



[DOI 10.28925/2663-4023.2024.25.410433](https://doi.org/10.28925/2663-4023.2024.25.410433)

УДК 004.056

Огонюк Назар Вікторович

студент

Національний університет «Львівська політехніка», Львів, Україна

ORCID ID: 0009-0001-1934-7632

nazar.ohoniuk.itis.2020@lpnu.ua

Назаркевич Марія Андріївна

д.т.н., професор

Національний університет «Львівська політехніка», Львів, Україна

ORCID ID: 0000-0002-6528-9867

mariia.a.nazarkevych@lpnu.ua

Мішковський Юрій Ігорович

аспірант

Національний університет «Львівська політехніка», Львів, Україна

ORCID ID: 0009-0004-0051-026X

yurii.i.myshkovskiy@lpnu.ua

Наконечний Назар Ігорович

аспірант

Національний університет «Львівська політехніка», Львів, Україна

ORCID ID: 0009-0000-2456-3498

Nazar.I.Nakonechnyi@lpnu.ua

Романчук Роман Васильович

аспірант

Національний університет «Львівська політехніка», Львів, Україна

ORCID ID: 0009-0004-4352-1073

roman.v.romanchuk@lpnu.ua

ПОБУДОВА СИСТЕМИ ІДЕНТИФІКАЦІЇ РУХОМИХ ОБ'ЄКТІВ НА ОСНОВІ ТЕХНОЛОГІЙ МАШИННОГО НАВЧАННЯ

Анотація. Дослідження присвячено побудові системи ідентифікації рухомих об'єктів при відеопотоці на основі технологій машинного навчання. Відслідковування та розпізнавання рухомих об'єктів є актуальною задачею сьогодення. Важливо є розпізнавання об'єктів в русі та провести їх ідентифікацію на основі штучного інтелекту. Система розділена на три основні модулі: розпізнавання облич, відстеження людей і збереження результатів розпізнавання. Описано використання сучасних технологій і алгоритми машинного навчання YOLOv7 для відстеження людей, та бібліотеку Face Recognition для розпізнавання обличч. Створено контекстну діаграму Data flow diagram, де показано послідовність кроків, необхідних для перетворення вхідного відеопотоку у нормалізовані зображення облич, які готові для подальшого розпізнавання. Побудовано ієрархію процесів системи ідентифікації рухомих об'єктів. Діаграма декомпозиції процесу обробки відео показує логічну послідовність етапів та потоків даних, необхідних для підготовки зображень облич. При класифікації поведінки пов'язуються виявлені патерни руху з конкретними типами поведінки. Система використовує ідентифікаційні дані осіб та інформацію про їх попередню поведінку для класифікації патернів руху. Декомпозиція процесів дозволила детально розглянути кожен з ключових аспектів системи та розкрити послідовність кроків та потоків даних, необхідних для їх реалізації. Побудова діаграми ієрархії процесів дала можливість якісно відобразити взаємозв'язки між усіма процесами та підпроцесами системи, демонструючи логічну послідовність їх виконання. ER-діаграма визначила структуру бази даних, що була використана для зберігання інформації про осіб. Системний аналіз заклав основу для подальшого проектування та розробки інформаційної системи відслідковування та розпізнавання людей. Він дозволив визначити основні функціональні вимоги, структуру



системи та взаємозв'язки між її компонентами. Особливо важливим є можливість застосування програмного забезпечення в цілях попередження загроз тероризму та диверсії. Завдяки подібним інформаційним системам стає можливим підвищення економічного становища як і окремих об'єктів, так і країни в цілому.

Ключові слова: відеопотік; інформаційна система; розпізнавання; ідентифікація.

ВСТУП

Наявні системи розпізнавання та відслідковування людей зазвичай вимагають значних фінансових інвестицій для їх інтеграції, або значних технічних знань. Більше того, більшість наявних сервісів, що вирішують подібні проблеми, пропонують збереження даних у власних «хмарах», що ставить під велике питання безпеку персональних даних. Розроблена система відслідковування та розпізнавання осіб вирішує такого роду проблеми.

Сфера штучного інтелекту активно розвивається і відкриває нові перспективи для інтеграції подібних технологій в повсякденне життя із метою спрощення та підвищення ефективності процесів. Однією із проблем, що вирішує штучний інтелект, є машинне бачення. Важлива галузь машинного бачення — це розпізнавання та відслідковування людей. Ця технологія широко впроваджується на підприємствах з метою підвищення рівня безпеки та ефективності роботи працівників. Найпростішим прикладом є робота охоронця в магазині, виконання обов'язків якого можна суттєво спростити завдяки інтеграції подібних технологій.

В українських реаліях використання подібного програмного забезпечення є особливо актуальним, оскільки підвищена загроза тероризму та диверсій потребує для їх попередження використання значних ресурсів спеціальних служб та державних організацій, які можна було б використати в інших напрямках для забезпечення незалежності країни. Інтеграція систем розпізнавання на підприємствах, що вже мають у своєму розпорядженні засоби для відеоспостереження допоможе закрити як потреби приватних підприємств, так і країни, влада якої за потреби зможе використовувати їх у цілях забезпечення безпеки громадян.

Таким чином, розробка та імплементація ефективної системи розпізнавання та відслідковування людей має надзвичайно велике значення для забезпечення національної безпеки, зміцнення правопорядку та підвищення загального рівня захищеності громадян України. В той же час, змога використовувати подібні технології на потреби бізнесу із метою підвищення ефективності процесів дозволить посилити економічну міць країни, завдяки більш якісному використанню ресурсів. Отже, дослідження у сфері відстеження та розпізнавання людей є вкрай актуальними в сучасних умовах.

Постановка проблеми. Дослідження присвячено побудові системи ідентифікації рухомих об'єктів при відеопотоці на основі технологій машинного навчання.

Аналіз останніх досліджень і публікацій. Ідентифікації осіб у відеопотоках даних розглянуто у [1]. Там представлено мультимодальний підхід для ідентифікації особи на основі відео в неконтрольованому середовищі, який підкреслює тривимірну форму тіла, позу, і зовнішній вигляд. Представлено два кодери: PSE кодує форму тіла за допомогою двійкових силуетів, рухів скелета та тривимірної форми тіла, а AAE забезпечує два рівні агрегації ознак зовнішнього вигляду: агрегацію ознак на основі уваги та усереднення



агрегація. Для агрегування ознак на основі уваги ми використано просторову та часову увагу, щоб зосередитися на ключових областях для розрізнення особи.

