Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА

на тему:

Розробка програмного модуля захисту від НСД на основі методу локальних бінарних шаблонів

Виконав: студент 4 курсу, гр.1КІТС-20б
спеціальності 125– Кібербезпека
Освітня програма – Кібербезпека
інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)

Матвієнко Д. В.
(прізвище та ініціали)

Керівник: д.т.н., професор каф. МБІС
Яремчук Ю.Є.
(прізвище та ініціали)
«____» _____ 2024 р.

Рецензент: к.т.н., доц., доцент каф. ОТ
Крупельницький Л.В.
(прізвище та ініціали)
«____» _____ 2024 р.

Допущено до захисту
Голова секції УБ кафедри МБІС
д.т.н., проф. Яремчук Ю.Є.
“_____” _____2024р.

Вінниця ВНТУ – 2024

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

Рівень вищої освіти І-й (бакалаврський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітньо-професійна програма - Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС

д.т.н., проф. Яремчук Ю.Є.
“ 22 ” березня 2024 р.

З А В Д А Н Н Я
на бакалаврську дипломну роботу студенту
Матвієнку Данилу Валерійовичу
(прізвище, ім`я, по-батькові)

1. Тема роботи Розробка програмного модуля захисту від НСД на основі методу локальних бінарних шаблонів
Керівник роботи Яремчук Юрій Євгенович д.т.н., проф. каф. МБІС
(прізвище, ім`я, по-батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від “11” березня 2024року № 80
2. Термін подання студентом роботи 6 червня 2024р
3. Вихідні дані до роботи Електронні джерела та наукові статті по темі. Існуюче програмне забезпечення, яке стосується теми бакалаврської дипломної роботи.
4. Зміст текстової частини: для досягнення мети було визначено наступні завдання: провести дослідження актуальності обраної теми; провести аналіз методів захисту інформації; провести аналіз методів шифрування інформації; оцінити принцип роботи даних методів захисту; оцінити переваги та недоліки аналогічних сервісів; вдосконалити алгоритм роботи системи безпеки засобу автентифікації; реалізувати засіб автентифікації; провести тестування розробленого засобу.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень)
Метод головних компонент,приклад роботи методу локальних бінарних шаблонів, приклад аналізу зображення методом локальних бінарних шаблонів, зображення в інтегральному поданні, графічне пояснення до інтегральної форми зображення, ознаки Хаара, алгоритм програмного модуля блокування НСД, інтерфейс програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	Яремчук Ю.Є. д.т.н., проф. каф. МБІС		
Розділ II	Яремчук Ю.Є. д.т.н., проф. каф. МБІС		
Розділ III	Яремчук Ю.Є. д.т.н., проф. каф. МБІС		

7.Дата видачі завдання 22 березня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва та зміст етапу	Термін виконання		Примітк
		початок	закінчення	
1	Дослідження актуальності обраної теми	23.03.2024	30.03.2024	Виконан
2	Аналіз методів захисту інформації	1.04.2024	11.04.2024	Виконан
3	Дослідження актуальності обраної теми	12.04.2024	21.04.2024	Виконан
4	Дослідження інформаційних джерел	22.04.2024	9.05.2024	Виконан
5	Аналіз методів захисту інформації	10.05.2024	16.05.2024	Виконан
6	Розробка алгоритму роботи	17.03.2024	21.05.2024	Виконан
7	Реалізація поштовго клієнта	22.05.2024	25.05.2024	Виконан
8	Тестування програмного модуля	26.05.2024	28.05.2024	Виконан
9	Оформлення матеріалів до захисту БДР	29.05.2024	2.06.2024	Виконан

Студент Матвієнко Д.В.
(підпис) (прізвище та ініціали)

Керівник роботи Яремчук Ю.Є.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

У цій дипломній роботі розглядається розробка та впровадження програмного модуля для захисту від несанкціонованого доступу з використанням методу локальних бінарних шаблонів, що забезпечує аутентифікацію за допомогою розпізнавання обличчя. Для реалізації даного сервісу була вибрана мова програмування Python.

В процесі дослідження було виконано глибокий аналіз предметної області, визначено актуальні проблеми та проведено порівняльний аналіз існуючих рішень, їхніх сильних та слабких сторін. На основі отриманих результатів був теоретично спроектований додаток та створений програмний модуль для його реалізації.

У підсумку дослідження було розроблено високоефективний і конкурентоспроможний додаток для розпізнавання обличчя, який відрізняється підвищеним рівнем безпеки. Це має критичне значення для забезпечення конфіденційності та захисту інформації від несанкціонованого доступу.

Ключові слова: несанкціонований доступ, біометрична автентифікація, програмний модуль, метод локальних бінарних шаблонів.

ANNOTATION

This thesis deals with the development and implementation of a software module to protect against unauthorized access using the method of local binary patterns, which provides authentication using facial recognition. The Python programming language was chosen for the implementation of this service.

In the course of the research, an in-depth analysis of the subject area was performed, actual problems were identified and a comparative analysis of existing solutions, their strengths and weaknesses was carried out. Based on the obtained results, the application was theoretically designed and a software module was created for its implementation.

As a result of the research, a highly effective and competitive application for face recognition was developed, which is characterized by an increased level of security. This is critical to ensuring privacy and protecting information from unauthorized access.

Keywords: unauthorized access, biometric authentication, software module, method of local binary patterns.

ЗМІСТ

ВСТУП

Ця дипломна робота присвячена розробці програмного забезпечення для протидії несанкціонованому доступу (НСД) до інформаційно-телекомунікаційних систем (ІТС) на об'єктах критичної інфраструктури (ОКІ). Робота є надзвичайно актуальною, оскільки в сучасних умовах бракує єдиного підходу до забезпечення кібербезпеки, а існуючі програмні рішення часто виявляються недосконалими. Аналіз науково-технічної літератури показує, що, незважаючи на наявність значних наукових досягнень у цій галузі, їх ефективність все ще залишається на недостатньому рівні. Це підтверджується високим рівнем успішних кібератак, як на міжнародному, так і на національному рівні.

Основною метою роботи є створення програмного забезпечення, здатного ефективно захищати інформаційно-телекомунікаційні системи від несанкціонованого доступу, незалежно від типу об'єкта.

Предметом даного дослідження є програмне забезпечення, створене для запобігання порушенням цілісності, конфіденційності та доступності інформації, що можуть виникнути через порушення правил розмежування доступу.

Об'єктом дослідження є процеси зберігання та обробки біометричних даних, які включають механізми захисту конфіденційності, цілісності та доступності. Важливою частиною роботи є розробка та впровадження методів протидії несанкціонованому доступу до інформаційно-телекомунікаційних систем.

1 ТЕОРЕТИЧНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

У цьому розділі буде проведено аналіз предметної області з метою визначення актуальності дослідження та розробки програмного модуля для запобігання несанкціонованому доступу.

1.1 Аналіз інформаційно-комунікаційних систем та шляхів НСД до них

Забезпечення кібербезпеки є надзвичайно важливим у сучасному світі, де інформаційно-телекомунікаційні системи (ІТС) відіграють ключову роль у функціонуванні різних організацій та інфраструктур. Несанкціонований доступ до ІТС може призвести до серйозних наслідків, включаючи втрату конфіденційної інформації, порушення цілісності даних та збої у роботі критично важливих систем[1].

Захист інформації, яка обробляється, зберігається та передається в ІТС, полягає в створенні та проведенні технічних й організаційних заходів, які дозволяють своєчасно виявляти та ефективно протидіяти потенційно- можливим інформаційним загрозам, а також знизити майбутні наслідки від їх реалізації, тобто захист інформації спрямовано на досягнення високого рівня безпеки інформації та ІТС в цілому[2].

Досягнення належного рівня кібербезпеки в інформаційно-телекомунікаційних системах (ІТС) може бути забезпечене шляхом реалізації визначених організаційних заходів. Проте, на практиці помітне значне зростання впровадження та використання технічних засобів і систем захисту. Це пов'язано з постійно зростаючими загрозами та складністю сучасних ІТС.

Виходячи з вимог функціонування ІТС, основні властивості інформації в контексті кібербезпеки можна класифікувати наступним чином[3]:

Конфіденційність - забезпечення доступу до інформації тільки авторизованим користувачам. Це включає використання методів шифрування даних, систем контролю доступу та аутентифікації користувачів. Біометрична аутентифікація, така як розпізнавання облич на основі методу локальних бінарних шаблонів (LBP), дозволяє значно підвищити рівень конфіденційності, забезпечуючи надійний

захист від несанкціонованого доступу.

Цілісність - захист інформації від несанкціонованих змін і забезпечення її достовірності. Для цього використовуються криптографічні методи, цифрові підписи та алгоритми хешування, які дозволяють виявити будь-які спроби зміни даних. Забезпечення цілісності є критичним для підтримки довіри до збереженої та переданої інформації.

Доступність - забезпечення постійного та безперебійного доступу до інформації для авторизованих користувачів. Це включає захист від атак типу "відмова в обслуговуванні" (DoS), використання резервного копіювання даних та систем відновлення після збоїв. Доступність є важливою для забезпечення безперервності бізнес-процесів та надання послуг.

Впровадження цих заходів вимагає комплексного підходу, який включає як організаційні, так і технічні аспекти. Організаційні заходи включають розробку політик безпеки, навчання персоналу та регулярний аудит систем. Технічні заходи охоплюють використання сучасних технологій і інструментів для захисту інформації на всіх етапах її обробки, зберігання та передачі.

Таким чином, для досягнення високого рівня кібербезпеки в ІТС необхідно поєднувати організаційні та технічні заходи, забезпечуючи комплексний підхід до захисту конфіденційності, цілісності та доступності інформації.[4].

Стандартні архітектурні принципи побудови, технічне оснащення, програмне та програмно-апаратне забезпечення, висока мобільність та інші характеристики визначають певний рівень доступу до інформаційних ресурсів та систем. Коли користувачі або групи користувачів використовують ІТС, виникає потреба у розмежуванні доступу, своєчасному виявленні та протидії несанкціонованому доступу (НСД).

НСД визначається як доступ до інформації в ІТС з порушенням встановлених правил розмежування доступу. Несанкціонований доступ може здійснюватися через помилки, допущені контролюючими структурами або системами безпеки мережевих компонентів ІТС. Це може включати підміну документів, що засвідчують особу, або незаконне отримання інформації про іншу особу, якій було

надано доступ.

Основними шляхами скоєння НСД до ІТС є:

- помилки в налаштуванні систем безпеки, тобто недосконалість конфігурації систем безпеки може призвести до виникнення вразливостей, які зловмисники можуть використати для отримання несанкціонованого доступу.;
- підміна документів, коли зловмисники можуть використовувати підроблені документи для видавання себе за авторизованих користувачів і отримання доступу до конфіденційної інформації;
- протиправне заволодіння інформацією про іншу особу, якій надано доступ, може дозволити зловмиснику використовувати ці дані для несанкціонованого доступу.

Причини виникнення НСД можуть бути наступними:

- невірно налаштована система моніторингу доступу до даних;
- організація захисту елементів ІТС на недостатньому рівні;
- використовується застаріле ПЗ;
- зловживання довірою та (або)службовими повноваженнями.

Для ефективного захисту від НСД необхідно впроваджувати комплексні заходи, такі як:

- розмежування доступу - Визначення чітких правил та політик щодо доступу користувачів до інформаційних ресурсів, що базуються на принципі найменших привілеїв.
- Моніторинг та виявлення загроз - Постійний моніторинг активності в мережі та використання систем виявлення вторгнень для своєчасного виявлення спроб НСД.[5].
- аутентифікація та авторизація - використання надійних методів аутентифікації, таких як біометрична аутентифікація на основі розпізнавання обличчя, та багаторівнева авторизація для захисту доступу до критичних ресурсів.
- навчання персоналу, тобто регулярне навчання користувачів та адміністраторів щодо основ кібербезпеки та нових методів захисту від НСД.

На високому рівні значимості стоїть стійкість системи ідентифікації та автентифікації, оскільки це безпосередньо впливає на здатність системи забезпечувати відповідність доступу інформації та ресурсів лише легітимним користувачам. У контексті цих понять, надійність гарантується шляхом забезпечення високих стандартів захисту, які перешкоджають потенційним зловмисникам у втіленні їхніх замислів. Чим складніші та більш надійні механізми автентифікації, тим ефективніше система в цілому.

На сьогоднішній день існують різні методи автентифікації, які класифікуються залежно від наявності у кожного користувача певних характеристик:

1. Щось, що ви володієте (Something you have): Цей тип автентифікації базується на унікальному об'єкті, який володіє кожен користувач. Це може бути фізичний предмет, такий як ключ-карта або фізичний токен.

2. Щось, що ви знаєте (Something you know): Цей метод вимагає від користувача знання певної інформації, яка перевіряється перевіряючою стороною. Це може бути пароль, PIN-код або відповідь на секретне питання.

3. Щось, що ви є (Something you are): Цей тип автентифікації використовує біометричні характеристики користувача, які є унікальними для кожної людини. Це може бути відбиток пальця, розпізнавання обличчя, голосу або інші біометричні параметри.

Комбінація цих методів часто використовується для підвищення рівня безпеки в системах автентифікації, забезпечуючи більш стійкий індивідуальний підхід до перевірки ідентичності користувача.

Прикладами методів автентифікації першої групи можуть бути посвідчення або перепустки, наявність яких надає право людині отримати доступ на вхід у певні приміщення.

До другої групи методів автентифікації відносяться методи з використанням паролів. Вони є надзвичайно розповсюдженими через низьку ціну їх впровадження. У сучасних операційних системах та великій кількісті застосунків наявний парольний захист.

Основною особливістю третьої групи є використання унікальних

характеристик користувача. До цих характеристик можна віднести відбитки пальців людини, структура райдужної оболонки ока і так далі. У них використовуються прилади, що вимірюють дані характеристики та порівнюють їх з еталонними, при цьому підробити їх майже неможливо.

Біометрична ідентифікація представляє собою ефективний метод автентифікації, оскільки використовує унікальні фізіологічні чи поведінкові характеристики кожної людини. Ці характеристики можуть бути статичними або динамічними.

Статичні методи біометричної ідентифікації базуються на аналізі стійких фізіологічних особливостей, які залишаються сталими протягом життя людини, таких як папілярні лінії пальців, райдужна оболонка ока і т.д. Ці характеристики є унікальними для кожної особи з народження і є основою для точної ідентифікації.

Динамічні методи біометричної ідентифікації враховують змінні аспекти, такі як почерк або голос, які можуть змінюватися з часом в залежності від фізіологічних чи психологічних факторів.

Порівняно з іншими методами автентифікації, біометрична ідентифікація забезпечує вищий рівень безпеки, оскільки вона базується на неповторних особливостях кожної людини, які важко підробити чи відтворити. Цей підхід робить біометричну ідентифікацію особливо ефективною для захисту конфіденційної інформації та ресурсів.

До даної групи методів ідентифікації належать методи, що порівнюють отримані значення з еталонними за відбитками пальців, геометрією обличчя, ДНК, сітківкою людського ока.

Розпізнавання обличчя є одним з найбільш популярних методів ідентифікації, який базується на унікальних рисах обличчя та формі черепа кожної людини. Цей метод має декілька значних переваг:

1. **Відсутність прямого контакту**: Для розпізнавання обличчя не потрібен прямий дотик до пристрою. Це забезпечує зручність та гігієнічність процедури ідентифікації.

