionResult result = recognizer.Predict(detectedFace.Resize(100, 100, Inter.Cubic));
                    FaceName = nameList[result.Label];
                    CameraCaptureFace = detectedFace.ToBitmap();
                }
                else
                {
                    FaceName = "Please Add Face";
                }
            }
            private BitmapImage BitmapToImageSource(Bitmap bitmap)
            {
                using (MemoryStream memory = new MemoryStream())
                {
                    bitmap.Save(memory, System.Drawing.Imaging.ImageFormat.Bmp);
                    return bitmapimage;
                }
            }
            #endregion
            private void Button_Click_1(object sender, RoutedEventArgs e)
            {
                try
                {
                    if (!VerifyData())
                    {
                        MessageBox.Show("Не вдала спроба реєстрації");
                        return;
                    }
                    RegistrFace(UserName.Text);
                    var user = new User()
                    {
                        UserName = UserName.Text
                    };
                    var users = GetUsers();
                    if (users == null)
                    {
                        users = new List<User>() { user };
                    }
                    else
                    {
                        users.Add(user);
                    }
                    string json = JsonConvert.SerializeObject(users);
                    File.WriteAllText($"{Path.GetDirectoryName(System.AppDomain.CurrentDomain.BaseDirectory)}/ResourceFile.json", json);
                    MessageBox.Show("Успішно зареєстровано нового користувача");
                    commonpage.ShowDialog();
```

```
            enableAutoNavigate = false;
            w.Close();
        }
        catch (Exception exception)
        {
            Console.WriteLine(exception);
            MessageBox.Show("Не вдала спроба реєстрації");
        }
    }
    private List<User> GetUsers()
    {
        List<User> users = new List<User>();
        {
            string json = r.ReadToEnd();
            users = JsonConvert.DeserializeObject<List<User>>(json);
        }
        return users;
    }

    private bool VerifyData()
    {
        if (String.IsNullOrEmpty(UserName.Text))
        {
            MessageBox.Show("Введіть логін");
            return false;
        }
        if (imgCamera.Source == null)
        {
            MessageBox.Show("Сталася помилка. Зразук обличчя не вдалося отримати");
            return false;
        }
        return true;
    }
    private void Registration_OnClosed(object sender, EventArgs e)
    {
        active = false;
        if (enableAutoNavigate)
        {
            captureTimer1 = null;
            var w = Application.Current.Windows[0];
            w.Hide();
            CommonPage wfAbout = new CommonPage();
            wfAbout.ShowDialog();

            w.Close();
        }
    }
  }
}
```

Додаток Г. Лістинг Login

```
using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Face;
using Emgu.CV.Structure;
using Emgu.CV.Util;
using FaceFinger.Context;
using FaceFinger.Models;
using Microsoft.Win32;
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Timers;
using System.Windows;
using System.Windows.Media.Imaging;
using System.Windows.Threading;


namespace FaceFinger
{
   public partial class Login : Window, INotifyPropertyChanged
   {
      public Login()
      {
         InitializeComponent();
         captureTimer = new Timer()
         {
            Interval = Config.TimerResponseValue

         };
         captureTimer.Elapsed += CaptureTimer_Elapsed;
      }
      private RecognizeContext _context = new RecognizeContext();
      #region Properties
      public event PropertyChangedEventHandler PropertyChanged;
      private VideoCapture videoCapture;
      private CascadeClassifier haarCascade;
      private Image<Bgr, Byte> bgrFrame = null;
      private Image<Gray, Byte> detectedFace = null;
      private List<FaceData> faceList = new List<FaceData>();
      private VectorOfMat imageList = new VectorOfMat();
      private List<string> nameList = new List<string>();
```

```
        private VectorOfInt labelList = new VectorOfInt();
        private bool active = true;
        private bool enableAutoNavigate = true;
        private EigenFaceRecognizer recognizer;
        private Timer captureTimer;
        #region FaceName
        private string faceName;
        public string FaceName
        {
          get { return faceName; }
          set
          {
            faceName = value.ToUpper();
            lblFaceName.Dispatcher.Invoke(DispatcherPriority.Normal, new Action(() => {
lblFaceName.Content = faceName; }));
            NotifyPropertyChanged();
          }
        }
        #endregion
        #region CameraCaptureImage
        private Bitmap cameraCapture;
        public Bitmap CameraCapture
        {
          get { return cameraCapture; }
          set
          {
            cameraCapture = value;
            imgCamera.Dispatcher.Invoke(DispatcherPriority.Normal,
           new Action(() => { imgCamera.Source = BitmapToImageSource(cameraCapture); }));
            NotifyPropertyChanged();
          }
        }
        #endregion
        #region CameraCaptureFaceImage
        private Bitmap cameraCaptureFace;
        public Bitmap CameraCaptureFace
        {
          get { return cameraCaptureFace; }
          set
          {
            cameraCaptureFace = value;
          }
        }
        #endregion
        #endregion
        #region Constructor
        #endregion
        #region Event
```