У [2] представляємо SEAS, використовуючи Shape-Aligned Supervision, щоб покращити функцію зовнішнього вигляду для повторної ідентифікації людини. Ми використовуємо неявне представлення форми тіла, щоб контролювати навчання кодера ідентифікації на основі зовнішнього вигляду, із екстрактором форм, щоб перевести карту функцій у форми тіла на рівні пікселів, забезпечуючи піксельний рівень керівництво формою. Для повторної ідентифікації особи за допомогою відео включено часову узгодженість у появі різних кадрів в одному відео шляхом додавання калібрування на рівні пікселів. Такий підхід забезпечує найсучаснішу продуктивність у оцінках повторної ідентифікації особи як на основі кадрів, так і на основі відео на загальнодоступних наборах даних. Процес детекції сильно залежить від сторонніх факторів, серед яких — технічні можливості камери, та стабільність її положення, яка є ключовою в процесі детекції руху. До них належить стабільне без мерехтіння світло та висока частота кадрів і показники роздільної здатності камери.

Повторна ідентифікація особи (ReID) спрямована на те, щоб знайти особу, яка цікавить, на кількох камерах, які не перекриваються. Це вважається важливим кроком для програми відстеження осіб. ReID особи можна дослідити на основі зображень або відео. Різні методи глибокого навчання були використані для відео ReID. Підходи глибокого вивчення ознак класифікуються на просторові і часовий підходи. Підходи до глибокого навчання розрізняють на: контрольований, неконтрольоване, слабоконтрольоване та одномоментне навчання. Проводиться детальний аналіз для архітектур найсучасніших підходів глибокого навчання. Їх виконання показано на чотирьох контрольних наборах даних.

В [3] досліджується повторна ідентифікація особи (ReID). на основі зображень або відео. Підходи глибокого вивчення ознак класифікуються на просторові і часові, тоді як глибоке метричне навчання поділяється на метричні підходи до навчання.

Повторна ідентифікація [4] (повторна ідентифікація) є складним завданням через порушення оклюзії. Хоча були докладені великі зусилля для запобігання фіксувати інформацію з одного зображення, нехтуючи багатою додатковою інформацією, доступною в кількох зображеннях, що зображують того самого пішохода. Під час цього процесу, щоб уникнути введення шуму оклюзії, пропонується розроблення модуль локалізації, який вибірково інтегрує інформацію, що сприяє до ідентифікації. Крім того, враховуючи різницю в дискримінаційній природі різних зображень, розроблено модуль кількісної оцінки з урахуванням ймовірностей, щоб чітко інтегрувати високонадійну інформацію.

Необхідність у системі Person ReID [5] для швидко розвиваються міських додатків відеоспостереження є серйозною проблемою за змінами домену. Повторна ідентифікація особи на основі відео (Re-ID) спрямована на отримання відеопослідовностей однієї особи під камерами, які не перекриваються. Попередні методи зазвичай орієнтуються на обмежені погляди, наприклад просторові, часові або просторово-часові вид, якому бракує спостережень у різних ознаках домену. У роботі запропоновано три трансформатори самопогляду для використання взаємозв'язків між локальними особливостями для покращення інформації у просторовій, часовій та просторово-часовій областях.

Метою статті є розробка інформаційної системи відслідковування та розпізнавання людей. Для успішної реалізації поставленої мети були поставлені такі завдання:



- провести аналіз предметної області дослідження, зокрема розкрити теоретичні засади розробки інформаційної системи розпізнавання та відслідковування людей;
- визначити, які існують аналоги інформаційної системи розпізнавання та відслідковування людей, проаналізувати їх особливості, переваги та недоліки;
- застосувати системний аналіз для створення дерева цілей, проєктування структури функцій майбутньої системи та сховища даних;
- вибрати оптимальний програмний засіб для розробки системи з усіх наявних варіантів;
- розробити інформаційну систему для відслідковування та розпізнавання людей, протестувати її функціонал та продемонструвати результати тестування.

Об'єкт дослідження. є процес відслідковування та розпізнавання людей.

Предмет дослідження. є засоби і методи, які використовуються для створення інформаційної системи відслідковування та розпізнавання людей.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Відеопотік, що надходить від камер спостереження або інших джерел, спочатку проходить етап декодування. Це дозволяє перетворити стиснений відеопотік у послідовність окремих кадрів, які можуть бути оброблені індивідуально. Далі, виділені кадри піддаються виявленню осіб, де алгоритми комп'ютерного зору ідентифікують області зображення, що містять обличчя людей [6].

Для підвищення точності розпізнавання обличчя, виявлені обличчя проходять етап вирівнювання. На цьому етапі корегується положення та масштаб обличчя, щоб забезпечити їх однакове розташування та розмір на зображенні. Це важливо, оскільки алгоритми розпізнавання обличчя зазвичай розроблені для роботи з вирівняними зображеннями. Контекстну діаграму обробки відео представлено на рис. 1.

Нарешті, вирівняні зображення обличчя піддаються нормалізації [7]. Цей етап уніфікує зображення за розміром, яскравістю та контрастністю, зменшуючи вплив зовнішніх факторів, таких як освітлення та ракурс зйомки. Нормалізовані зображення обличчя готові для подальшого розпізнавання, де система буде порівнювати їх з базою даних відомих осіб для ідентифікації.

У результаті декомпозиції ми отримали діаграму процесу Обробка відео, яку зображено на рис. 2. Діаграма декомпозиції цього процесу демонструє послідовність кроків, необхідних для перетворення вхідного відеопотоку у нормалізовані зображення обличчя, готові для подальшого розпізнавання. Діаграма декомпозиції процесу «Обробка відео» демонструє логічну послідовність етапів та потоків даних, необхідних для підготовки зображень обличчя до розпізнавання.

Діаграма декомпозиції процесу розпізнавання осіб (рис. 3) дозволяє системі ідентифікувати осіб на зображеннях.