2. **Дистанційне розпізнавання**: Завдяки сучасному обладнанню можливо

розпізнати особу на великій відстані, навіть серед товпи людей. Це робить метод ефективним для використання в різних сценаріях, включаючи великі громадські заходи або контроль на територіях з високим рівнем безпеки.

3. **Не потребує спеціалізованої техніки**: Порівняно з іншими методами ідентифікації, які можуть вимагати дорогого обладнання або складної настройки, розпізнавання обличчя можна виконати за допомогою стандартного обладнання, такого як веб-камера чи мобільний телефон з камерою.

4. **Доступність обличчя користувача**: Більшість людей завжди мають доступ до свого обличчя, що робить процедуру ідентифікації зручною та надійною в повсякденному використанні.

5. **Динамічна ідентифікація**: Розпізнавання обличчя можливе з динамічним зображенням, що означає, що процес може відбуватися в реальному часі, не потребуючи статичного фото чи зображення.

Ці переваги роблять розпізнавання обличчя одним із найбільш зручних та ефективних методів автентифікації, особливо в умовах, де важлива висока ступінь безпеки та зручність для користувачів.

Процес ідентифікації користувачів на основі геометрії обличчя залежить від різних чинників, які можуть впливати на його ефективність. Наприклад, фізіологічні ознаки обличчя змінюються з віком, а також можуть бути впливові емоційний стан людини і фізичні атрибути, такі як волосяний покрив, окуляри або макіяж [6].

З метою підвищення точності ідентифікації використовується стратегія збільшення обсягу навчальної вибірки. Це означає збільшення кількості фотографій однієї людини, зафіксованих у різних психологічних станах. Наприклад, цей підхід дозволяє вирішити завдання ідентифікації двох близнюків, які мають схожі фізичні ознаки.

Динамічні методи ідентифікації базуються на поведінкових характеристиках людини, які змінюються з часом, але залишаються індивідуальними. Прикладом таких методів є клавіатурний почерк.

Для виявлення зображення людського обличчя на відеопотоці базовими є так

звані примітиви. Ці ознаки розраховуються на основі скануючого вікна динамічної розмірності, яке переміщується по відеозображенню. Щоб покращити точність і оперативність аналітичних розрахунків, відеоінформація зазвичай подається в інтегральній формі, і застосовується boosting для оптимізації процесу аналізу.

1.2 Огляд методів виявлення обличчя людини на зображенні

Одним із найпоширеніших і найефективніших методів ідентифікації користувачів є розпізнавання за унікальними рисами обличчя. Цей метод базується на порівнянні характеристик людини, яка намагається пройти ідентифікацію, з даними, що зберігаються в базі даних. Під час процесу розпізнавання система створює набір ключових ознак обличчя за допомогою спеціалізованого алгоритму. Потім ці ознаки використовуються для точного визначення особи шляхом послідовного виконання кількох етапів.

Процес розпізнавання обличчя складається з декількох основних етапів. Спочатку система захоплює зображення обличчя і виділяє його ключові точки, такі як положення очей, носа, рота та контурів обличчя. Ці точки використовуються для створення унікального вектора характеристик, який ідентифікує особу.

Потім отримані характеристики порівнюються з еталонними зразками, що зберігаються в базі даних. Для цього використовуються різні алгоритми машинного навчання та обробки зображень, такі як нейронні мережі, аналіз головних компонент (PCA) або лінійний дискримінантний аналіз (LDA). Ці алгоритми дозволяють системі враховувати варіації в освітленні, виразі обличчя та інших чинниках, що можуть впливати на точність розпізнавання.

Однією з ключових переваг розпізнавання обличчя є його здатність працювати без необхідності прямого контакту з користувачем. Це робить його зручним для застосування в різних умовах, включаючи безпекові системи, мобільні пристрої та інші технологічні рішення.

Крім того, сучасні системи розпізнавання обличчя можуть використовувати додаткові технології, такі як тривимірне сканування та інфрачервоні камери, щоб підвищити точність ідентифікації. Це дозволяє вирішувати складні завдання, такі

як розпізнавання осіб у натовпі або ідентифікація людей з подібними рисами обличчя, наприклад, близнюків.

Завдяки постійному розвитку технологій і алгоритмів, розпізнавання обличчя стає дедалі надійнішим і широко застосовується у різних сферах, забезпечуючи високий рівень безпеки та зручності для користувачів.

Першим етапом процесу розпізнавання обличчя є виявлення обличчя на відео чи фотографії. На цьому етапі велика кількість факторів може впливати на якість результату, зокрема якість зображення та характеристики відеозапису. Важливі параметри включають відстань між обличчям і камерою, а також освітленість місця зйомки. В умовах відео зйомки особливе значення має рівень освітлення.

Через високу важливість якості зображення, наступний етап передбачає її оцінку. Серед ряду зображень необхідно відібрати ті, що відповідають умовам для успішного виявлення людини. Зображення повинні бути високої якості, обличчя повинно знаходитися на оптимальній відстані від камери, а освітлення має бути достатнім, але не надмірним. Обличчя не повинно бути закритим, і наявність бороди може впливати на точність розпізнавання.

Якщо всі ці умови виконуються, переходять до наступного етапу, де на основі особливостей ознак обличчя людини будується шаблон. На цьому етапі виділяються ключові риси обличчя, такі як розташування очей, носа, рота, а також контури обличчя.

На останньому етапі відбувається порівняння створеного шаблону обличчя з шаблонами, які зберігаються в базі даних системи. У простому випадку, система виконує пряме порівняння даних і визначає, чи знаходяться відмінності в допустимих межах. Якщо перевірка проходить успішно, користувач успішно ідентифікується.

Цей підхід дозволяє забезпечити високу точність розпізнавання обличчя, враховуючи різні фактори, що можуть впливати на якість зображення, і забезпечує надійну ідентифікацію користувачів у різних умовах.

Показником успішності біометричної аутентифікації є рівні помилок першого і другого роду, відомі як FRR (False Rejection Rate) та FAR (False Acceptance Rate).

FRR, або рівень хибних відмов, відображає відсоток легітимних користувачів, яких система помилково не допускає. Високий FRR означає, що система часто помиляється, відмовляючи доступ справжнім користувачам, що може створювати незручності та знижувати ефективність використання системи.

FAR, або рівень хибних прийняттів, показує відсоток нелегітимних користувачів, яких система помилково допускає. Високий FAR означає, що система недостатньо захищена від несанкціонованого доступу, що є серйозною загрозою для безпеки.

Балансування між FRR і FAR є критичним завданням при розробці біометричних систем аутентифікації. Оптимізація алгоритмів та налаштування порогових значень дозволяють знизити обидва типи помилок до прийнятного рівня, забезпечуючи при цьому високу надійність і безпеку системи[9].

Успішність та надійність процедури біометричної аутентифікації залежать від численних умов. Одним із важливих факторів є кількість записів у базі даних, оскільки більша кількість зразків може підвищити точність системи. Час, коли була знята фотографія, також має значення, оскільки обличчя людини змінюється з віком і може змінюватися навіть у короткостроковій перспективі в залежності від емоційного стану, повороту голови, наявності волосся на обличчі у чоловіків та використання косметики у жінок.

Існують різні методи виявлення обличчя, кожен з яких має свої особливості, сильні та слабкі сторони. Наприклад, метод головних компонент (PCA) використовується для зниження розмірності даних і виявлення найважливіших характеристик обличчя. Метод локальних бінарних шаблонів (LBP) фокусується на текстурних характеристиках обличчя, що робить його ефективним для розпізнавання при різних освітленнях і умовах. Метод Віоли-Джонса застосовує каскадний класифікатор на основі ознак Хаара для швидкого та ефективного виявлення обличчя в реальному часі.

Кожен з цих методів має свої переваги. PCA може бути дуже ефективним для зменшення складності моделі, але може втратити деякі деталі, важливі для розпізнавання. LBP добре працює з різними текстурами, але може бути чутливим

до великих змін у зовнішності. Метод Віоли-Джонса є дуже швидким, але може бути менш точним у складних умовах освітлення та фону.

Вибір методу залежить від конкретних вимог системи та умов, у яких вона буде використовуватися. Ідеальним рішенням може бути комбінація кількох методів для досягнення високої точності та надійності біометричної аутентифікації.

Метод головних компонент

Метод обчислення власних векторів зображень обличчя має як переваги, так і недоліки. Однією з основних переваг цього методу є висока швидкість розпізнавання. Процес передбачає використання навчальної вибірки зображень, для яких обчислюються власні вектори. Ці вектори потім використовуються для кодування інших зображень, що дозволяє швидко ідентифікувати нові зображення.

Проте, метод має кілька суттєвих недоліків. По-перше, додавання нових облич до бази даних вимагає повторного проведення всіх обчислень власних векторів для всієї вибірки. Це є трудомістким і часовитратним процесом, що ускладнює масштабування системи. По-друге, для досягнення точних результатів необхідно, щоб усі зображення у навчальній вибірці були зроблені в однакових умовах, з однаковим ракурсом обличчя для кожної людини. Це обмеження робить метод менш ефективним у середовищах, де зображення можуть мати значні відмінності в умовах зйомки.

У випадках, коли на підприємстві відбувається часта зміна працівників, цей метод стає непрактичним через необхідність постійного оновлення бази даних і повторного обчислення власних векторів. Незважаючи на це, до сильних сторін методу можна віднести компактне зберігання даних користувачів, оскільки використовується лише обмежена кількість власних векторів. Це дозволяє ефективно управляти обсягом даних і зменшувати вимоги до зберігання.

Загалом, метод обчислення власних векторів зображень обличчя є швидким і ефективним для розпізнавання, але його застосування обмежується певними умовами, які можуть ускладнювати його використання в динамічних або неконтрольованих середовищах.

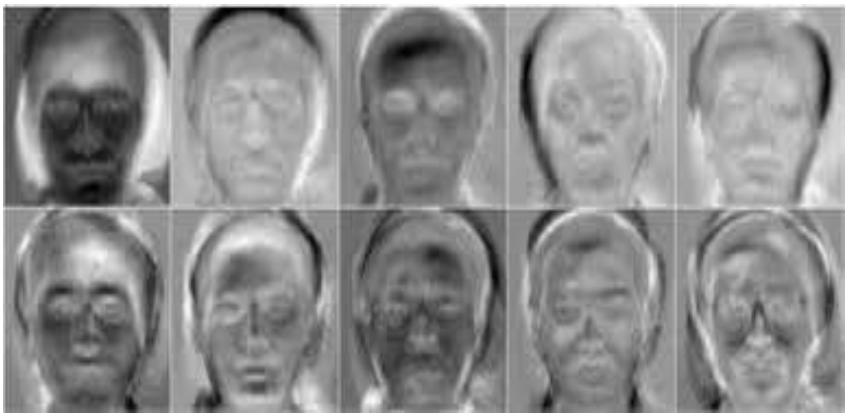


Рисунок 1.1 – Метод головних компонент

Метод локальних бінарних шаблонів

Ідея методу полягає в аналізі інтенсивностей кольору пікселів. Для кожного пікселя зображення аналізуються інтенсивності кольору восьми пікселів, що оточують його. Якщо інтенсивність одного з цих сусідніх пікселів більша або рівна інтенсивності центрального пікселя, йому присвоюється значення одиниці. Якщо інтенсивність менша, присвоюється значення нуля. Приклад використання даного методу зображений на рис 2.

5	6	1
2	4	4
2	7	3

1	1	0
0		1
0	1	0

Рисунок 1.2 Приклад роботи методу локальних бінарних шаблонів

До переваг цього методу можна віднести його стабільність до змін яскравості зображення та повороту голови, а також високу деталізованість результатів. Метод забезпечує надійну ідентифікацію навіть за умов варіативності освітлення та положення об'єкта.

Проте, метод має і свої недоліки. Для коректної роботи необхідно забезпечити високу якість зображення, оскільки він чутливий до дефектів і цифрового шуму. Наявність шуму може призвести до помилок в аналізі, що знижує точність і надійність розпізнавання. Тому перед застосуванням методу часто необхідна попередня обробка зображення для видалення шуму та покращення якості.

Рисунок 1.3 – Приклад аналізу зображення методом локальних бінарних шаблонів

1.3 Аналіз методу локальних бінарних шаблонів

Метод локальних бінарних шаблонів (LBP) є одним із найефективніших і найпоширеніших методів для аналізу текстури в комп'ютерному зорі та обробці зображень. Запропонований Тімо Оялом та його колегами у 1994 році, він базується на простій, але потужній ідеї порівняння кожного пікселя з його сусідами у локальному околі, що дозволяє кодувати текстурні властивості зображення у вигляді бінарних чисел.

Основний принцип методу полягає у створенні локального околу для кожного пікселя зображення. Найчастіше використовується окіл розміром 3x3, де центральний піксель порівнюється з восьми сусідніми. Якщо значення сусіднього пікселя більше або дорівнює значенню центрального пікселя, результат порівняння дорівнює 1, інакше — 0. Отримані бінарні значення об'єднуються у бінарний код, який перетворюється в десяткове число для подальшого аналізу.

Після обчислення LBP-кодів для всього зображення, формується гістограма, яка відображає розподіл цих кодів і слугує описом текстурних властивостей зображення. Така гістограма може бути використана для порівняння текстур різних зображень, класифікації текстур або розпізнавання об'єктів.

Однією з ключових переваг методу LBP є його незалежність від змін освітлення. Оскільки порівняння здійснюється між значеннями пікселів у локальному околі, абсолютні зміни освітлення, які однаково впливають на всі пікселі, не впливають на результати порівняння. Це робить метод стійким до різних умов освітлення. Крім того, метод є обчислювально ефективним і може бути

реалізований у реальному часі, що робить його привабливим для різних застосувань у комп'ютерному зорі, таких як розпізнавання облич, класифікація текстур і детектування об'єктів.

Проте метод LBP має і свої обмеження. Він може бути чутливим до шуму, особливо в однорідних областях зображення, де незначні зміни значень пікселів можуть суттєво впливати на бінарні коди. Використання фіксованого розміру околу також обмежує здатність методу захоплювати текстурні деталі на різних масштабах. Крім того, LBP кодує лише локальні відносини між пікселями, не враховуючи їх абсолютні позиції на зображенні, що може призвести до втрати просторової інформації.

Для покращення методу LBP було запропоновано кілька модифікацій. Наприклад, ротаційно інваріантний LBP (RLBP) дозволяє зменшити кількість можливих кодів і підвищити стійкість до обертання зображення шляхом обертання бінарного коду до мінімального значення. Мультироздільний LBP (MLBP) використовує декілька масштабів околу для кращого захоплення текстурних властивостей на різних масштабах. Вагові LBP (Weighted LBP) включають ваги для сусідніх пікселів залежно від їх відстані до центрального пікселя, що зменшує вплив шуму і підвищує точність.

1.4 Висновки та постановка задач

На підставі аналізу існуючих методів ідентифікації користувачів було встановлено, що ефективним засобом блокування несанкціонованого доступу (НСД) до інформаційно-телекомунікаційної системи (ІТС) є застосування методу біометричної ідентифікації осіб за геометрією обличчя. Для розробки програмного забезпечення блокування НСД були висунуті ряд функціональних, технічних та системних вимог. Таким чином, на підставі аналізу ІТС, шляхів НСД до нього, а також методів розпізнавання осіб за біометричними ознаками, виникає важливе науково-практичне завдання, яке полягає у розробці ПЗ для блокування НСД.