```
        public void OnLoad()
        {
            GetFacesList();
            if (videoCapture == null)
            {
                videoCapture = new VideoCapture(Config.ActiveCameraIndex);
                videoCapture.SetCaptureProperty(CapProp.Fps, 30);
                videoCapture.SetCaptureProperty(CapProp.FrameHeight, 500);
                videoCapture.SetCaptureProperty(CapProp.FrameWidth, 400);
                captureTimer.Start();
            }
        }
        protected virtual void NotifyPropertyChanged([CallerMemberName] String
propertyName = "")
        {
            var handler = PropertyChanged;
            if (handler != null) handler(this, new PropertyChangedEventArgs(propertyName));
        }
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            OnLoad();
        }
        private void CaptureTimer_Elapsed(object sender, ElapsedEventArgs e)
        {
            ProcessFrame();
        }
        private void AboutButton_Click(object sender, RoutedEventArgs e)
        {
            WFAbout wfAbout = new WFAbout();
            wfAbout.ShowDialog();
        }
        private void NewFaceButton_Click(object sender, RoutedEventArgs e)
        {
            if (detectedFace == null)
            {
                MessageBox.Show("No face detected.");
                return;
            }
            detectedFace = detectedFace.Resize(100, 100, Inter.Cubic);
            StreamWriter writer = new StreamWriter(Config.FaceListTextFile, true);
            writer.Close();
            GetFacesList();
            MessageBox.Show("Successful.");
        }
        private void OpenVideoFile_Click(object sender, RoutedEventArgs e)
        {
            OpenFileDialog openDialog = new OpenFileDialog();
            if (openDialog.ShowDialog().Value == true)
            {
```

```
            captureTimer.Stop();
            videoCapture.Dispose();

            videoCapture = new VideoCapture(openDialog.FileName);
            captureTimer.Start();
            this.Title = openDialog.FileName;
            return;
        }
    }
#endregion
#region Method
public void GetFacesList()
{
    if (!File.Exists(Config.HaarCascadePath))
    {
        string text = "Cannot find Haar cascade data file:\n\n";
        text += Config.HaarCascadePath;
        MessageBoxResult result = MessageBox.Show(text, "Error",
            MessageBoxButton.OK, MessageBoxImage.Error);
    }
    haarCascade = new CascadeClassifier(Config.HaarCascadePath);
    faceList.Clear();
    string line;
    FaceData faceInstance = null;
    if (!Directory.Exists(Config.FacePhotosPath))
    {
        Directory.CreateDirectory(Config.FacePhotosPath);
    }
    if (!File.Exists(Config.FaceListTextFile))
    {
        string text = "Cannot find face data file:\n\n";
        text += Config.FaceListTextFile + "\n\n";
        text += "If this is your first time running the app, an empty file will be created for
you.";
        MessageBoxResult result = MessageBox.Show(text, "Warning",
            MessageBoxButton.OK, MessageBoxImage.Warning);
        switch (result)
        {
            case MessageBoxResult.OK:
                String dirName = Path.GetDirectoryName(Config.FaceListTextFile);
                        }
    }
    StreamReader reader = new StreamReader(Config.FaceListTextFile);
    int i = 0;
    while ((line = reader.ReadLine()) != null)
    {
        string[] lineParts = line.Split(':');
        faceList.Add(faceInstance);
    }
```