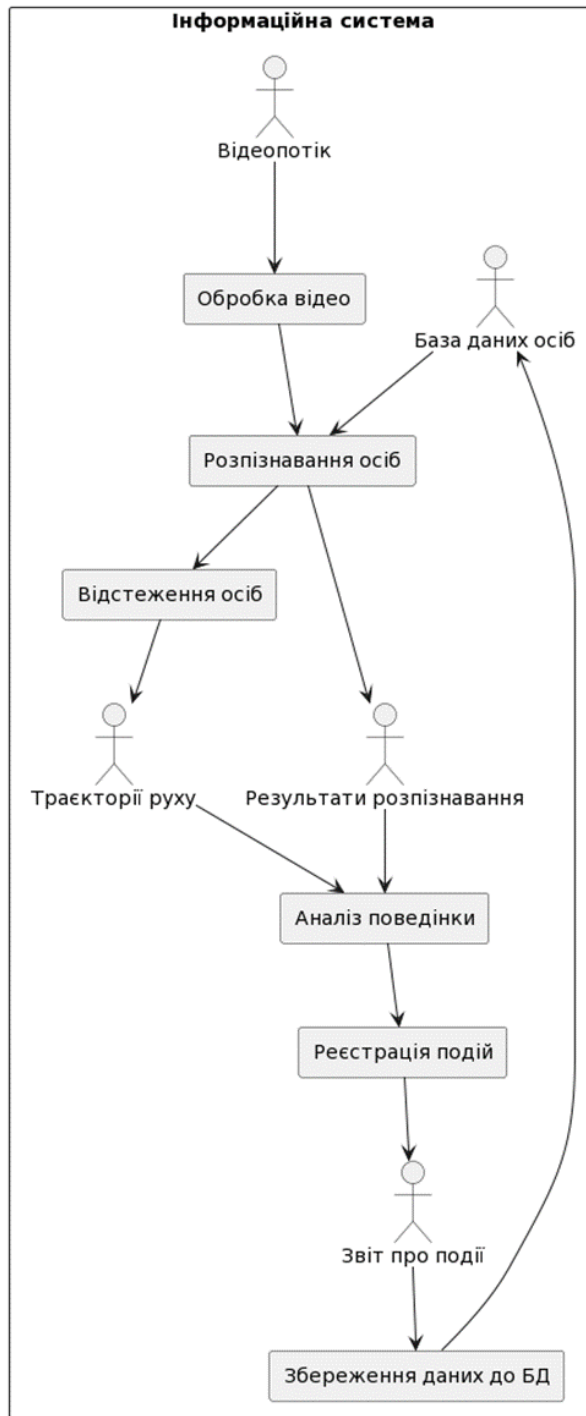


Рис. 1. Контекстна діаграма

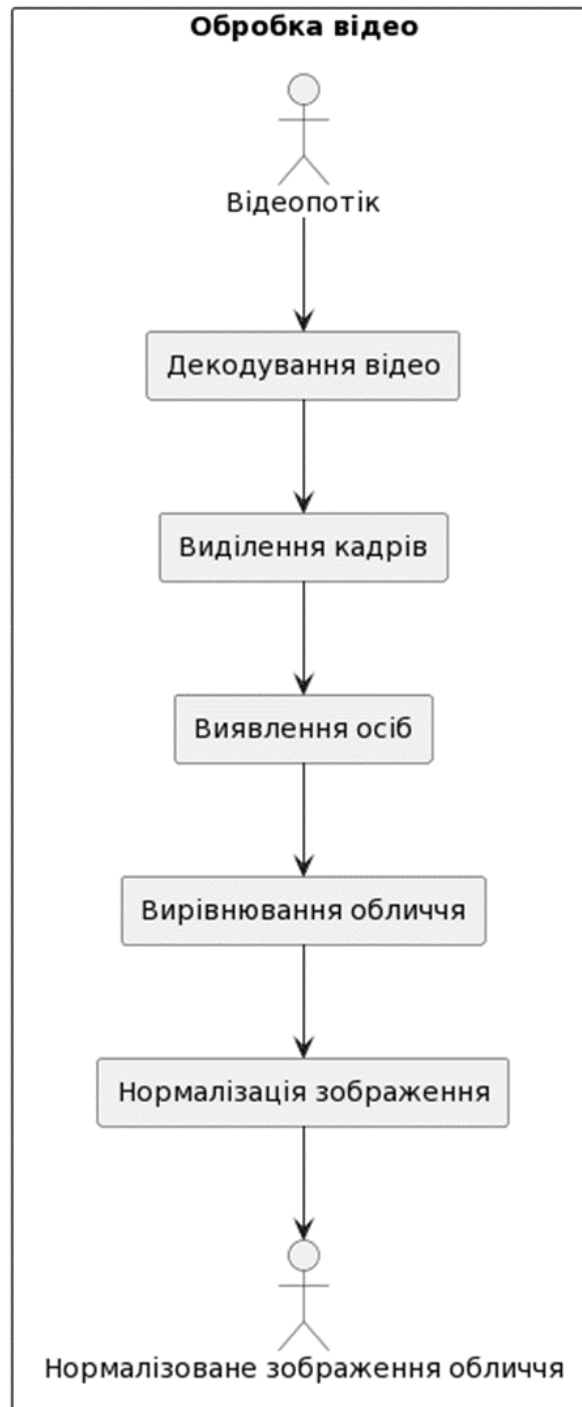


Рис. 2. Діаграма декомпозиції обробки відео

Після того, як нормалізовані зображення обличч отримані на етапі обробки відео, вони надходять на етап виділення ознак [8]. Тут алгоритми комп'ютерного зору аналізують зображення та визначають характерні риси обличчя, такі як відстань між очима, форма носа, лінія рота та інші унікальні характеристики. Ці ознаки перетворюються у математичні вектори, що представляють собою цифровий відбиток обличчя.

Наступний етап — порівняння з базою даних — є серцем процесу розпізнавання. Виділені ознаки обличчя порівнюються з ознаками облич, що зберігаються в базі даних відомих осіб. Система шукає найближчі збіги, використовуючи різні алгоритми порівняння. Результати порівняння надходять на етап класифікації. Тут система приймає рішення про те, чи належить обличчя до особи з бази даних, та визначає рівень достовірності цього рішення. Якщо рівень достовірності перевищує заданий поріг, система вважає особу ідентифікованою та присвоює їй відповідний ідентифікатор.

Діаграма декомпозиції процесу розпізнавання осіб демонструє, як система використовує складні алгоритми для перетворення зображень облич у цифрові відбитки та порівнює їх з базою даних для ідентифікації. Цей процес є ключовим для забезпечення точності та надійності роботи Інформаційної системи відслідковування та розпізнавання людей.