На основі глибокого аналізу поточної ситуації в галузі захисту інформації в інформаційно-телекомунікаційних системах (ІТС), було встановлено, що

використання біометричних технологій для ідентифікації осіб є найбільш перспективним напрямком для протидії несанкціонованому доступу. Цей підхід базується на унікальних фізіологічних або поведінкових характеристиках кожної людини, що робить його важким для підробки чи обходу.

Розробка програмного забезпечення (ПЗ) для блокування несанкціонованого доступу до ІТС вимагає врахування різноманітних функціональних, технічних і системних вимог. Особливу увагу слід звернути на стійкість та надійність біометричної системи, оскільки вона є основою для визначення і автентифікації користувачів. Вимоги до такого ПЗ включають точність розпізнавання, швидкість обробки даних, адаптивність до змін у фізичних параметрах осіб та умовах зйомки.

Крім того, для забезпечення ефективності системи необхідно розробляти алгоритми обробки зображень, які забезпечують стабільну роботу навіть у варіабельних умовах освітлення та при різних позах користувачів. Це означає, що алгоритми виявлення та аналізу обличчя мають бути достатньо адаптивними для впевненого розпізнавання осіб у реальному часі.

Отже, розробка ПЗ для захисту ІТС від несанкціонованого доступу за допомогою біометричних технологій є складним завданням, яке вимагає комплексного підходу до врахування всіх аспектів безпеки та ефективності.

Для вирішення цього завдання необхідно:

- проаналізувати математичний апарат методу біометричної ідентифікації користувачів за геометрією обличчя;
- розробити алгоритм ПЗ, на основі якої реалізувати програма блокування НСД до ІТС;
- перевірити працездатність та оцінити надійність розробленого ПЗ блокування НСД до ІТС.

.

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ

В даному розділі ми детально розглянемо процес проектування програмного додатку, який є ключовим етапом у створенні програмного модуля. В даному розділі розглянемо математичну модель, структуру та функціональність програмного додатку, а також проаналізуємо та виберемо мову програмування.

2.1 Розробка математичної моделі біометричної ідентифікації користувача

У даній роботі використовується метод біометричної ідентифікації осіб на основі геометрії обличчя для захисту інформації в інформаційно-телекомунікаційних системах (ІТС). Цей метод складається з кількох основних етапів. По-перше, виконується розрахунок контурів або геометричної структури зображення обличчя користувача. Це дозволяє отримати основні параметри обличчя, які будуть використовуватися для подальших обчислень.

Далі проводиться первинна обробка зображення для нормалізації його за яскравістю та іншими параметрами, що дозволяє стандартизувати дані і забезпечити однакові умови для подальшого аналізу. Після цього формується множина базових характеристик, які представляють унікальні ознаки обличчя користувача, такі як відстані між ключовими точками, кути між векторами та інші важливі параметри.

Останнім етапом є порівняння отриманих характеристик з еталонами, що зберігаються в базі даних. Під час порівняння застосовуються спеціалізовані алгоритми, які оцінюють ступінь відповідності між характеристиками користувача і відомими зразками. Цей процес забезпечує високу точність ідентифікації, але вимагає відповідної якості зображень і уважного контролю за процесом збереження і обробки персональних даних користувачів.

Знаходження геометрії обличчя користувача здійснюється максимально оперативно, при цьому комп’ютерне навчання класифікаційних ознак відбувається навпаки – довго [10]. Варто відмітити, що вказаний метод може бути реалізованим у вигляді програмного застосунку, що є особливо актуальнов умовах розвитку ІТ.

Метод локальних бінарних шаблонів ґрунтується на використанні інформації, що міститься в локальних областях зображення. Основним поняттям у цьому методі є локальний бінарний шаблон (LBP), який використовується для опису текстур і структури зображення.

Ідея полягає у тому, що кожен піксель у зображенні порівнюється з його сусідами. Він отримує бінарне значення (1 або 0), в залежності від того, чи більше або менше інтенсивності він має порівняно з кожним з його сусідів. Таким чином, кожен піксель замінюється на бінарний шаблон, що відображає структуру локальної області.

Для підвищення точності ідентифікації часто використовують каскади ознак Хаара. Ознаки Хаара дозволяють ефективно виявляти основні структурні елементи в зображенні, такі як різні типи граней, краї обличчя тощо. Каскади ознак Хаара дозволяють впроваджувати глибокі аналізи зображень, підвищуючи якість та точність процесу ідентифікації.

Інтегральне подання зображення представляє собою спосіб зберігання і обробки даних про інтенсивність пікселів. Це матриця, розміри якої співпадають з вихідним зображенням, де кожен елемент матриці містить суму інтенсивностей пікселів певної області зображення. Інтегральне подання забезпечує ефективний і швидкий доступ до інформації про яскравість пікселів у випадкових прямокутниках, що є важливим для швидкодії алгоритмів обробки зображень, зокрема методу локальних бінарних шаблонів.. Це показано на рис. 2.2 [11].

1	3	5	1
2	4	1	3
3	5	6	4
2	4	8	6

а) початкове зображення

1	4	9	10
3	10	16	20
6	18	30	38
8	24	44	58

б) інтегральна форма

Рисунок 2.1 – Зображення в інтегральному поданні

Для знаходження елементів зображення в інтегральній формі подання використовують формулу:

$$L(x,y) = \sum_{i=0}^{i \le x} \sum_{j=0}^{j \le y} I(i,j) \qquad (2.1)$$

де $I(x,y)$ – яскравість пікселя на зображенні.

Знайдене значення $L(x,y)$ являє собою суму усіх пікселів, що містяться умежах від (0,0) до (*x,y*). Для швидкого знаходження значень прямокутників всередині матриці використовується наступний вираз:

$$L(x,y) = I(x,y) - L(x-1,y-1) + L(x,y-1) + L(x-1,y) \qquad (2.2)$$

Вираз (2.2) розраховує значення суми яскравості пікселів будь-якого прямокутника із використанням лише чотирьох значень. Так, суму пікселей прямокутника ABCD, що виділено синім кольором та наведено на рис. 2.3, можна порахувати за формулою

$$ABCD = L_1 + L_2 - L_3 - L_4 \qquad (2.3)$$

Вираз (2.3) надає змогу швидко знаходити суму пікселів певногопрямокутника, значно пришвидшуючи обрахунки.

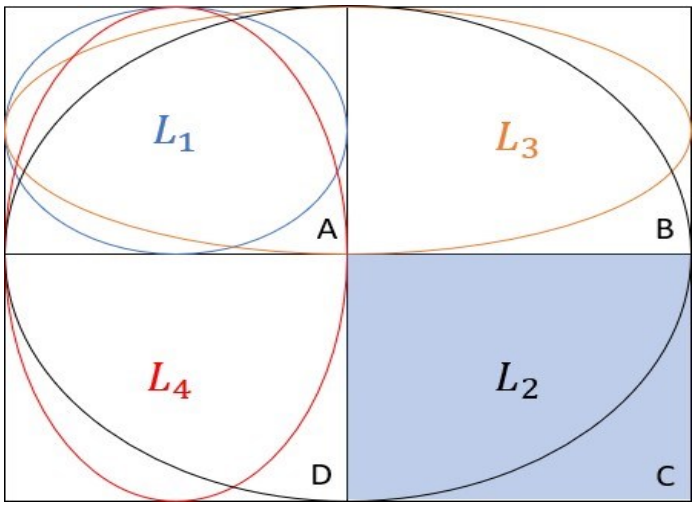


Рисунок 2.2 – Графічне пояснення до
інтегральної форми зображення

Під час сканування зображення застосовується вікно змінного розміру, яке поступово просувається по всьому зображенню для виявлення наявності ознак обличчя. Спочатку вікно просувається по осі x, зсуваючись на один піксель під час кожного аналізу. Після досягнення кінця ряду вікно зсувається вниз на один рядок по осі y і так далі, доки не буде проаналізовано всі зображення.

Важливою частиною процесу є використання значень пікселів, які варіюються від 0 до 255 у випадку однотонного зображення (grayscale) та від 0 до 255^3 для кольорових зображень. Оскільки grayscale споживає менше обчислювальних ресурсів, його доцільніше використовувати для аналізу. Це дозволяє здійснювати розрахунки значно швидше, що особливо важливо під час обробки великих обсягів даних, таких як великі набори зображень чи відео. Математично такий процес проводиться ізвикористанням виразу:

$$rectangle = \{(x, y), (w, h), \alpha\} \qquad (2.4)$$

де *w, h*, α – геометричні розміри (ширина, висота, кут нахилу) прямокутника.

Ознаками Хаара є відображення *f*:

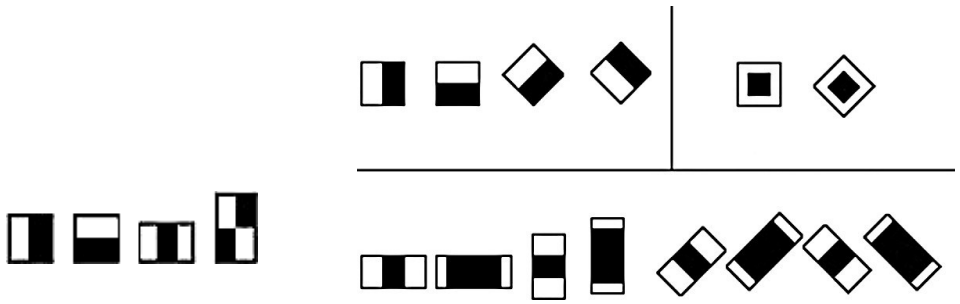
$$\chi => D_f \qquad (2.5)$$

де D_f – множина допустимих значень ознаки.

За умови, що ознака $f_1, \dots, f_n$ визначена, вираз (2.5) прийме вигляд:

$$\chi => \{f_1, \dots, f_n\} \qquad (2.6)$$

У методі локальних бінарних шаблонів застосовуються ознаки, відображення яких подане на рис. 2.4, а в удосконаленому – примітиви. Вони є складовою бібліотеки комп’ютерного зору OpenCV.



а) примітиви б) додаткові

Рисунок 2.3 —Ознаки Хаара

При функціонування методи є неможливим з практичної точки зору використовувати майже усі ознаки Хаара. А тому класифікатор має враховуватиме лише деякі найважливіші ознаки.

Метод локальних бінарних шаблонів використовується для визначення обличчя людини на зображеннях, а основою його є метод машинного навчання AdaBoost. Оскільки використання всіх ознак Хаара є непрактичним з технічної точки зору, класифікатор враховує лише найважливіші з них.

Для навчання класифікатора використовуються великі набори даних, які включають як зображення з обличчями та їх компонентами (ніс, очі тощо), так і зображення без об'єктів із шуканими ознаками. Цей процес є трудомістким через необхідність обробки великої кількості даних високої якості і вимагає значних обчислювальних ресурсів, можливо, кількох днів безперервних обрахунків.

Бібліотека OpenCV надає навчені класифікатори безкоштовно у вигляді XML-файлів, які вже містять моделі для виявлення обличчя людини та окремих його компонентів. Це значно спрощує процес розробки програм, що використовують

методи біометричної ідентифікації, так як розробники можуть скористатися готовими рішеннями з високою точністю.

2.2 Розробка алгоритму програмного модуля захисту від НСД

Алгоритм – це формально описана обчислювальна процедура, яка отримує вхідні дані, що називаються входом алгоритму або його аргументом,слідуючи якій можна досягнути поставлену мету.[12]

Графічне подання алгоритму у вигляді блок-схеми є досить зручним і ефективним способом візуалізації логіки і послідовності операцій, необхідних для вирішення конкретної задачі програмного забезпечення. Це форма представлення, при якій процес розв'язання задачі розбивається на окремі етапи, кожен з яких відображений у вигляді блоків. Кожен блок представляє тип дій або обчислень, які виконуються на кожному кроці алгоритму.

Графічна блок-схема дозволяє чітко показати послідовність кроків, умовні переходи, цикли та інші структури, які використовуються в алгоритмі. Це допомагає розробникам та інженерам програмного забезпечення краще розуміти, аналізувати та вдосконалювати алгоритми, а також спрощує комунікацію між членами команди, які працюють над проектом.

Однак, крім графічного представлення, існують і інші способи опису алгоритмів, такі як словесний опис, табличне представлення, псевдокод тощо. Вибір форми подання залежить від специфіки завдання, типу алгоритму і призначення його опису.

Алгоритм ПЗ блокування НСД до ІТС ОКІ складається з 7 основних блоків (схема алгоритму наведена на рис. 3.1.):

1. Блок завантаження бази даних та класифікатора.
2. Блок ініціалізація засобу відеоспостереження.

3-4. Блок пошуку (виявлення) геометрії обличчя на відеопотоці.

5-7. Блок ідентифікація (автентифікація) користувачів та блокування НСД до ІТС ОКІ

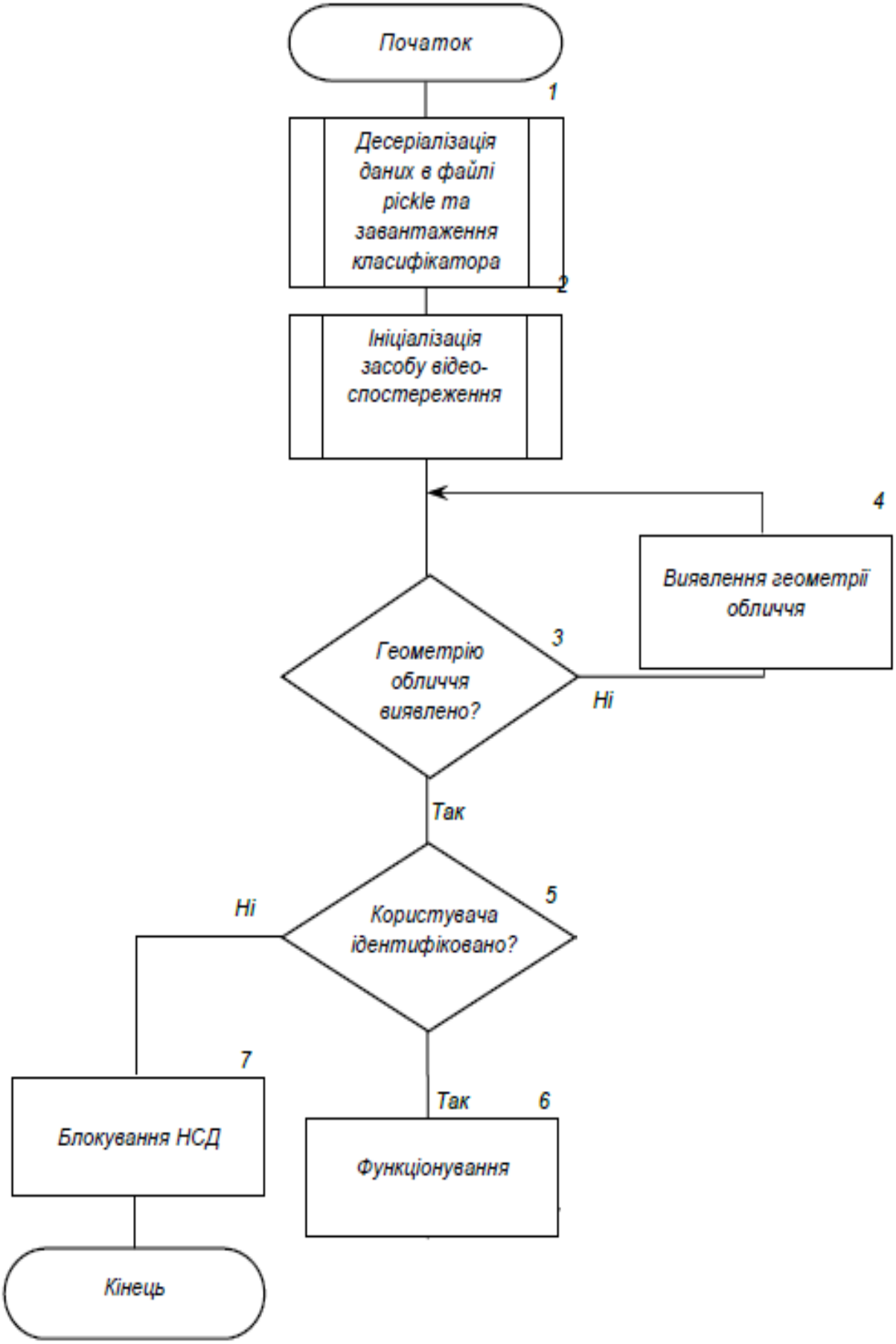


Рисунок 3.1 – Алгоритм програмного модуля блокування НСД

*Блок 1***.** Завантаження БД.