```
            foreach (var face in faceList)
            {
                imageList.Push(face.FaceImage.Mat);
                nameList.Add(face.PersonName);
                labelList.Push(new[] { i++ });
            }
            reader.Close();
            if (imageList.Size > 0)
            {
                recognizer = new EigenFaceRecognizer(imageList.Size);
                recognizer.Train(imageList, labelList);
            }
        }
        private void ProcessFrame()
        {
            if (active)
            {
                bgrFrame = videoCapture.QueryFrame().ToImage<Bgr, Byte>();

                if (bgrFrame != null)
                {
                    try
                    {
                        Image<Gray, byte> grayframe = bgrFrame.Convert<Gray, byte>();
                        {
                            bgrFrame.Draw(face, new Bgr(255, 255, 0), 2);
                            detectedFace = bgrFrame.Copy(face).Convert<Gray, byte>();
                            FaceRecognition();
                            break;
                        }
                        CameraCapture = bgrFrame.ToBitmap();
                    }
                    catch (Exception ex)
                    {
                        throw new ArgumentException("Error: " + ex);
                    }
                }
            }
        }
        private void FaceRecognition()
        {
            if (imageList.Size != 0)
            {
                FaceRecognizer.PredictionResult result =
recognizer.Predict(detectedFace.Resize(100, 100, Inter.Cubic));
                FaceName = nameList[result.Label];
                CameraCaptureFace = detectedFace.ToBitmap();
            }
            else
```

```
    {
        FaceName = "Please Add Face";
    }
}
private BitmapImage BitmapToImageSource(Bitmap bitmap)
{
    using (MemoryStream memory = new MemoryStream())
    {
        bitmap.Save(memory, System.Drawing.Imaging.ImageFormat.Bmp);
        memory.Position = 0;
        return bitmapimage;
    }
}
#endregion
private void Button_Click_1(object sender, RoutedEventArgs e)
{
    if (!VerifyData())
        return;
    var res = GetUsers();
    if (res == null)
    {
        MessageBox.Show("Не вдала спроба входу- \"Кориистувача не знайдено\"");
        return;
    }
    var user = res.FirstOrDefault(x => x.UserName == UserName.Text);
    if (faceIdentification)
    {
        var w = Application.Current.Windows[0];
        w.Hide();
        captureTimer.Stop();
        videoCapture.Dispose();
        commonpage.Exit.Visibility = Visibility.Visible;
        enableAutoNavigate = false;
        MessageBox.Show("Спроба входу була вдалою!");
        commonpage.ShowDialog();
        w.Close();
    }
    else
    {
        MessageBox.Show("Не вдала спроба ідентифікації");
    }
}
private bool VerifyData()
{
    if (String.IsNullOrEmpty(UserName.Text))
    {
        MessageBox.Show("Введіть логін");
        return false;
    }
```

```
            if (imgCamera.Source == null)
            {
                MessageBox.Show("Сталася помилка. Зразук обличчя не вдалося отримати");
                return false;
            }
            return true;
        }
        private void newAccount_Click(object sender, RoutedEventArgs e)
        {
            active = false;

            var w = Application.Current.Windows[0];
            w.Hide();

            captureTimer.Stop();
            videoCapture.Dispose();

            Registration wfAbout = new Registration();
            wfAbout.ShowDialog();

            w.Close();
        }
        private List<User> GetUsers()
        {
            if (!exists)
                System.IO.File.Create(path);

            using (StreamReader r = new StreamReader(path))
            {
                string json = r.ReadToEnd();
                users = JsonConvert.DeserializeObject<List<User>>(json);
            }
            return users;
        }
        private void Login_OnClosed(object sender, EventArgs e)
        {
            captureTimer = null;
            active = false;
            if (enableAutoNavigate)
            {
                var w = Application.Current.Windows[0];
                w.Hide();
                CommonPage wfAbout = new CommonPage();
                wfAbout.ShowDialog();
                w.Close();
            }
        }
    }
}
```

Додаток Д. Інтерфейс додатку

Додаток Е. Ілюстративний матеріал

Додаток Ж. Протокол перевірки

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: Підвищення достовірності ідентифікації користувачів засобами технології Face ID на основі вдосконаленого методу фільтрування зображення
Тип роботи: Магістерська кваліфікаційна робота
Підрозділ: Факультет МІБ, кафедра менеджменту та безпеки інформаційних систем,
гр. УБ-20м
Науковий керівник Азарова А.О., професор каф. МБІС, к.т.н.

Показники звіту подібності

Plagiat.pl (StrikePlagiarism)		Unicheck	
КП1			
КП2		Оригінальність	87 %
Тривога/Білі знаки	/	Схожість	13 %

Аналіз звіту подібності (відмінити подібне)

- ☐ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений(-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор _____ Богачук В.В.
(підпис) (прізвище, ініціали)

Опис прийнятого рішення

Ступінь оригінальності роботи відповідає вимогам, що висуваються до МКР
Особа, відповідальна за перевірку_____ Коваль Н.П.
(підпис) (прізвище, ініціали)

Експерт _____ _____
(за потреби) (підпис) (прізвище, ініціали)