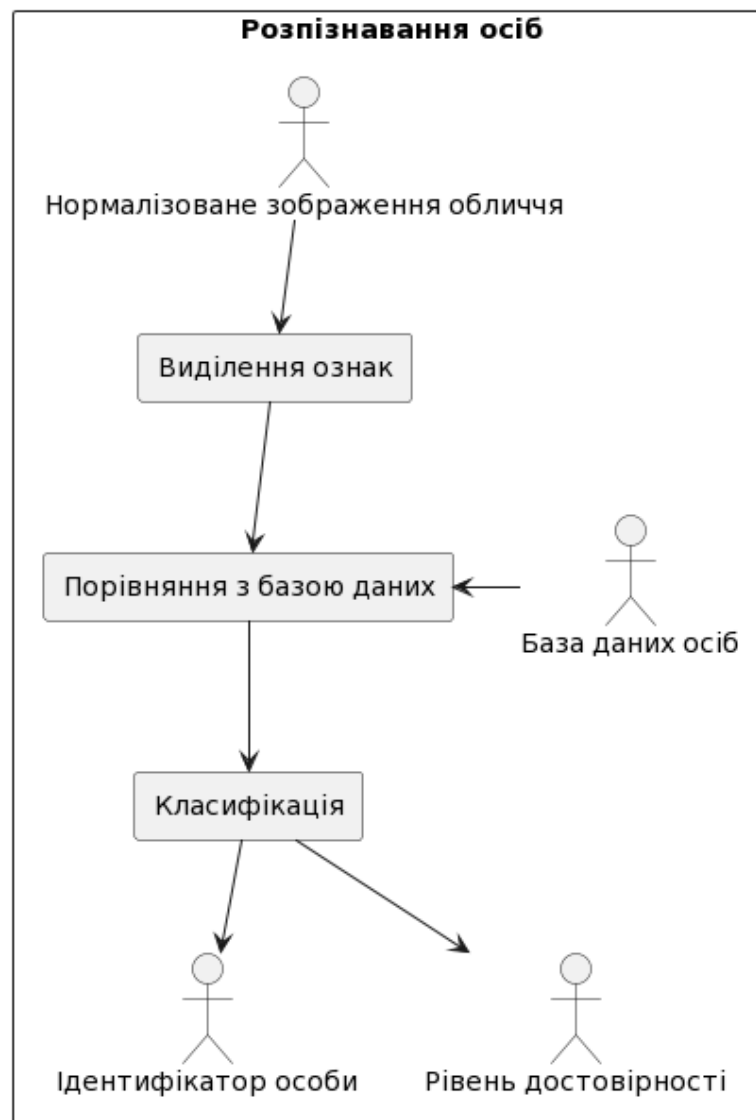


Рис. 3. Діаграма декомпозиції розпізнавання осіб

Діаграма декомпозиції процесу відстеження осіб (рис. 4) розкриває послідовність кроків, що дозволяють системі точно та надійно відстежувати траєкторії руху осіб.

Першим кроком у процесі відстеження є виявлення осіб на кожному кадрі відеопотоку. Цей етап дозволяє системі відстежувати переміщення осіб у просторі та часі.

Далі, система виконує асоціацію виявлених осіб з ідентифікаторами, отриманими на етапі розпізнавання. Це дозволяє пов'язати візуальну інформацію про особу з її ідентифікаційними даними, що зберігаються в базі даних.

Після асоціації з ідентифікаторами, система переходить до відстеження траєкторій руху осіб. На цьому етапі система аналізує послідовність кадрів та визначає, як особи переміщуються в просторі. Це дозволяє побудувати траєкторії руху, що відображають шлях кожної особи.

Останнім кроком є фільтрація шуму з отриманих траєкторій. Це дозволяє видалити помилкові або неточні дані, що можуть виникнути через шум у відеопотоці або помилки алгоритмів. Фільтрація шуму покращує точність та надійність відстеження, забезпечуючи якісні дані для подальшого аналізу поведінки.

Діаграма декомпозиції процесу відстеження осіб демонструє, як система використовує різні технології для відстеження переміщення осіб та побудови їх траєкторій руху. Цей процес є важливим для забезпечення безперервного моніторингу та аналізу поведінки людей в межах ІС.

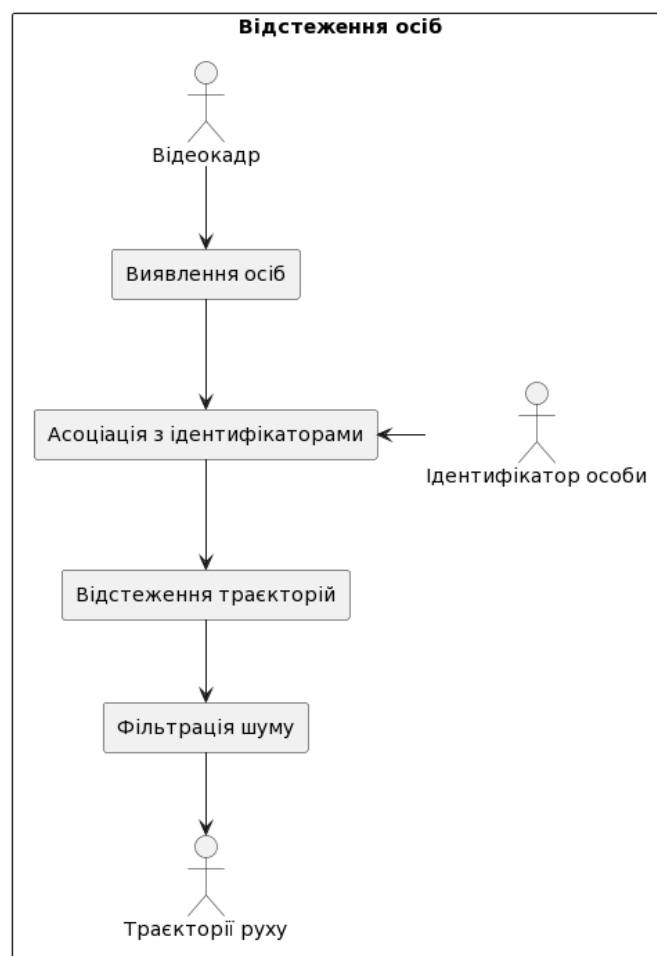


Рис. 4. Діаграма декомпозиції відстеження осіб

Діаграма декомпозиції процесу аналізу поведінки (рис. 5) розкриває логіку аналізу, що базується на траєкторіях руху та ідентифікаційних даних осіб.

Першим кроком у процесі аналізу є виявлення аномалій у траєкторіях руху. Система аналізує дані про переміщення осіб та визначає нетипові або підозрілі патерни, що можуть свідчити про нестандартну поведінку. Це може включати раптові зміни напрямку руху, зупинки в незвичних місцях або нетипову швидкість переміщення.