На першому етапі роботи програми відбувається завантаження даних, які містять унікальні біометричні дані користувачів. Ці дані знаходяться у файлі з розширенням pickle і зберігаються у вигляді потоку байтів для зручності зберігання і передачі. Після завантаження файлу відбувається процес десеріалізації, під час якого дані перетворюються з їхнього байтового представлення у структуру даних Python.

Після десеріалізації дані представлені у вигляді словника, де ключі відповідають порядковому номеру користувача, а значеннями є список з двома елементами: перший елемент це рядок з прізвищем користувача, а другий елемент, а саме масив, що містить характеристики, які є унікальними для даного користувача.

Додатково до цього відбувається завантаження xml файлу з класифікаторами, які будуть використовуватись у бібліотеці OpenCV для виявлення обличчя. Ці класифікатори містять у собі навчені моделі, які дозволяють ефективно виявляти та розпізнавати обличчя на зображеннях.

Загальний процес завантаження та підготовки даних є критичним для подальшого функціонування програми, оскільки від цього залежить правильність та швидкодія аналізу біометричних даних користувачів.

Блок 2 призначений для ініціалізації засобу відеоспостереження.

На цьому етапі програми ініціалізується об'єкт класу VideoCapture з бібліотеки OpenCV, який використовується для отримання відеокадрів з камери або з відеофайлу. Цей об'єкт дозволяє програмі отримувати послідовні кадри зображень для подальшої обробки і аналізу.

Після створення об'єкта VideoCapture програма завантажує дані з файлу, що містить унікальні біометричні дані користувачів. Ці дані включають імена користувачів, їх характеристики біометрії і зберігаються у вигляді словника, де ключами є порядкові номери, а значеннями - списки з прізвищами та характеристиками.

Крім цього, програма завантажує xml-файл з класифікаторами для виявлення обличчя за допомогою бібліотеки OpenCV. Ці класифікатори допомагають точно визначати положення обличчя на кожному кадрі відео.

Далі ініціалізуються змінні, які використовуються для подальшого розпізнавання і ідентифікації користувачів. Ці змінні включають місцезнаходження обличчя в кожному кадрі, характеристики обличчя поточного користувача для порівняння з вхідними даними та змінну для відображення імені користувача на екрані, якщо ідентифікація виявиться успішною.

Ці етапи ініціалізації дозволяють програмі ефективно працювати з відеоданими та виконувати завдання ідентифікації користувачів за їх біометричними даними.

Блок 3,4. Алгоритм виявлення ознак знаходження обличчя людини на зображенні.

На даному етапі програми передбачено зменшення розміру кадру у 4 рази, що дозволяє зменшити використання ресурсів і прискорити процес виявлення обличчя користувача. Крім того, змінюється кольорова палітра зображення для оптимізації обробки.

Далі, за допомогою функції face_locations, яка входить до складу модуля face_recognition, відбувається пошук та виявлення обличчя на зменшеному кадрі. Якщо характеристики користувача знайдені у файлі з біометричними даними, програма виділяє обличчя, намальовуючи прямокутник навколо нього.

Після цього програма відновлює розмір кадру до початкового стану (збільшує його у 4 рази). У нижній частині цього рамки відображається прізвище користувача, яке відповідає знайденим характеристикам.

Цей підхід дозволяє ефективно використовувати обмежені ресурси для швидкого виявлення та ідентифікації користувачів за їх біометричними даними.

Блок 5 – 7. Ідентифікація (автентифікація) користувачів та блокування

Після виявлення обличчя людини програма проводить порівняння характеристик обличчя користувача з збереженими даними характеристик працівників, які мають доступ до системи ІТС. Цей процес виконується за

допомогою функції compare_faces, яка знаходиться у підключеному модулі face_encodings.

Якщо характеристики обличчя користувача співпадають з даними характеристик працівників у встановлених межах, то людина має доступ до системи ІТС і може продовжувати роботу. У випадку, якщо зіставлення не підтверджує співпадіння, програма виконує блокування доступу.

Цей механізм забезпечує високий рівень безпеки і точності ідентифікації користувачів системи ІТС на основі їх біометричних даних.

2.3 Аналіз мов програмування і баз даних та обґрунтування вибору

Після завершення розробки архітектури проектування на основі методу локальних бінарних шаблонів наступним кроком є обґрунтований вибір середовища розробки та мови програмування для реалізації даного веб-сервісу. Важливо врахувати такі аспекти, як ефективність, швидкість розробки, масштабованість, підтримка засобів безпеки, доступність бібліотек та інструментів для роботи з зображеннями та обробки даних, а також можливість інтеграції з базою даних для зберігання та управління інформацією про скановані зображення.

Виходячи із поставленої мети та задач роботи, для розробки програмного модуля захисту від НСД було обрані такі засоби програмування:

Python - це мова програмування, яка призначена для розробки веб-застосунків та динамічних веб-сайтів;

MySQL - це безкоштовна реляційна система управління базами даних, яка використовується для зберігання та управління даними веб-додатків;

Далі в підрозділі розглянемо детальніше наведені засоби програмування та проаналізуємо основні переваги, що зумовили їх вибір для даної роботи.

Вибір мови Python для розробки програмного модуля блокування несанкціонованого доступу (НСД) має кілька важливих переваг. По-перше, Python відомий своєю простотою та зрозумілістю синтаксису, що спрощує написання та читання коду. Це дозволяє розробникам швидше зрозуміти та розробляти програмні модулі, а також полегшує підтримку та оновлення коду.

По-друге, Python має багату екосистему бібліотек і модулів, які можуть бути корисні для розробки систем безпеки. Наприклад, існують бібліотеки для роботи з мережею, шифруванням, аутентифікацією та іншими аспектами безпеки. Крім того, Python є кросплатформенною мовою, що дозволяє розробляти програми, які працюватимуть на різних операційних системах, таких як Windows, Linux та macOS. Це важливо для забезпечення захисту на різних платформах.

Ще однією перевагою є те, що Python дозволяє швидко створювати прототипи та тестувати ідеї. Це особливо корисно на початкових етапах розробки, коли важливо швидко перевірити функціональність та ефективність рішення. Велика та активна спільнота розробників Python означає, що є багато ресурсів, документації, прикладів коду та форумів, де можна знайти допомогу та поради. Це може прискорити вирішення проблем і підвищити якість коду.

Python легко інтегрується з іншими мовами програмування, такими як C/C++ для підвищення продуктивності, а також з різними системами та інструментами, включаючи бази даних, веб-сервіси та аналітичні платформи. Ще однією важливою перевагою є високий рівень безпеки завдяки можливостям управління пам'яттю, вбудованим методам захисту та численним бібліотекам для забезпечення безпеки даних і комунікацій.

Окрім вищезгаданих переваг, важливо врахувати, що Python активно використовується в академічних та наукових дослідженнях, що робить його ідеальним для впровадження новітніх досягнень у галузі кібербезпеки. Завдяки своїй гнучкості та потужності, Python є ідеальним вибором для розробки модулів, які вимагають високого рівня надійності та безпеки.

MySQL - це безкоштовна реляційна система управління базами даних, розроблена та підтримується корпорацією Oracle. Розглянемо переваги та особливості цієї системи управління базами даних для даного онлайн сервісу.

Універсальність та функціональність:

- MySQL надає засоби для створення, налаштування та адміністрування баз даних різного розміру та складності.

- Система пропонує різні типи таблиць, такі як MyISAM та InnoDB, з різними можливостями, що розширюють функціональність та ефективність.

Інструменти розробки:

- MySQL має інтегровані засоби для роботи з базами даних, включаючи редактори запитів, візуальні засоби керування об'єктами та інші інструменти, які спрощують розробку та адміністрування.

Надійність та продуктивність:

- MySQL відомий своєю стабільністю та продуктивністю, що дозволяє використовувати його для розробки навіть великих та вимогливих додатків.

MySQL є потужним та універсальним інструментом для управління базами даних, який відповідає вимогам розробки БД для веб-сервісу. Його широкий функціонал, надійність та підтримка роблять його відмінним вибором для розробки та адміністрування баз даних.

У сучасному світі розробка програмного забезпечення стала важливою складовою бізнесу та технологічного прогресу. Вибір мови програмування впливає на ефективність розробки, зручність підтримки та швидкість виконання програми. При виборі мови програмування та системи керування базами даних (СКБД) необхідно враховувати ряд факторів, таких як:

Продуктивність

Надійність

Швидкість розробки

Сумісність із завданнями проекту.

Python разом з MySQL гарний вибір для створення ефективних, надійних та продуктивних програмних модулів, що відповідають потребам сучасного бізнесу та користувачів.

Отже, враховуючи функціональні можливості обраних мов програмування, їх доступність та зрозумілість при наявності відповідних навичок, ми плануємо використовувати ці засоби для реалізації програмної частини даного проекту.

2.4 Висновки до розділу 2

Для вирішення поставленого науково-практичного завдання у розділі 1 було конкретизовано математичний апарат процесу ідентифікації користувачів, який базується на використанні розширеного алгоритму локальних бінарних шаблонів. Цей алгоритм обрано як найбільш доцільний для забезпечення ефективності і надійності ідентифікації, що є ключовим аспектом у розробці програмного забезпечення для блокування несанкціонованого доступу до ІТС.

Мова програмування Python була обрана для розробки програмного модуля через її широке використання в сфері розробки програмного забезпечення, високу швидкість виконання коду та наявність численних готових рішень і фреймворків, що значно спрощують процес розробки та підтримки проекту. Python дозволяє зосередитися на реалізації алгоритмів без необов'язкового глибокого занурення в деталі оптимізації, що особливо важливо в науково-дослідницьких ініціативах з ідентифікації осіб.

У даному розділі надано детальний опис структури та порядку роботи розробленого програмного забезпечення (ПЗ), яке було створено для ідентифікації користувачів у системах інформаційно-технічної безпеки (ІТС). ПЗ базується на використанні розширеного методу локальних бінарних шаблонів для надійної ідентифікації осіб.

Програмне забезпечення реалізує алгоритм ідентифікації користувачів за допомогою біометричних даних, які зберігаються у файлі у форматі pickle. Після завантаження даних, вони десеріалізуються та утворюють словник, де кожен користувач має унікальний порядковий номер із списком характеристик, що включають інформацію про обличчя та інші ідентифікаційні параметри.

ПЗ використовує модуль VideoCapture з бібліотеки OpenCV для відеозахоплення. Перед обробкою кожен кадр зменшується у 4 рази для ефективності і подальшого виявлення обличчя. Застосовується зміна кольорової палітри для полегшення обробки. За допомогою бібліотеки face_recognition

відбувається виявлення обличчя, після чого порівнюються його характеристики зі списком авторизованих користувачів за допомогою функції compare_faces.

Якщо обличчя ідентифікованого користувача збігається з одним із зареєстрованих і характеристики відповідають заданим межам, користувачу дозволяється доступ до системи ІТС. У протилежному випадку, доступ блокується. Результати ідентифікації відображаються на екрані, де показується ім'я користувача, який має доступ.

Таким чином, розроблене програмне забезпечення відповідає всім вимогам ефективності і безпеки, що були висунуті у першому розділі роботи, і є ефективним інструментом для захисту об'єктів інформаційно-технічної системи.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

В даному розділі зосередимося на програмній реалізації обраних підходів та концепцій, що були описані та проаналізовані у попередніх розділах. Буде розроблено інтерфейс програмного модуля, модулі захисту а також модуль блокування НСД.

3.1 Програмна реалізація користувацького інтерфейсу

Згідно з структурою алгоритмічної схеми було написано ПЗ блокування НСД до ІТС ОКІ, головне вікно даного ПЗ зображено на рисунку 3.1

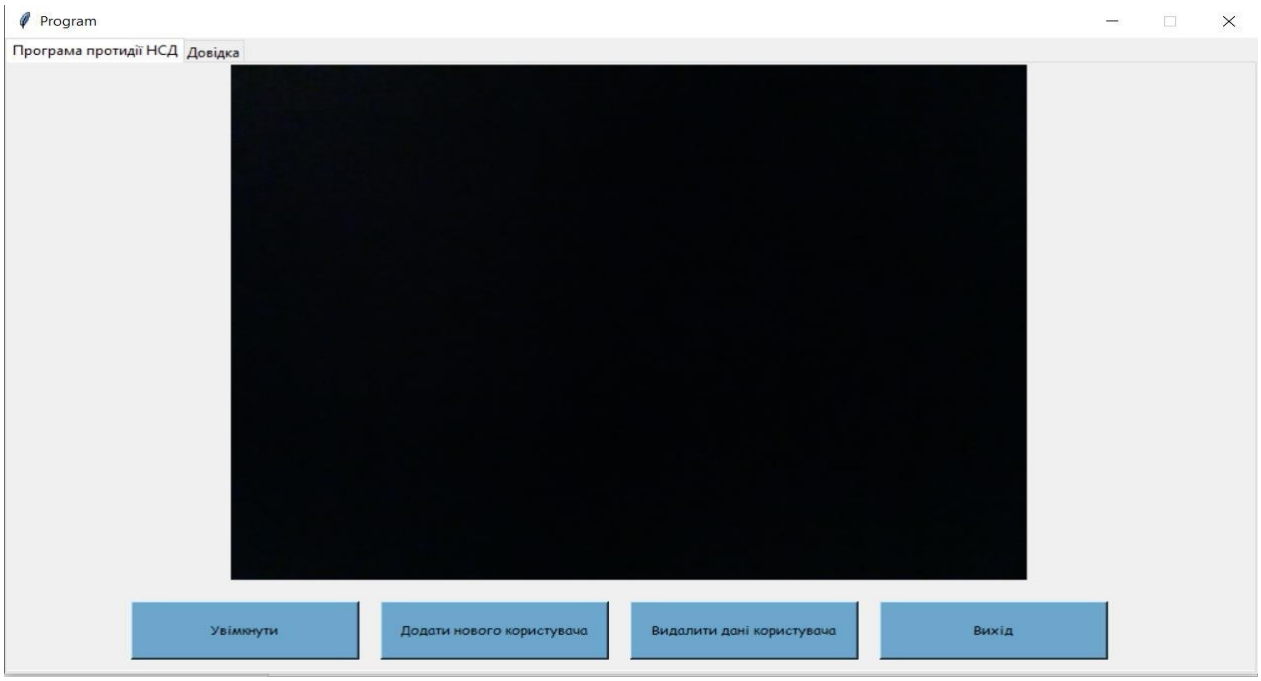


Рисунок 3.1 – Інтерфейс ПЗ з блокування НСД

В першій вкладці програми протидії НСД користувач знаходить вікно, де відображається відеопотік з веб-камери, спрямованої на об'єкт інформаційно-технічної системи (ІТС). Це вікно використовується для реального виявлення осіб, які намагаються отримати доступ до системи.