Далі, система виконує кластеризацію траєкторій руху. Це дозволяє групувати траєкторії за схожими характеристиками, такими як напрямок, швидкість або тривалість. Кластеризація допомагає виявити загальні патерни руху та розділити траєкторії на різні категорії.

Наступний етап — класифікація поведінки — пов'язує виявлені патерни руху з конкретними типами поведінки. Система використовує ідентифікаційні дані осіб та інформацію про їх попередню поведінку для класифікації патернів руху. Це дозволяє присвоїти виявленим патернам такі категорії, як «біг», «ходьба», «стояння» або «підозріла поведінка».

Діаграма декомпозиції процесу «Аналіз поведінки» демонструє, як система використовує інформацію про траєкторії руху та ідентифікаційні дані для виявлення та класифікації патернів поведінки. Цей процес дозволяє зрозуміти, як люди поведуться в межах моніторингової зони, та виявити потенційні загрози або нестандартні ситуації.



Рис. 5. Діаграма декомпозиції аналізу поведінки

Процес реєстрація подій (рис. 6) є завершальним етапом у ланцюжку обробки інформації в Інформаційній системі відслідковування та розпізнавання людей. Діаграма декомпозиції цього процесу розкриває, як система перетворює дані про поведінку людей у структуровані звіти, що відображають ключові події та їх характеристики.

На початку процесу реєстрації відбувається фільтрація подій. Система аналізує патерни поведінки, виявлені на попередньому етапі, та відбирає значущі події, що відповідають заданим критеріям. Це дозволяє відсіяти незначні або надмірні дані та сфокусуватися на подіях, що можуть бути важливими для безпеки або аналізу.

Далі, система виконує кореляцію подій. На цьому етапі окремі події пов'язуються з конкретними особами, використовуючи їх ідентифікаційні дані. Це дозволяє встановити послідовність подій та зрозуміти, як поведінка однієї особи впливає на інших.

Нарешті, система переходить до формування звіту. На цьому етапі інформація про відібрані та скорельовані події об'єднується у структурований звіт, що містить деталі про час, місце, особу та тип поведінки. Звіт може бути представлений у різних форматах, таких як текст, таблиця або графік, залежно від потреб користувачів.

Діаграма декомпозиції процесу «Реєстрація подій» демонструє, як система перетворює розрізнені дані про поведінку людей у цінну інформацію, що відображає ключові події та їх взаємозв'язки. Звіти, сформовані на цьому етапі, можуть бути використані для забезпечення безпеки, аналізу поведінки та прийняття обґрунтованих рішень.

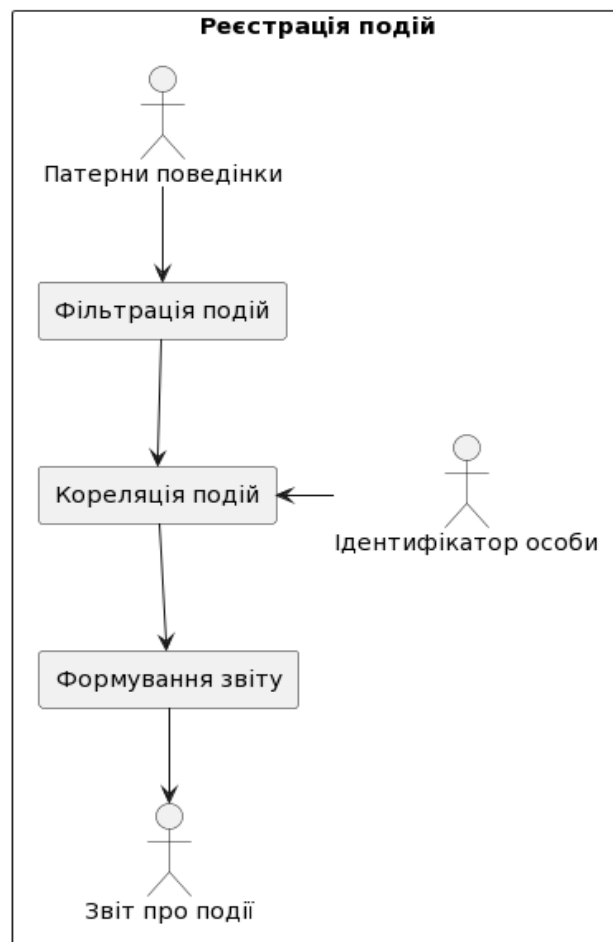


Рис. 6. Діаграма декомпозиції реєстрації подій

Процес збереження даних до БД (рис. 7) забезпечує надійне та ефективне зберігання інформації про події, виявлені Інформаційною системою відслідковування та розпізнавання людей. Діаграма декомпозиції цього процесу розкриває кроки, що гарантують цілісність та якість даних, що зберігаються в базі даних.

Звіт про події, сформований на попередньому етапі, спочатку проходить перевірку цілісності даних. Система аналізує звіт на наявність помилок, невідповідностей або відсутніх даних. Це гарантує, що до бази даних потрапляє лише достовірна та якісна інформація.

Далі, дані звіту проходять трансформацію, тобто перетворення у формат, сумісний з базою даних. Це може включати зміну структури даних, перетворення типів даних або додавання додаткової інформації, необхідної для зберігання в базі даних.

Нарешті, підготовлені дані записуються до бази даних осіб. Система оновлює відповідні записи в базі даних, додаючи інформацію про нові події, пов'язані з конкретними особами. Це забезпечує збереження історії подій та дозволяє використовувати цю інформацію для подальшого аналізу та прийняття рішень.

Діаграма декомпозиції процесу «Збереження даних до БД» демонструє, як система забезпечує цілісність та якість даних, що зберігаються в базі даних. Цей процес є важливим для забезпечення надійності та довгострокової доступності інформації, зібраної ІС.

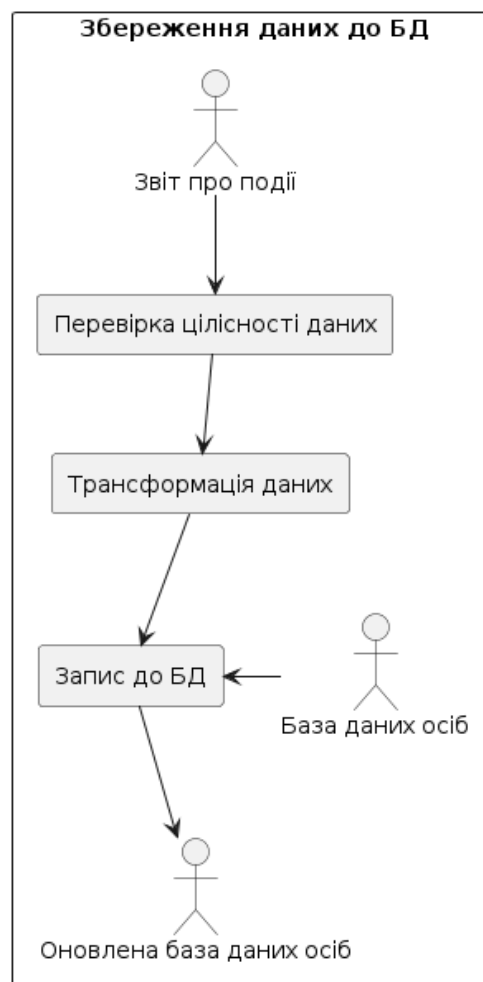


Рис. 7. Діаграма декомпозиції збереження даних до БД

Побудова ієрархії процесів

Діаграма ієрархії процесів демонструє всі задачі та їх підзадачі, необхідні для успішної роботи системи. На даній діаграмі (рис. 8) вершиною є інформаційна система, яка розгалужується на задачі, які, у свою чергу, включають підзадачі, що в них входять. Таким чином створюється ієрархія процесів, де про виконання кожної задачі свідчить виконання усіх вказаних підзадач.

В основні діаграми ієрархії процесів лежать контекстна діаграма та діаграма декомпозиції, де вершиною ієрархії є ІС.

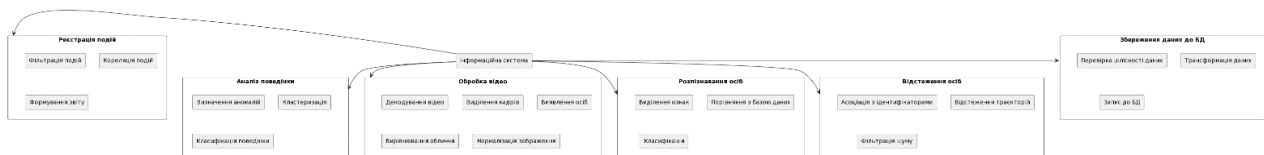


Рис. 8. Ієрархія процесів

Оскільки ІС передбачає наявність бази даних, потрібно провести її проектування на концептуальному рівні. Це означає, що найкращим підходом буде побудова ER-діаграми, яка відобразить структуру майбутньої бази даних інформаційної системи. Діаграма наведена на рис. 9.

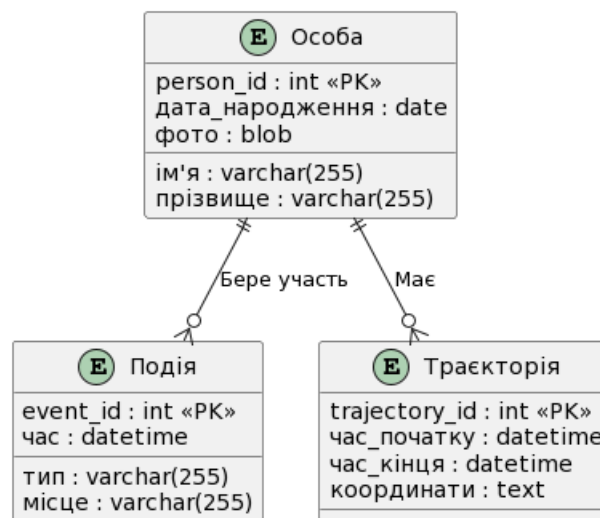


Рис. 9. ER-діаграма

• Сутності:

Особа: Зберігає інформацію про осіб (ідентифікатор, ім'я, прізвище, дату народження, фото).

Подія: Зберігає інформацію про події (ідентифікатор, час, тип події, опис).

Місце: Зберігає інформацію про місця (ідентифікатор, назва, опис).

Зв'язки:

Особа — Подія (Бере участь): Зв'язує осіб з подіями, в яких вони брали участь.

Подія — Місце (Відбувається в): Зв'язує події з місцями, де вони відбулися.

Ця ER-діаграма демонструє базову структуру бази даних для Інформаційної системи відслідковування та розпізнавання людей. Вона може бути розширена додатковими сутностями та зв'язками, залежно від конкретних вимог системи.

МЕТОДИКА ДОСЛІДЖЕННЯ

Python [9] вирізняється величезною екосистемою бібліотек, що покривають різноманітні сфери, включаючи обробку зображень, машинне навчання та комп'ютерний зір. Це дозволяє вам використовувати готові рішення для багатьох задач, що виникають при розробці інформаційної системи. Зокрема бібліотеки OpenCV та Pillow надають інструменти для обробки відеопотоку, виділення кадрів, виявлення облич та виконання інших операцій з зображеннями.

YoloV7 [10], як основа для відстеження людей в нашій інформаційній системі, вражає своїми технічними можливостями та ефективністю. Ця нейронна мережа здатна до надзвичайно точного виявлення об'єктів, включаючи людей, в реальному часі, що є ключовим аспектом для успішної роботи нашої системи.

Точність бібліотеки рівна 99.38%, що було перевірено на відомому набору даних для розпізнавання облич Labeled Faces in the Wild [11], [12]. Нижче продемонстровано здатність розпізнавання лиць при різних значеннях похибки розпізнавання (див. рис. 10).