У вікні першої вкладки розміщені чотири кнопки. Перша кнопка, підписана як "Увімкнути", відповідає за активацію веб-камери та запуск процесу виявлення обличчя людини на відеопотоці. Після натискання цієї кнопки програма починає отримувати відеодані з веб-камери і виконує аналіз кадрів для виявлення присутності обличчя.

Процес виявлення обличчя виконується за допомогою алгоритмів комп'ютерного зору, що інтегровані у програму. Коли обличчя знайдено, програма переходить до наступного етапу, де проводиться порівняння характеристик обличчя зі збереженими даними у базі даних. Це дозволяє визначити, чи є особа авторизованою для доступу до ІТС.

Разом із кнопкою "Увімкнути", інші кнопки в цьому вікні можуть включати функції для зупинки відображення веб-камери, збереження або відображення результатів ідентифікації, а також інші дії, пов'язані з управлінням процесом виявлення і ідентифікації користувачів ІТС.

Кнопка "Додати нового користувача" в програмі відповідає за відкриття форми, яка зображена на рисунку 3.2. На цій формі користувач бачить текстове поле, призначене для введення прізвища працівника, яке використовується для ідентифікації в системі. Також на формі є кнопка, яка запускає процес додавання нових даних користувача до бази даних.

При натисканні користувачем цієї кнопки програма виводить сповіщення, що користувача буде сфотографовано для генерації біометричних даних. Після цього програма виконує знімок з веб-камери, який використовується для визначення характеристик обличчя користувача.

Отриманий знімок обробляється програмою для виділення області з обличчям за допомогою алгоритмів комп'ютерного зору. Ця область зберігається у форматі jpg у відповідній папці, що належить програмі. Цей зображення з обличчям буде використовуватися як частина біометричних даних користувача для подальшої ідентифікації в системі.

Процес додавання нового користувача передбачає не лише збереження зображення, але й занесення інших характеристик обличчя до бази даних, що дозволяє програмі ефективно ідентифікувати користувачів при подальших операціях з системою.

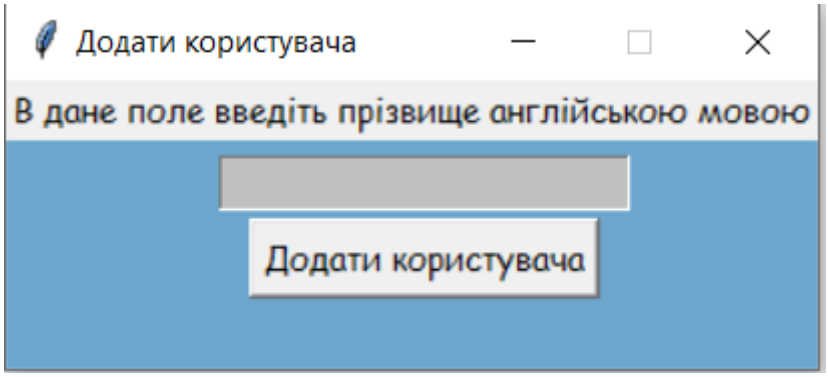


Рисунок 3.2 – Форма для додавання даних нового користувача

Кнопка "Видалити дані користувача" в програмі виконує аналогічну функцію, що зображена на формі, аналогічній рисунку 3.2. Однак її основна опція полягає у видаленні даних конкретного користувача з бази даних. Користувачу необхідно ввести ім'я працівника, для якого відсутній доступ до ІТС ОКІ з різних причин.

При натисканні кнопки "Видалити дані користувача" програма проводить видалення усіх збережених даних, що стосуються вказаного користувача, зокрема біометричні дані та всі інші пов'язані характеристики з бази даних програми.

Кнопка "Вихід" призначена для завершення роботи програми. При натисканні цієї кнопки програма закривається, а користувач повертається до основного робочого оточення.

У другій вкладці програми, яка зображена на рисунку 3.3, користувач знаходить коротку інформацію про цілі та функціонал програми. Ця вкладка призначена для надання користувачеві загального розуміння, які завдання програма вирішує і які можливості вона надає для захисту інформації таобмеження доступу до ІТС.

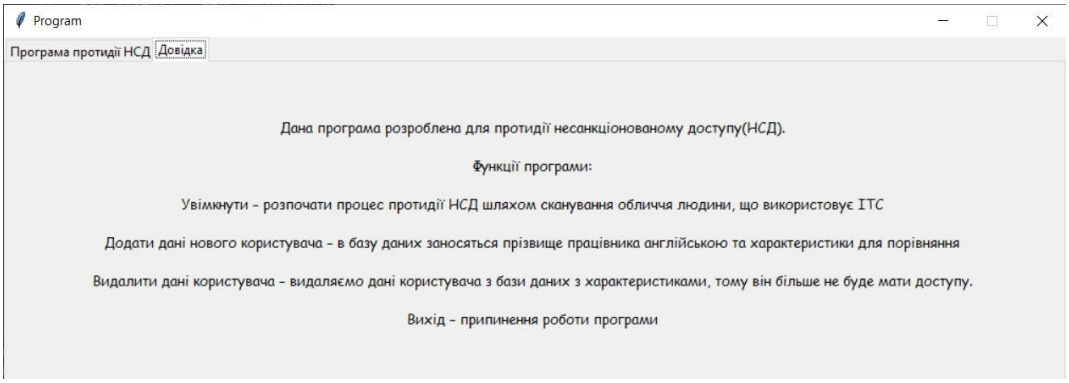


Рисунок 3.3 – Довідка

Після запуску програми відкривається головне вікно програми. Для початку роботи програми необхідно натиснути кнопку “Увімкнути”, але спершу необхідно для ідентифікації необхідно додати 128 характеристик обличчя людини, для цього треба натиснути кнопку Додати дані користувачата ввести ідентифікатор людини у відповідне поле форми, що зображено на рисунку 3.4

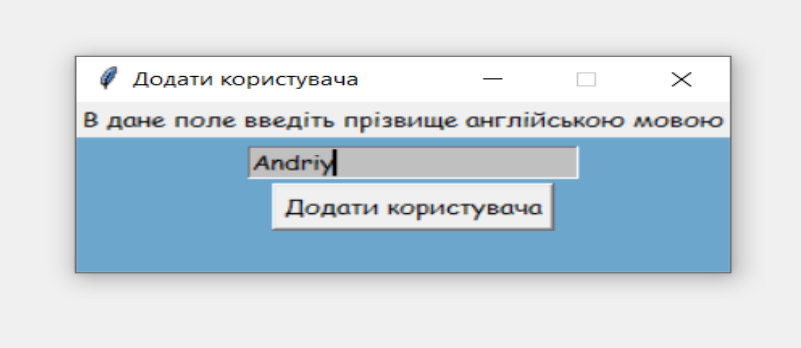


Рисунок 3.4 – Внесення ідентифікатора користувача

На етапі додавання нового користувача у програмі, після натискання кнопки "Додати нового користувача", відбувається підготовка до фотографування обличчя людини. Користувач отримує повідомлення у вікні MessageBox, яке інформує його про намір програми зробити фотографію обличчя. Повідомлення також вказує на необхідність стояти прямо перед веб-камерою на відповідній відстані, утримувати голову непорушно та не відхиляти обличчя вліво або вправо, оскільки це може негативно вплинути на точність процесу виявлення обличчя.

Після відображення цього повідомлення програма автоматично зроблює фотографію, в якій обличчя користувача виділене синьою рамкою. Цей етап важливий для збору біометричних даних користувача, які потім будуть використані для процесу ідентифікації під час подальшого доступу до інформаційно-технічних систем.

Після успішного фотографування обличчя дані про користувача, включаючи фотографію та інші біометричні характеристики, додаються до бази даних програми для подальшого використання у процесі ідентифікації.

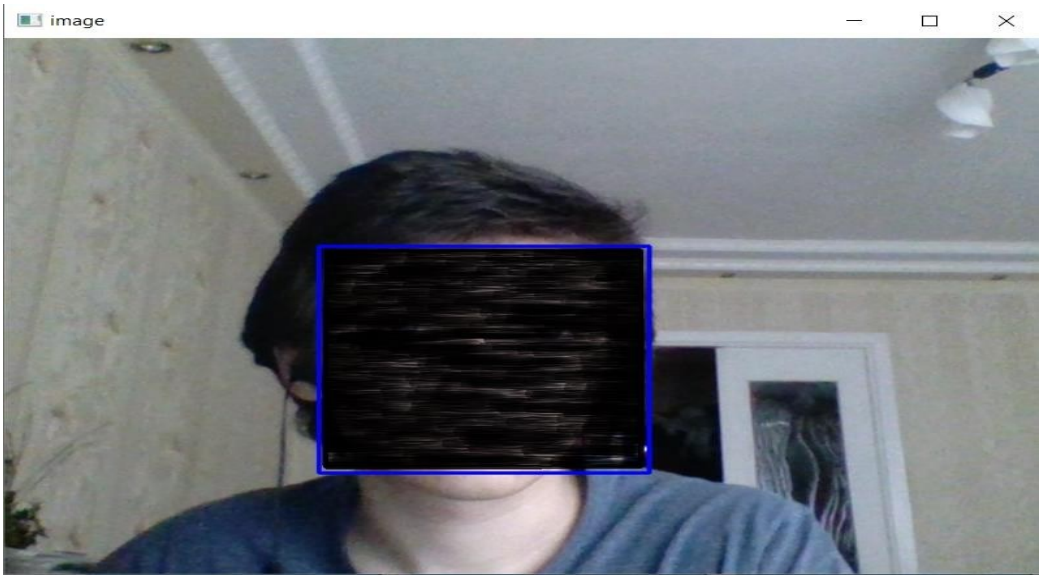


Рисунок 3.5 – Фото людини, дані якої заносяться в БД

Для того, щоб зміни вступили в силу, потрібно перезавантажити програму. Якщо цього не зробити, то при запуску програми ІТС автоматично заблокується, при цьому також вона робить фото порушника і зберігає її увідповідній папці, де знаходиться ПЗ. Приклад наведено на рисунку 3.6

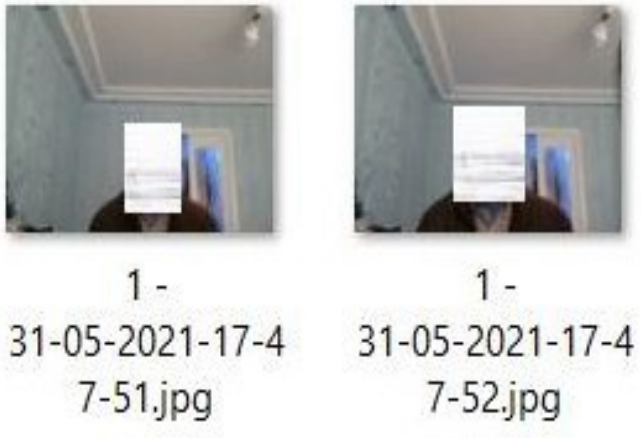


Рисунок 3.6 – Фіксування несанкціонованого користувача

Після того, як дані працівника занесені до бази даних програми, можна запустити програму, натиснувши кнопку "Увімкнути", яка розташована у лівому нижньому куті вікна. Після ініціалізації відеоспостереження через веб-камеру, відеопотік виводиться на головний екран програми. Якщо користувач має доступ до інформаційно-технічних систем (ІТС), то на екрані відображається зображення з веб-камери, а обличчя користувача, виявлене на зображенні, буде виділено червоним прямокутником. У лівому нижньому куті цього прямокутника буде

показано ідентифікатор користувача, який дозволяє його ідентифікувати (рисунок 3.7).

Цей функціонал дозволяє в реальному часі відслідковувати присутність та ідентифікувати користувачів, які мають доступ до інформаційно-технічних систем, що забезпечує безпеку та контроль за доступом.

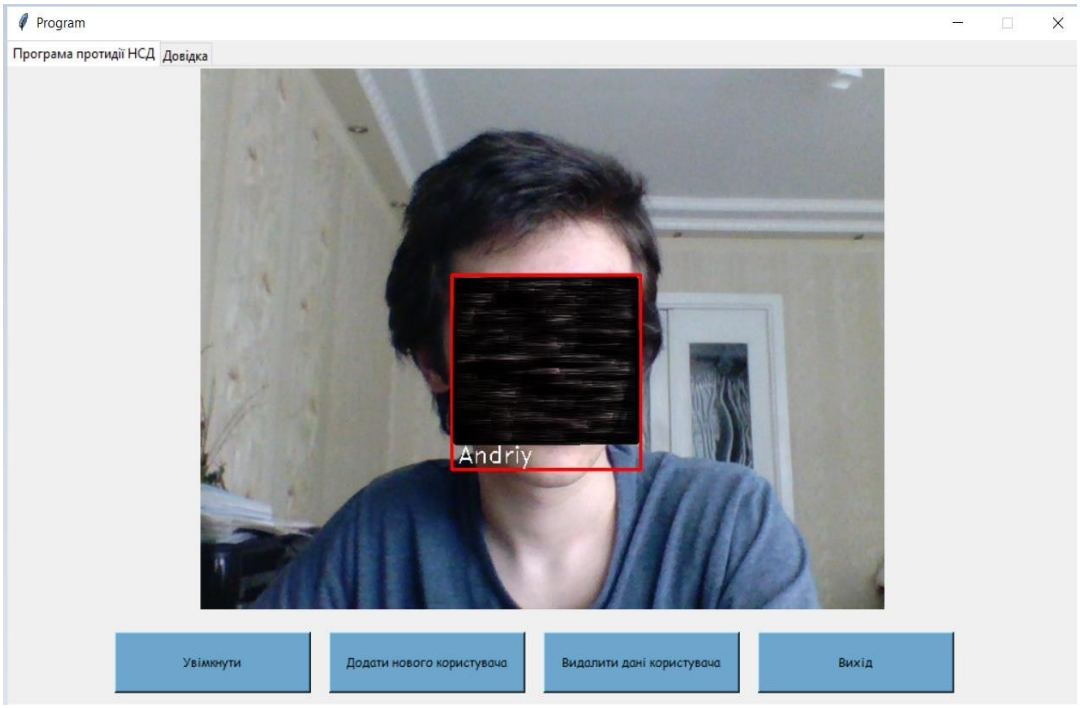


Рисунок 3.7 – Виявлення обличчя користувача

У випадку потреби заборонити доступ користувачеві до інформаційно-технічних систем (ІТС), необхідно натиснути кнопку "Видалити дані користувача". Після цього відкриється вікно, де потрібно ввести ідентифікатор користувача в відповідне поле, як показано на прикладі на рисунку 3.8. Якщо поле вводу залишити порожнім, користувач отримає повідомлення про відсутність введених даних і отримає можливість ввести прізвище користувача.

Після введення ідентифікатора користувача і виконання процесу видалення даних з бази даних, необхідно перезапустити програму, щоб зміни вступили в силу і враховувалися в подальшій роботі системи.