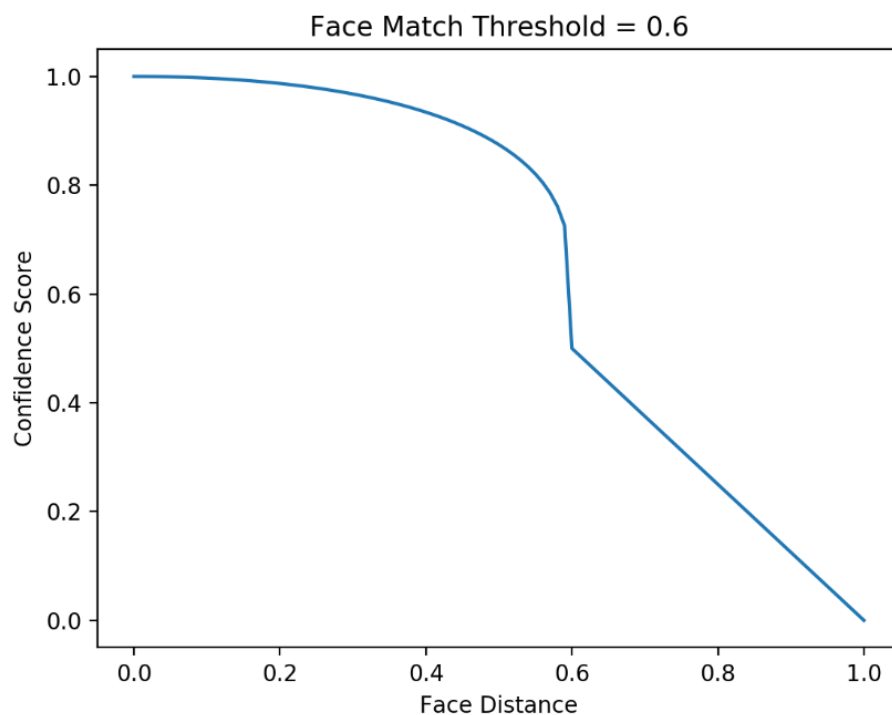


Рис. 10. Точність розпізнавання при допустимій похибці 0.6

При зменшенні значення похибки розпізнавання стане більш строгим, що може дозволити нам налаштувати інформаційну систему більш точно із врахуванням потреб та вимог ПЗ [13], [14].

Ще однією важливою технічною характеристикою Face Recognition є її масштабованість. Бібліотека може бути легко інтегрована в системи з великою кількістю користувачів та великими базами даних облич. Це дозволяє нам використовувати її для розпізнавання великої кількості людей без втрати точності та швидкості роботи.



Розроблення програмного забезпечення відстеження та розпізнавання рухомих об'єктів

Вся система розроблена на мові програмування Python, що є популярним вибором для завдань машинного навчання та комп'ютерного зору. Python має просту та зрозумілу синтаксис, а також велику кількість бібліотек та інструментів, що спрощують розробку та підтримку інформаційних систем.

Система надає можливість вмикати та вимикати функцію відстеження осіб. Це дозволяє зосередитися на розпізнаванні осіб без відстеження їх переміщення, або навпаки, відстежувати переміщення осіб без їх ідентифікації. Така гнучкість дозволяє адаптувати систему до різних сценаріїв використання та забезпечити відповідність вимогам законодавства щодо захисту персональних даних.

Також існує можливість вмикати та вимикати функцію розпізнавання осіб. Це дозволяє використовувати систему лише для відстеження переміщення осіб без їх ідентифікації, або навпаки, зосередитися на розпізнаванні осіб без відстеження їх траєкторій. Така можливість надає користувачам додатковий контроль над процесом обробки даних та дозволяє забезпечити конфіденційність, якщо це необхідно.

Система також пропонує можливість запису обробленого відео. Це дозволяє зберігати відеозаписи з результатами розпізнавання та відстеження осіб для подальшого аналізу або використання як докази.

Система відстеження та розпізнавання людей створена на основі мови програмування Python. Система інтегрує YOLOv7 та Face Recognition для забезпечення точного та швидкого виявлення та ідентифікації осіб у різних умовах та середовищах.

Класи вирішуваних завдань

Основними завданнями ІС є підвищення безпеки та ефективності при використанні систем відеоспостереження. Також програма може бути використана для оптимізації бізнес-логіки та персоналізації досвіду кінцевих споживачів.

Для вирішення даних завдань було обрано написання програми із GUI на основі мови Python з використанням бібліотек YoloV7 та Face Recognition.

YoloV7 використовується для виявлення осіб на зображеннях та відео. YoloV7 здатна виявляти обличчя з високою точністю та швидкістю, що є критично важливим для ефективної роботи системи в режимі реального часу.

Для розпізнавання осіб система використовує технологію Face Recognition, що базується на алгоритмах глибокого навчання. Face Recognition здатна визначати унікальні риси обличчя та порівнювати їх з базою даних відомих осіб, забезпечуючи високу точність ідентифікації.

Опис створеного програмного засобу

Функціонал системи загалом можна розділити на три основні модулі. Перший — це розпізнавання обличчя, другий це відслідковування людей та останній запис і збереження результатів розпізнавання в окремий файл для подальшого аналізу. Усі ці модулі використовуються в нашому GUI інтерфейсі для спрощення взаємодії користувачів із системою. Також варто згадати процес навчання нашої системи на датасеті, що виконується перед компілюванням програми, та використовує різні методи навчання для розпізнавання і відстеження.

Такий підхід дозволяє дотримуватися правила виконання модулем однієї задачі, збільшити читабельність коду та зменшити використання дублювання.

У програмі використана така логіка:

Завантаження закодованих даних обличч з серіалізованого файлу:



with `open('face_encodings.pkl', 'rb')` as `f`: Відкриває файл `face_encodings.pkl` у бінарному режимі читання ('rb'). Конструкція `with` використовується для автоматичного закриття файлу після завершення роботи з ним.

`all_face_encodings = pickle.load(f)`: Використовує модуль `pickle` для генерації даних з файлу. `pickle`. В даному випадку, змінна `all_face_encodings` стає словником, де ключами є імена людей, а значеннями — списки кодувань їхніх облич.

Ініціалізація змінної для відстеження (`tracking`):

`tracking = False`: Ця змінна, імовірно, використовується для управління активацією чи деактивацією відстеження облич. На початковому етапі встановлюється як `False` і означає, що відстеження за замовчуванням вимкнене.

Ініціалізація списків для зберігання кодувань облич та відповідних імен:

`known_face_encodings = []`: Цей список призначений для зберігання кодувань облич, які система може розпізнавати.

`known_face_names = []`: Цей список містить імена, відповідні кодуванням облич у `known_face_encodings`.

Заповнення списків даними з словника `all_face_encodings`:

`for name, encodings in all_face_encodings.items()`: Цей цикл проходить по кожній парі ключ-значення в словнику `all_face_encodings`, де `name` — ім'я особи, а `encodings` — список її обличчя в кодованій формі.

Внутрішній цикл `for encoding in encodings`: Для кожного кодування обличчя у списку `encodings` цей кодування додається до `known_face_encodings`, а відповідне ім'я — до `known_face_names`.

В представленому вище коді відбувається власне процес розпізнавання, який є частиною функції `update_image()`. Він працює за такою логікою:

Перевірка умови `if enable_recognition.get()`: Код виконується, якщо `enable_recognition.get()` повертає `True`. Це перемикач для активації або деактивації функціоналу розпізнавання облич.

Отримання місцезнаходження облич та їх кодувань:

`face_locations = face_recognition.face_locations(rgb_frame)`: Визначає координати облич у кадрі. Ці координати включають верхній, правий, нижній і лівий кути обличчя.