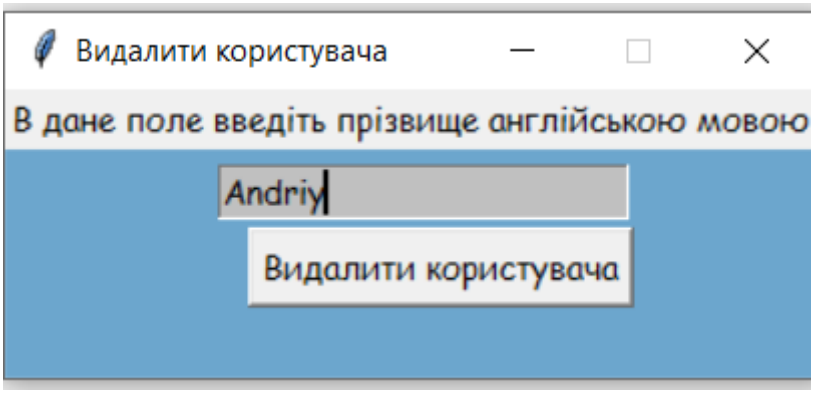


Рисунок 3.8 – Видалення даних користувача

Після проведення даної операції, працівник більше не зможе мати доступу до ІТС, приклад наведено на рисунку 3.9

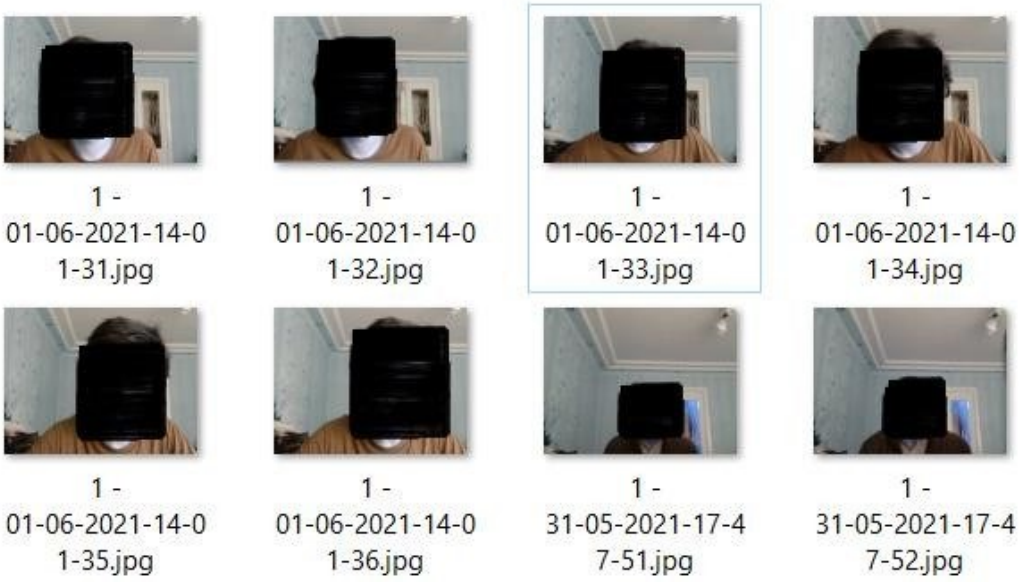


Рисунок 3.9 – Блокування доступу для видаленого користувача.

3.2 Програмна реалізація методу локальних бінарних шаблонів

Метод локальних бінарних шаблонів (Local Binary Patterns, LBP) використовується для аналізу текстур в зображеннях. Його основна ідея полягає в описі текстури за допомогою бінарних кодів, які формуються на основі порівняння інтенсивностей пікселів з інтенсивністю центрального пікселя.

Спочатку необхідно імпортувати потрібні бібліотеки для роботи. Ось приклад програмного коду:

```
import tkinter
from tkinter import messagebox
import cv2
```

```
import PIL.Image, PIL.ImageTk
from PIL import Image as Ims
from PIL import ImageTk
from PIL import ImageTk as IMTK
from cv2 import *
from tkinter import *
import numpy as np
import pickle
import os
from builtins import max as maximum
from time import sleep
import face_recognition
from tkinter import ttk
import tkinter.filedialog as tkFileDialog
from sys import exit
import warnings
import numpy as np
import time
```

Далі відбувається обчислення бінарного шаблону для кожного пікселя зображення. Функція обходить кожен піксель (крім країв) і порівнює значення центрального пікселя з його сусідами. Результат зберігається у вигляді цілого числа за допомогою побітових операцій.

Ось приклад блоку коду, що відповідає за це перетворення:

```
while True:
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    faces = face_detector.detectMultiScale(gray, 1.3, 5)
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
        count += 1
        cv2.imwrite(os.path.join(path_for_photo, "User." + str(face_id) + '.' + str(count) + ".jpg"),gray[y:y + h, x:x + w])
        image = face_recognition.load_image_file( os.path.join(path_for_photo, "User."+ str(face_id) + '.' + str(count) + ".jpg"))
        face_encoding = face_recognition.face_encodings(image)[0]
        users[str(face_id)] = [self.user_name_add.get(), face_encoding]
        cv2.imshow('image', img)
        with open('datab.pickle', 'wb') as f:
            pickle.dump(users, f)
            self.user_name_add.set('')
            k = cv2.waitKey(100) & 0xff
            if k == 27:
                break
            elif count >= 1:
                break
```

```
            cam.release()
            cv2.destroyAllWindows()
```

Останнім кроком алгоритм порівнює отримане значення з наявними у базі даних дозволу. Якщо значення відповідає, доступ надається, якщо ж відповідного значення в базу даних не знайдено, програма закриває доступ та робить фото порушника. Приклад програмного коду наведено нижче:

```
    while True:
            ret, img = cam.read()
           gray = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
           faces = face_detector.detectMultiScale(gray, 1.3, 5)
           for (x, y, w, h) in faces:
              cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
              count += 1
              cv2.imwrite(os.path.join(path_for_photo, "User." + str(face_id) + '.' + str(count) + ".jpg"),gray[y:y + h, x:x + w])
              image = face_recognition.load_image_file( os.path.join(path_for_photo,       "User."+ str(face_id)     +          '.'          + str(count) + ".jpg"))
              face_encoding = face_recognition.face_encodings(image)[0]
              users[str(face_id)] = [self.user_name_add.get(), face_encoding]
              cv2.imshow('image', img)
              with open('datab.pickle', 'wb') as f:
                 pickle.dump(users, f)
                 self.user_name_add.set('')
                 k = cv2.waitKey(100) & 0xff
                 if k == 27:
                    break
                 elif count >= 1:
                    break
              cam.release()
              cv2.destroyAllWindows()
```

Цей код використовує побітові операції для обчислення значення бінарних шаблонів без необхідності зберігати бінарний рядок в списку, що робить його більш ефективним і простим. Сам алгоритм використовує вбудовані бібліотеки Python, такі як обробка зображення за допомогою OpenCV, функцію розпізнавання лиця face_recognition з базового пакету Pillow, а також систему управління баз даних MySQL.

3.3 Тестування та оцінка результатів роботи програмного модуля

Дослідження проводилося за умовами, які відтворюють реальні сценарії використання системи ідентифікації осіб на основі біометричних даних обличчя.

Використовувалась вбудована в ноутбук веб-камера USB2.0 HD UVC з роздільною здатністю 1280×720. Основна мета експериментів полягала у визначенні ефективності системи за різних умов освітлення та відстані до обличчя.

Процес дослідження включав створення профілів для декількох користувачів, що передбачало введення їх біометричних даних до бази даних програми. Перевірка системи проводилась у приміщенні з різними рівнями освітлення: хорошій природній світлі, поганій природній освітленості та штучному освітленні. Це дозволило оцінити стійкість алгоритму в умовах, які можуть виникати в реальному використанні, наприклад, у приміщеннях зі складними умовами освітлення.

Відстань між веб-камерою та обличчям, яке підлягало ідентифікації, варіювалась від 20 до 100 см з кроком у 20 см. Це дозволило з'ясувати, як далеко користувач може знаходитись від камери, щоб система ефективно розпізнавала обличчя.

Дослідження включало аналіз як для авторизованих користувачів з наявними профілями в базі даних, так і для несанкціонованих користувачів, для яких очікувалося блокування доступу. Кожен експеримент був повторений 50 разів при кожній умові освітлення, що забезпечило статистичну достовірність результатів.

Отримані дані вказали на високу ефективність системи ідентифікації в умовах хорошої природної та штучної освітленості, але з низьким рівнем успішності в умовах поганої природної освітленості. Результати дослідження слугують основою для подальшого вдосконалення алгоритмів і розробки системи ідентифікації на основі біометричних даних обличчя.

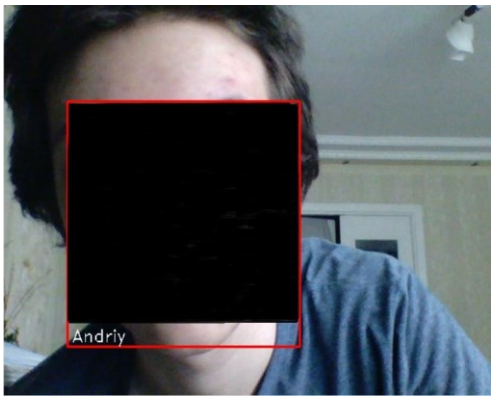
Результат досліджень при хорошій природній освітленості результати наведені у таблиці 3.1

Таблиця 3.1 – Результати досліджень роботи програми

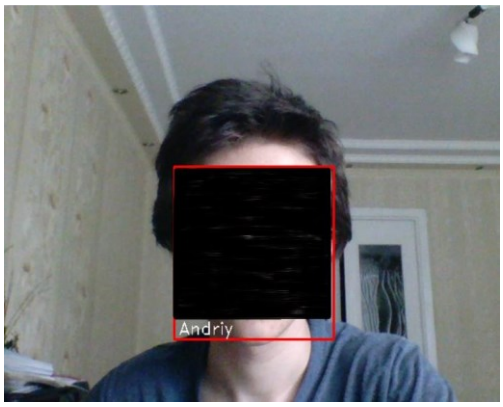
Відстань, см	Вдалі ідентифікації	Невдалі ідентифікації	Заблоковано неавторизованих користувачів
20	96	4	3
40	88	12	9
60	82	18	15
80	74	26	22
100	65	35	28

Для проведення експерименту за умов хорошої природньої освітленості була обрана світла пора доби, приблизно середина дня, коли приміщення мало оптимальне освітлення від сонячних променів. Це забезпечило найкращі умови для використання вбудованої веб-камери з роздільною здатністю 1280×720 для зйомки облич у високій якості.

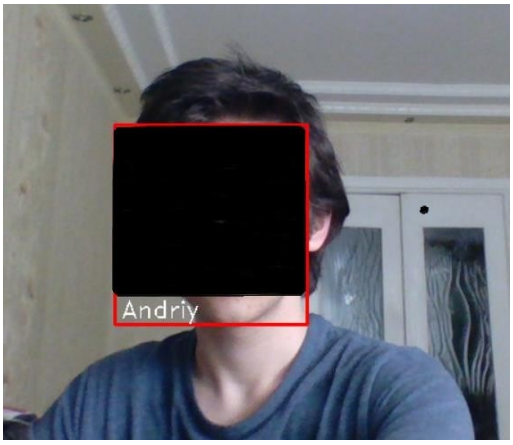
Дослідження також включало моделювання умов поганої природньої освітленості, коли приміщення мало мінімальне освітлення від природних джерел, а також штучної освітленості, що імітувало умови вечірнього чи нічного часу.



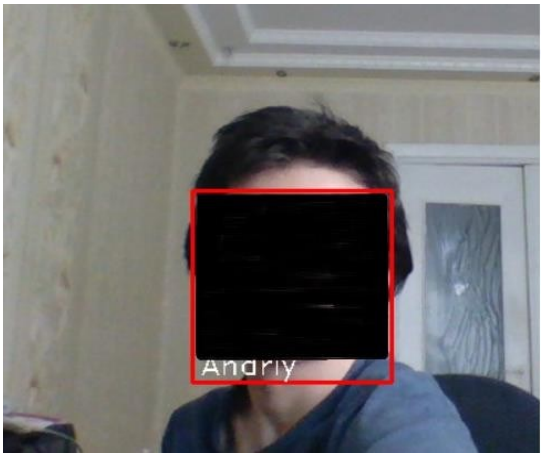
а) 20 см



б)40 см



в) 60 см

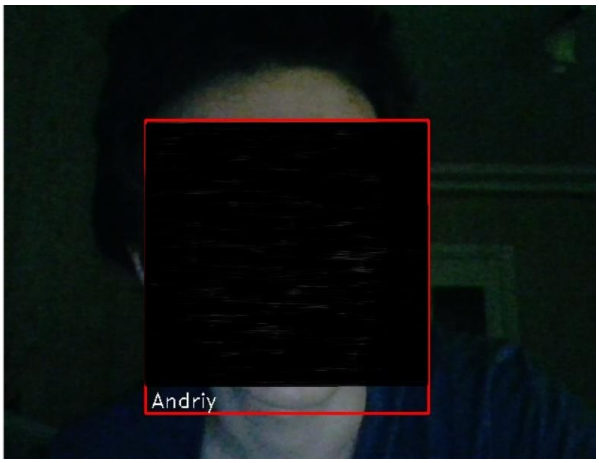


г) 80 см

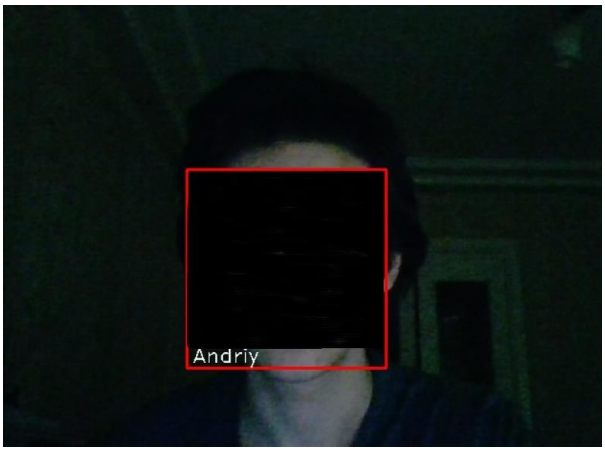
д) 100см

Рисунок 3.10 – Ідентифікація користувача при достатній освітленості

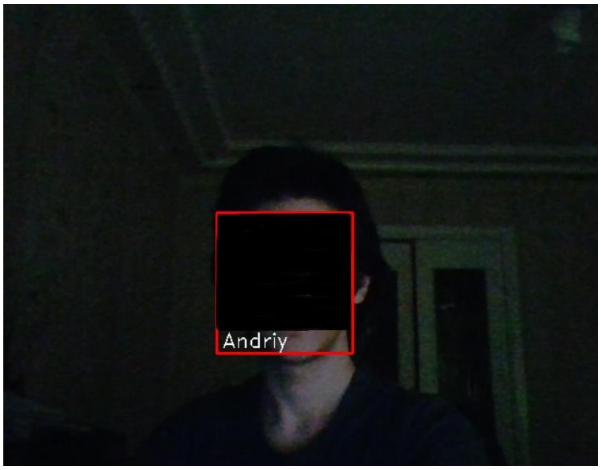
Результат дослідження функціонування програми в умовах недостатньої освітленості наведено на рисунку 3.11.



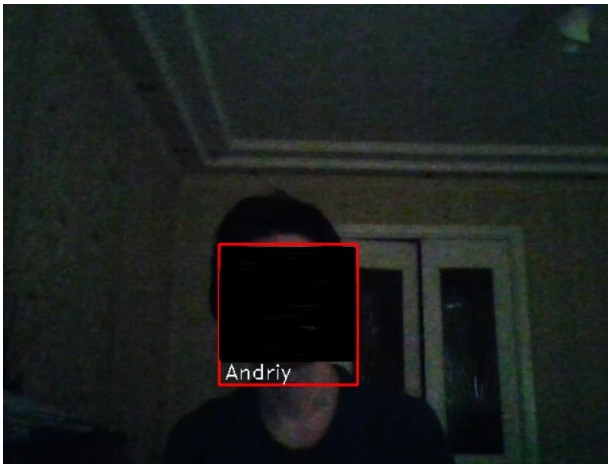
а) 20 см



б) 40 см



в) 60 см

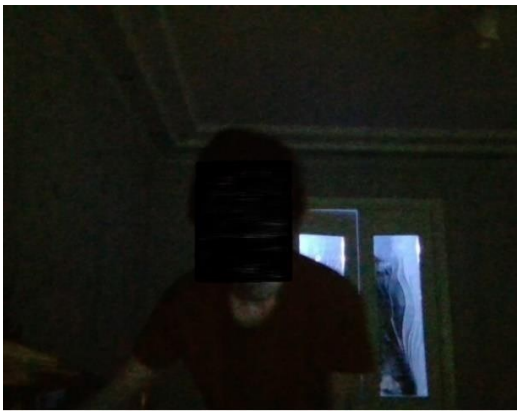


г) 80 см

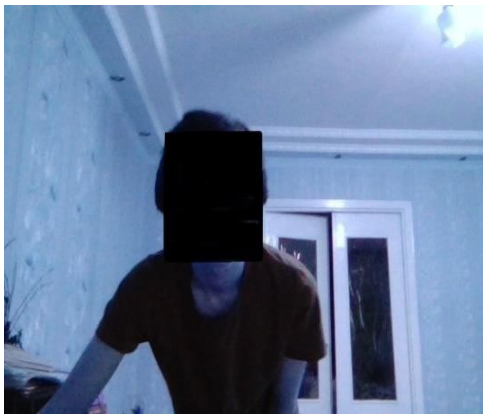
д) 100 см

Рисунок 3.11 – Ідентифікація користувача в умовах недостатньої освітленості

Результат виявлення НСД в умовах достатньої та недостатньої освітленості, наведено на рисунку 3.12.



а) погана природня освітленість



б) хороша природня освітленість

в) штучна освітленість

Рисунок 3.12 – Фіксація НСД у випадку різної освітленості

На підставі результатів апробації ПЗ блокування НСД до ІТС, яке використовує метод біометричної ідентифікації користувачів на основі локальних бінарних шаблонів, встановлено, що програма успішно виявляє обличчя користувачів на відстані до 80 сантиметрів у будь-яких умовах освітлення.

Експеримент показав, що оптимальна відстань для найточнішого виявлення облич користувачів складає від 20 до 60 сантиметрів. На цих відстанях програма демонструє мінімальну кількість помилок і надійно ідентифікує особу за її біометричними даними.

Однак, при відстані більше 80 сантиметрів виявлення обличчя стає менш ефективним, і програма може не здати успішно визначити користувача. Це пов'язано з обмеженнями алгоритму локальних бінарних шаблонів, який вимагає достатньої близкості об'єкта сканування для точного визначення біометричних рис.

Отримані результати є важливими для подальшого вдосконалення системи, зокрема у розробці алгоритмів, які забезпечують надійну роботу на більших відстанях, а також у врахуванні різних умов освітлення для покращення загальної ефективності системи ідентифікації користувачів.

3.4 Висновки до розділу 3

У даному розділі детально описано структуру та порядок роботи розробленого програмного забезпечення (ПЗ) для блокування несанкціонованого доступу (НСД) до інформаційно-технологічних систем (ІТС). ПЗ було розроблено з урахуванням функціональних та технічних вимог, які були сформульовані у першому розділі.

Робота програми включає кілька основних модулів, необхідних для забезпечення її функціональності. Основний модуль відповідає за отримання відеопотоку з веб-камери, яка є вбудованою в ноутбук із роздільною здатністю 1280×720. Цей відеопотік використовується для виявлення облич користувачів.

Програма підтримує можливість створення профілів для різних користувачів, які включаються до бази даних ПЗ. Для перевірки працездатності ПЗ було проведено експерименти за умов різної освітленості: хорошої природньої, поганої природньої та штучної. Результати показали, що програма ефективно виявляє обличчя користувачів на відстані до 80 см за будь-яких умов освітлення.

Особливу увагу було приділено точності виявлення облич. Зокрема, на відстанях від 20 до 60 см програма показала мінімальну кількість помилок. За необхідності блокування доступу до ІТС, користувач має можливість ввести ідентифікатор для видалення його даних з бази даних.

Висновки експериментальних досліджень підтверджують, що розроблене ПЗ є надійним і ефективним засобом для виявлення рухомих об'єктів у системі ІТС ОКІ. Воно має зрозумілий і зручний інтерфейс, що спрощує взаємодію з програмою та підвищує її придатність для практичного застосування в умовах відсутності альтернативних технічних рішень.

ВИСНОВОК

У дипломній роботі обгрунтовано важливість розробки програмного забезпечення для протидії несанкціонованому доступу (НСД) до інформаційно-технологічних систем (ІТС) на основі методу локальних бінарних шаблонів. Аналіз показав, що існує серйозна загроза кібербезпеці ІТС через можливість кібератак та інцидентів НСД.

Дослідження підтвердило, що одним з ефективних методів протидії НСД є застосування сучасних алгоритмів ідентифікації користувачів, зокрема методів біометричної ідентифікації за геометрією обличчя. Цей підхід дозволяє надійно ідентифікувати особу на основі унікальних геометричних параметрів її обличчя, таких як розміри і відстані між ключовими точками.

Розроблене програмне забезпечення спрямоване на забезпечення безпеки ІТС шляхом реалізації методу локальних бінарних шаблонів для виявлення та ідентифікації користувачів. Воно включає в себе модулі для відеоспостереження, обробки зображень та порівняння біометричних даних зі збереженими в базі даних.

Цей підхід є актуальним і важливим у контексті сучасних викликів кібербезпеки, де забезпечення захисту особистих даних та обмеження несанкціонованого доступу до конфіденційної інформації відіграють ключову роль у забезпеченні стабільності та безпеки інформаційних систем.

Було проведено аналіз та деталізація математичного апарату процесу ідентифікації користувачів на основі методу локальних бінарних шаблонів, що становить основу для розроблення відповідного програмного забезпечення. Математичний апарат включає в себе обробку зображень, визначення унікальних біометричних рис обличчя та їх подальше порівняння з базовими даними.

На основі розробленого математичного апарату було створено структурно-алгоритмічну схему програмного забезпечення. Реалізація програмного продукту дозволила успішно впровадити функції відеоспостереження, виявлення облич користувачів та їх ідентифікацію. Практично виявлено, що програмне забезпечення ефективно протидіє несанкціонованому доступу до ІТС, надаючи надійний

механізм захисту в умовах відсутності відповідних технічних та програмних засобів.

Отримані результати мають значну практичну цінність для застосування в різних сферах, включаючи приватні та державні організації, компанії та установи. Використання цього програмного забезпечення дозволяє досягати високого рівня кібербезпеки, забезпечуючи захист від потенційних кібератак і несанкціонованого доступу до важливих інформаційних ресурсів.

Подальші наукові дослідження в області подолання несанкціонованого доступу до ІТС будуть спрямовані на удосконалення програмного забезпечення, розробленого на основі методу локальних бінарних шаблонів. Одним із напрямків є оптимізація обчислювальних операцій для підвищення швидкодії програми, що забезпечить більш ефективне виявлення та ідентифікацію користувачів з використанням біометричних рис обличчя.

Далі, дослідження будуть спрямовані на розроблення нових методик забезпечення відповідного рівня кібербезпеки для ІТС. Це включає в себе вдосконалення методів аутентифікації, зокрема через використання біометричних даних, що дозволяє зменшити ймовірність несанкціонованого доступу та підвищити безпеку інформаційних систем.

Отже, ціль дипломної роботи досягнута, всі часткові завдання дослідження успішно виконані. Отримані результати створюють базу для подальших наукових досліджень у сфері кібербезпеки та ідентифікації користувачів на основі біометричних технологій.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про затвердження Загальних вимог до кіберзахисту об’єктів критичної інфраструктури. [Постанова] – Режим доступу: https://zakon.rada.gov.ua/laws/show/518-2019-п#Text (дата звернення: 26.03.2024)

2. С.О. Гнатюк, М.О. Рябий, В.М. Лядовська: Визначення критичної інформаційної інфраструктури та її захисту: аналіз підходів. [Текст]

3. Guidance on the Essential Critical Infrastructure Workforce**.** [Електронний ресурс] – Режим доступу: https://www.cisa.gov/critical-infrastructure- sectors (дата звернення: 28.03.2024)

4. НД ТЗІ 1.1-002-99. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. [Текст]

5. Терміни та визначення понять : ДСТУ 7448:2013. [Текст]

6. Бідюк П., Бондарчук В. Сучасні методи біометричної ідентифікації[Текст]/ Правове, нормативне та метрологічне забезпечення системи захистуінформації в Україні. 2009, № 1 (18). С. 137–146.

7. Єсін В. І., Кузнецов О. О., Сорока Л. С.: Безпека інформаційних систем і технологій: навчальний посібник для студентів вищих навчальнихзакладів. Харків [Текст]/ ХНУ імені В.Н. Каразіна, 2013. 632 с

8. Ю.А Тимошин, С.П. Орленко: Алгоритм розпізнавання обличчя людей на базі згорткової нейронної мережі [Текст]/ Міжвідомчий науково- технічний збірник “Адаптивні системи автоматичного управління” №1’(32) 2018

9. Лисак А. Ідентифікація и аутентифікація особи : огляд основних биометричних методів перевірки справжності користувачакомп’ютерних систем [Текст]/ Математичні структури та моделювання. №2(26), 2012. С.124-134.

10. Левенець Т., Кравець І. Дослідження методів розпізнавання обличь при використанні мобільних технологій [Текст] / Наукові праці Чорноморського державного університету імені Петра Могили комплексу “Києво-Могилянська академія”. Серія : Комп’ютерні технології. 2016, Т. 283, Вип. 271. С. 28–35.

11. Jensen O. Implementing the Viola-Jones face detection algorithm [Текст]/ Informatics and Mathematical Modelling, Denmark : Technical University ofDenmark

2008. 36 p.

12. НД ТЗІ 2.5-005-99. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу.[Текст]

13. Бойченко, О. С., Гуменюк, І. В., Сметанін, К. В., & Некрилов, О. В.: Метод блокування доступу до інформаційно-телекомунікаційних систем на основі біометричної ідентифікації/аутентифікації користувачів.[Текст]/ *Технічна інженерія*, (1(85), 171–176, 2020

14. Рогоза М. Є., Рамазанов С. К., Велігура А. В. Основи інформатики та технологій програмування.[Текст]/ Луганськ : СНУ, 2012. 74 с.

15. Eric Matthes: Python Crash Course(2nd Edition): A Hands-On, Project- Based Introduction to Programming [Текст]/ Eric Matthes. – 2019.

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

_____ ЯРЕМЧУК Ю.Є

“ 02 ” березня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до бакалаврської дипломної роботи на тему:

Розробка програмного модуля захисту від НСД на основі методу локальних бінарних шаблонів

08-72.БДР.014.00.000.ТЗ

Керівник бакалаврської дипломної роботи

д.т.н., професор

_____ ЯРЕМЧУК Ю.Є.

“ ” _____

Вінниця – 2024 р.

1. Найменування та область застосування

Програмний засіб для захисту електронної пошти від фішингових атак з використанням API VirusTotal. Область застосування: захист інформаційних ресурсів та конфіденційних даних користувачів у системах електронної пошти.

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ № 80 від 11.03.2024 р.

3. Мета та призначення розробки

3.1 Мета розробки: розробка ефективного методу захисту від несанкціонованого доступу, забезпечення високого рівня безпеки та конфіденційності даних користувачів.

3.2 Призначення: розроблений програмний засіб виконує автентифікацію користувача, перевіряючи риси обличчя за допомогою веб-камери.

4. Джерела розробки

4.1 С.О. Гнатюк, М.О. Рябий, В.М. Лядовська: Визначення критичної інформаційної інфраструктури та її захисту: аналіз підходів. [Текст]

4.2 Guidance on the Essential Critical Infrastructure Workforce. [Електронний ресурс] – Режим доступу: https://www.cisa.gov/critical-infrastructure- sectors (дата звернення: 28.03.2024)

4.3 НД ТЗІ 1.1-002-99. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. [Текст]

4.4 Eric Matthes: Python Crash Course(2nd Edition): A Hands-On, Project- Based Introduction to Programming [Текст]/ Eric Matthes. – 2019

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний, легкий у використанні інтерфейс користувача;

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Програмний засіб повинен виконувати процес автентифікації користувача за .рисами обличчя

5.2 Вимоги до надійності:

5.2.1 Програмний засіб повинен працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень;

5.2.2 Бази даних повинні бути налаштовані на автоматичне створення резервних копій;

5.2.3 Програмний засіб повинен виконувати свої функції з високою точністю та надійністю.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – Pentium 1500 МГц і подібні до них;

– оперативна пам’ять – не менше 512 Mb;

– середовище функціонування – операційна система сімейство Windows;

– вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп’ютерною технікою.

6. Вимоги до програмної документації

6.1 Обов’язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.4

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист розроблюваного програмного засобу від несанкціонованого використання.

7.2 Неможливість отримання доступу незареєстрованих користувачів до інформаційних ресурсів.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Має бути реалізований таким чином, щоб підходити для використання широкого загалу.

9. Стадії та етапи розробки

з/п	Назва етапів бакалаврської дипломної роботи	Початок	Закінчення
.	Визначення напрямку бакалаврської роботи, формулювання теми	23.03.2024	25.03.2024
.	Аналіз предметної області обраної теми	26.03.2024	30.03.2024
.	Розробка алгоритму роботи	31.03.2024	2.04.2024
.	Написання бакалаврської роботи на основі розробленої теми	3.04.2024	15.05.2024
.	Передзахист бакалаврської дипломної роботи	24.05.2024	24.05.2024
.	Виправлення, уточнення, корегування бакалаврської дипломної роботи	25.05.2024	27.05.2024
.	Захист бакалаврської дипломної роботи	13.06.2024	15.06.2024

10. Порядок контролю та прийому

10.1 До приймання бакалаврської дипломної роботи надається:

- ПЗ до бакалаврської дипломної роботи;
- демонстрація результату бакалаврської дипломної роботи;
- презентація;
- відгук керівника роботи;
- відгук рецензента

Технічне завдання до виконання прийняв _____ Матвієнко Д.В.

ДОДАТОК Б. ЛІСТИНГ ПРОГРАМНОГО КОДУ

```
import tkinter
from tkinter import messagebox
import cv2
import PIL.Image, PIL.ImageTk
from PIL import Image as Ims
from PIL import ImageTk
from PIL import ImageTk as IMTK
from cv2 import *
from tkinter import *
import numpy as np
import pickle
import os
from builtins import max as maximum
from time import sleep
import face_recognition
from tkinter import ttk
import tkinter.filedialog as tkFileDialog
from sys import exit
import warnings
import numpy as np
import time


class App:

   def _init_(self, window, window_title, video_source):
      self.initil()

      self.window = window
      self.window.geometry('1005x590')

      self.window.title(window_title)
      self.window.resizable(width=False,height=False)
      self.user_name_add = StringVar()
      self.user_name_del = StringVar()
      self.program_tab = ttk.Notebook(window)
      self.tab1 = ttk.Frame(self.program_tab)
      self.program_tab.add(self.tab1, text='Програма протидії НСД')
      self.tab2 = ttk.Frame(self.program_tab)
      self.program_tab.add(self.tab2, text='Довідка')
      self.program_tab.pack(expand=1, fill='both')
      self.button1= Button(self.tab1,text="Увімкнути",width=25, height=3, font=("Comic Sans MS",
         8),bg='Skyblue3',
highlightthickness=0, command=self.update).place(x=100, y=500)
      self.button2 = Button(self.tab1, text="Додати нового користувача", width=25, height=3,
font=("Comic Sans MS", 8),bg='Skyblue3', highlightthickness=0,
command=self.add_user_window).place(x=300, y=500)
```

```
        self.button3 = Button(self.tab1, text="Видалити дані користувача", width=25, height=3,
font=("Comic Sans MS", 8),bg='Skyblue3', highlightthickness=0,
command=self.delete_user_window).place(x=500, y=500)

        self.button4 = Button(self.tab1, text="Вихід", width=25, height=3, font=("Comic Sans MS",
            8),bg='Skyblue3', highlightthickness=0, command=self.program_close).place(x=700, y=500)

        self.dovidka = Label(self.tab2, text = '''Дана програма розроблена для протидії
несанкціонованому доступу(НСД).\n
Функції програми:\n
Увімкнути – розпочати процес протидії НСД шляхом сканування обличчя людини, що
використовує ІТС\n
Додати дані нового користувача – в базу даних заносяться прізвище працівника англійською та
характеристики для порівняння\n
Видалити дані користувача – видаляємо дані користувача з бази даних з характеристиками,
тому він більше не буде мати доступу.\n
Вихід – припинення роботи програми''', font=("Comic Sans MS",
10)).place(x=80,y=50)

try:
   with open('datab.pickle', 'rb') as f:
      users = pickle.load(f)
except Exception as ex:
   print(ex)
   users = {}
   if os.path.exists('dataset'):
      for root, dirs, files in os.walk('dataset', topdown=False):
         for name in files:
            os.remove(os.path.join(root, name))
            self.video_source = video_source
            self.known_face_names = [x[0] for x in users.values()]

self.known_face_encodings = [x[1] for x in users.values()]
self.face_locations = []
self.face_encodings = []
self.face_names = []

self.process_this_frame = True
self.video_capture = cv2.VideoCapture(self.video_source)

if not self.video_capture.isOpened():
   raise ValueError("Unable to open video source", self.video_source)
self.width = self.video_capture.get(cv2.CAP_PROP_FRAME_WIDTH)
self.height = self.video_capture.get(cv2.CAP_PROP_FRAME_HEIGHT)
self.face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

self.canvas = tkinter.Canvas(self.tab1, width = self.width, height = self.height)
self.canvas.pack(side=TOP)
```

```
self.window.mainloop()

def add_user_window(self):
   self.a1 = Toplevel()
   self.a1.title('Додати користувача')
   self.a1.geometry("324x115")


self.a1['bg'] = 'Skyblue3'
self.a1.resizable(width=False, height=False)
self.lbll = Label(self.a1, text="В дане поле введіть прізвище англійською
мовою", font=("Comic Sans MS", 10)).place(x=0,y=0)
self.name_face = Entry(self.a1, textvariable=self.user_name_add,font=("Comic Sans MS",
      10), bg="silver"
).place(x=85,y=30)
self.btn4 = Button(self.a1, text="Додати користувача", command=self.add_new_user ,
font=("Comic Sans MS", 10)).place(x=97,y=55)

def delete_user_window(self):
   self.a2 = Toplevel()
   self.a2.title('Видалити користувача')
   self.a2.geometry("324x115")
   self.a2['bg'] = 'Skyblue3'
   self.a2.resizable(width=False, height=False)
   self.lbll = Label(self.a2, text="В дане поле введіть прізвище
англійською мовою", font=("Comic Sans MS", 10)).place(x=0,y=0)
   self.name_face = Entry(self.a2, textvariable=self.user_name_del,font=("Comic Sans
      MS", 10), bg="silver"
).place(x=85,y=30)
   self.btn4 = Button(self.a2, text="Видалити користувача",
command=self.delete_user , font=("Comic Sans MS", 10)).place(x=97,y=55)

def update(self):
   if ret:
      self.photo= PIL.ImageTk.PhotoImage(image = PIL.Image.fromarray(self.frame))
      self.canvas.create_image(0, 0, image = self.photo, anchor = tkinter.NW)
      self.delay = 30
      self.window.after(self.delay, self.update)


def program_close(self):
   exit()

def delete_user(self):
   with open('datab.pickle', 'rb+') as f:
      name_del = self.user_name_del.get()
      if not self.user_name_del.get():
         messagebox.showerror('', 'Ви не ввели і\'мя користувача')
```

```
        return None
        L = pickle.load(f)
        user_list = {}

#for k in L.copy():
 #   if name_del not in L[k]: messagebox.showinfo('','Такого користувача немає!\nСпробуйте ще раз') break

#with warnings.catch_warnings() as f:
 #   warnings.simplefilter(action='ignore',
                                    #category=FutureWarning) for k in L.copy():
  #  if name_del not in L[k]: user_list[k] = L[k]

#f.seek(0) f.truncate()
#pickle.dump(user_list, f) f.close()

def add_new_user(self):

   with open('datab.pickle', 'rb') as f:
      users = pickle.load(f)
      path_for_photo = os.path.join(os.getcwd(), 'dataset')
      cam = cv2.VideoCapture(CAP_ANY)
      cam.set(3, 640)
      cam.set(4, 480)
      face_detector      = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
      if not self.user_name_add.get():
         messagebox.showerror('', 'Ви не ввели і\'мя користувача')
         return None
         messagebox.showinfo('', 'Станьте перед камерою та не рухайтесь!')
         ides = users.keys()
      if len(ides) == 0:
         face_id = 1
      else:
            z=[int(x[0]) for x in ides]
            face_id = maximum([int(x[0]) for x in ides]) + 1
            count = 0
   while True:

      ret, img = cam.read()
      gray = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
      faces = face_detector.detectMultiScale(gray, 1.3, 5)
      for (x, y, w, h) in faces:
         cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
         count += 1
         cv2.imwrite(os.path.join(path_for_photo, "User." + str(face_id) + '.' + str(count) + ".jpg"),gray[y:y + h, x:x + w])
         image = face_recognition.load_image_file( os.path.join(path_for_photo,    "User."+
          str(face_id)     +          '.'          + str(count) + ".jpg"))
```

```
            face_encoding = face_recognition.face_encodings(image)[0]
            users[str(face_id)] = [self.user_name_add.get(), face_encoding]
            cv2.imshow('image', img)
            with open('datab.pickle', 'wb') as f:
                pickle.dump(users, f)
                self.user_name_add.set('')
                k = cv2.waitKey(100) & 0xff
                if k == 27:
                    break
                elif count >= 1:
                    break
            cam.release()
            cv2.destroyAllWindows()
            messagebox.showinfo('', 'Ваші дані було додано, перезапустіть програму!')

    def get_frame(self):
        if self.video_capture.isOpened():
            ret, frame = self.video_capture.read()
            small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
            rgb_small_frame = small_frame[:, :, ::-1]
            name = "Unknown"
        if self.process_this_frame:
            self.face_locations = face_recognition.face_locations(rgb_small_frame)
            self.face_encodings = face_recognition.face_encodings(rgb_small_frame,
    self.face_locations)
            self.face_names = []
        for self.face_encoding in self.face_encodings:
            matches = face_recognition.compare_faces(self.known_face_encodings, self.face_encoding)
            face_distances = face_recognition.face_distance(self.known_face_encodings,
    self.face_encoding)
            best_match_index = np.argmin(face_distances)
        if matches[best_match_index]:
            name = self.known_face_names[best_match_index]
            self.face_names.append(name)
        if name != 'Unknown':
            face_distances = face_recognition.face_distance(self.known_face_encodings,
    self.face_encoding)
            best_match_index = np.argmin(face_distances)
        if matches[best_match_index]:
            name = self.known_face_names[best_match_index]
            self.face_names.append(name)

            print(name)
        else:
            cv2.imwrite("NSD/1 - " + time.strftime("%d-%m-%Y-%H-%M-%S") + ".jpg",

            cv2.cvtColor(frame, cv2.COLOR_RGB2BGR))
            os.system("off.bat")
```

```
        self.process_this_frame = not self.process_this_frame
    for (top, right, bottom, left), name in zip(self.face_locations, self.face_names):
        top *= 4
        right *= 4
        bottom *= 4
        left *= 4
        cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2), 255, 1

        font = cv2.FONT_HERSHEY_DUPLEX
        cv2.putText(frame, name, (left + 6, bottom - 6), font, 0.7, 255, 255) # Return a boolean success
flag and the current frame converted to
    if ret:
        return (ret, cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))

    else:
       return (ret, None)

def initil(self):
    os.getcwd()
    if not os.path.exists('dataset'):
        os.mkdir('dataset')

    if not os.path.exists('NSD'):
        os.mkdir('NSD')
        path_for_photos = os.path.join(os.getcwd(), 'NSD')

  if    name   == "    main    ":

        My_app = App(tkinter.Tk(), "Program", video_source=0)
```

ДОДАТОК В. ІЛЮСТРАТИВНИЙ МАТЕРІАЛ (ПРЕЗЕНТАЦІЯ)

Вінницький національний технічний університет

РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ЗАХИСТУ ВІД НСД НА ОСНОВІ МЕТОДУ ЛОКАЛЬНИХ БІНАРНИХ ШАБЛОНІВ

Виконав: Матвієнко Данило
1КІТС-20б
Керівник: Яремчук Ю.Є.
Вінниця 2024

Актуальність теми

- Зростання випадків НСД
- Потреба в надійному захисті інформаційно-комунікаційних систем
- Важливість в надійному захисті користувачів

Мета та завдання дослідження

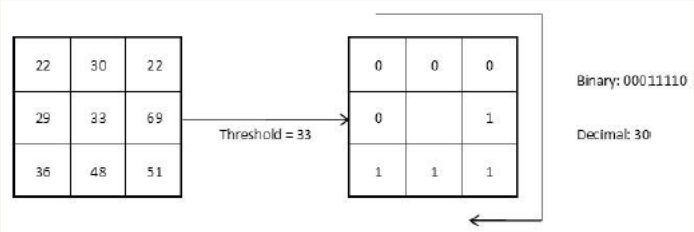
- Розробка програмного модуля захисту
- Запобігання порушення цілісності, конфіденційності та доступності
- Протидія та запобігання НСД

Аналіз існуючих рішень

- Існуючі рішення для протидії НСД
- Переваги та недоліки
- Вибір оптимального підходу

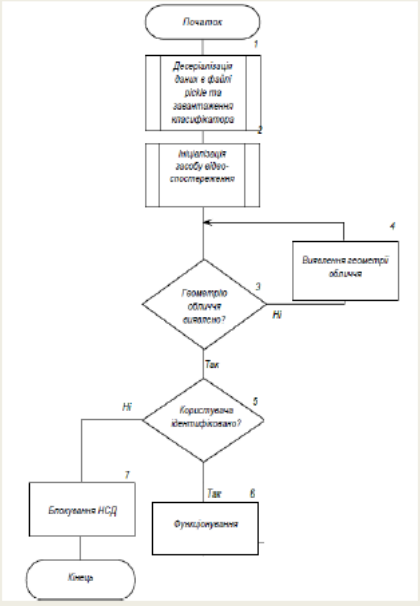
Метод локальних бінарних шаблонів

- Принцип дії
- Переваги та недоліки методу
- Аргументація вибору



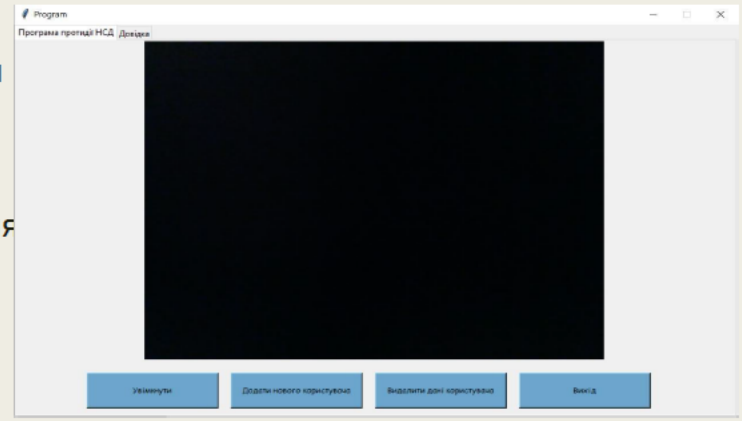
Реалізація захисного модуля

- Алгоритм модуля
- Інтеграція з MySQL
- Захист від НСД



Інтерфейс користувача

- Зручність використання
- Універсальність та мультиплатформність
- Додавання та видалення користувачів



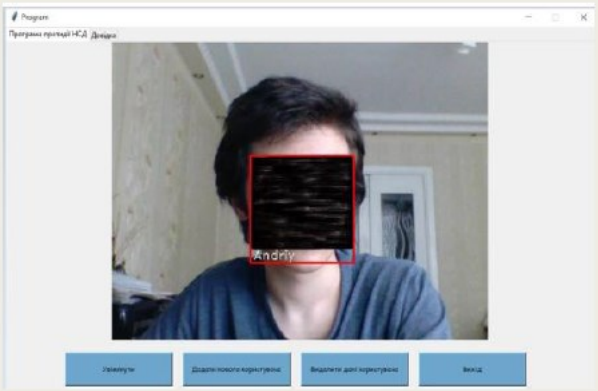
Результати тестування

- Виявлення обличчя на різних відстанях та при різному освітленні
- Швидкість роботи програмного модуля
- Точність роботи програмного модуля

Відстань, см	Вдалі ідентифікації	Невдалі ідентифікації	Заблоковано неавторизованих користувачів
20	96	4	3
40	88	12	9
60	82	18	15
80	74	26	22
100	65	35	28

Переваги розробленого модуля захисту

- Ефективність блокування НСД
- Простота інтеграції в ІТС
- Виявлення обличчя при різних умовах



Висновки та майбутні перспективи

- Підсумки роботи
- Можливості для подальшого розвитку
- Важливість постійного вдосконалення захисту

ДОДАТОК Г. ПРОТОКОЛ ПЕРЕВІРКИ НА АНТИПЛАГІАТ

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка програмного модуля захисту від НСД на основі методу локальних бінарних шаблонів

Тип роботи: бакалаврська дипломна робота
(БДР, МКР)

Підрозділ: Кафедра менеджменту та безпеки інформаційних систем
Факультет менеджменту та інформаційної безпеки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 73 % Схожість 27 %

Аналіз звіту подібності (відмітити потрібне):
1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
3. 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.

Особа, відповідальна за перевірку _____ (підпис) Коваль Н.П. (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ (підпис) Матвієнко Д.В. (прізвище, ініціали)

Керівник роботи _____ (підпис) Яремчук Ю.Є. (прізвище, ініціали)