`face_encodings = face_recognition.face_encodings(rgb_frame, face_locations)`: Генерує унікальні кодування облич на основі знайдених розташувань у `rgb_frame`.

Перебір кожного обличчя в кадрі:

`for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings)`: Цикл обробляє кожне обличчя, знаходячи його координати та кодування.

Порівняння виявленого обличчя з відомими кодуваннями:

`matches = face_recognition.compare_faces(known_face_encodings, face_encoding)`: Порівнює кодування виявленого обличчя з вже відомими кодуваннями, повертає список булевих значень (`True/False`).

Ідентифікація обличчя:

Ініціалізується ім'я `name = «Unknown»` на випадок, якщо обличчя знайдено але не розпізнано

`face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)`: Обчислює відстані між виявленим обличчям та відомими кодуваннями.

`best_match_index = np.argmin(face_distances)`: Визначає індекс найменшої відстані (найкраще відповідність).

Якщо знайдена відповідність `if matches[best_match_index]`: присвоює відповідне ім'я обличчю.



Масштабування місцезнаходження облич:

Розміри координат масштабуються в 4 рази, оскільки розпізнавання здійснювалося на кадрі, зменшеному в 4 рази.

Малювання прямокутника та підпису навколо обличчя:

`cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)`: Малює червоний прямокутник навколо обличчя.

`cv2.putText(frame, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.5, (255, 255, 255), 1)`: Додає ім'я під обличчям білим кольором.

Очікування перед обробкою наступного кадру:

`cv2.waitKey(1)`: Ця команда зупиняє виконання коду на короткий час (1 мілісекунду), що дозволяє відображати оброблені кадри в режимі реального часу.

Представлений вище код відповідає за ініціалізацію та завантаження навчених ваг і класів для моделі YOLOv7. Також тут встановлюються глобальні змінні для запису відео.

Ініціалізація моделі YOLOv7:

`yolov7 = YOLOv7()`: Цей рядок створює об'єкт `yolov7`, який представляє модель YOLOv7. YOLO (You Only Look Once) є популярною архітектурою нейронної мережі для задач об'єктного виявлення в реальному часі.

Завантаження ваг і класів для моделі:

`yolov7.load('coco.weights', classes='coco.yaml', device='cpu')`: Цей рядок завантажує переднавчені ваги з файлу `coco.weights` і визначення класів з файлу `coco.yaml`. Ці файли використовуються для ідентифікації різних об'єктів, які модель може виявляти, використовуючи набір даних COCO (Common Objects in Context). Параметр `device='cpu'` вказує, що виконання моделі повинно відбуватися на центральному процесорі (CPU), а не на графічному процесорі (GPU).

Ініціалізація змінних для запису відео:

`video_writer = None`: Ця змінна призначена для збереження об'єкта, який відповідає за запис відео. Початково вона встановлена як `None`, що означає, що запис ще не ініціалізований.

`is_recording = False`: Ця булева змінна вказує, чи активний запис відео. `False` означає, що запис наразі не ведеться.

Представлений вище код відповідає за реалізацію відображення відео в GUI, процес відстеження людей й запис відео. Розглянемо кожен крок алгоритму детальніше:

Відстеження осіб за допомогою моделі YOLOv7:

`if tracking`: Цей умовний блок виконується, якщо змінна `tracking` має значення `True`, що вказує на активацію режиму відстеження.

`detections = yolov7.detect(frame)`: За допомогою моделі YOLOv7 виконується відстеження об'єктів у поточному кадрі відео.

`frame = draw(frame, detections)`: На кадрі відео візуалізуються результати детектування, а саме, рамки навколо виявлених осіб.

`cv2.waitKey(1)`: Затримка на короткий час (1 мілісекунда), щоб GUI застосунка міг адекватно відображати оновлене відео.

Перетворення зображення з формату BGR в RGB і подальше перетворення в формат PIL:

`image = Image.fromarray(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))`: Зображення, яке за замовчуванням у бібліотеці OpenCV (`cv2`) є в форматі BGR, конвертується в RGB, а потім перетворюється у формат бібліотеки PIL для обробки та відображення в графічному інтерфейсі.



image = ImageTk.PhotoImage(image): Перетворення зображення у формат, сумісний з бібліотекою Tkinter для відображення у GUI.

Оновлення графічного інтерфейсу з новим зображенням:

label_image.config(image=image): Оновлення віджету *label_image* з новим зображенням.

label_image.image = image: Збереження посилання на зображення в атрибуті *image* віджета, щоб уникнути його знищення сміттєзбиранням Python.

Запис відео, якщо він активований:

if is_recording: Цей умовний блок виконується, якщо змінна *is_recording* має значення True.

video_writer.write(frame): Запис поточного кадру у файл відео за допомогою об'єкта *video_writer*.

Планування наступного кадру для захоплення:

if running: Якщо змінна *running* має значення True, означає, що процес відеозахоплення активний.

root.after(10, update_image): Виклик функції *update_image* через 10 мілісекунд для обробки наступного кадру. Це забезпечує неперервне захоплення і оновлення відео в реальному часі.

Функція *start_recording()*:

global video_writer, is_recording: Оголошення змінних *video_writer* та *is_recording* як глобальних, щоб мати можливість модифікувати їх у функції.

if not is_recording: Перевірка, чи не активовано запис відео. Якщо *is_recording* має значення False, то запис активується.

fourcc = cv2.VideoWriter_fourcc('XVID')*: Визначення кодека для запису відео. XVID — це популярний кодек, що дозволяє стиснення відео.

video_writer = cv2.VideoWriter('output.avi', fourcc, 10.0, (640, 480)): Створення об'єкта *VideoWriter* для запису відео у файл *output.avi* з частотою 10 кадрів на секунду та розміром кадру 640×480 пікселів.

is_recording = True: Встановлення прапорця *is_recording* в стан True, що означає активацію запису відео.

start_record_btn.config(text='Start Recording (Active)'): Оновлення тексту кнопки запису у GUI, що вказує на активний стан запису.

Функція *stop_recording()*:

Аналогічно використовується глобальне оголошення змінних.

if is_recording: Якщо запис активний, то він зупиняється.

video_writer.release(): Звільнення ресурсів, асоційованих із *video_writer*, закриваючи файл відео.

is_recording = False: Зміна стану запису на неактивний.

start_record_btn.config(text='Start Recording (not recording)'): Оновлення тексту кнопки, для демонстрації, що запис не ведеться.

Функція *toggle_tracking()*:

global tracking: Оголошення *tracking* як глобальної змінної.

Перемикання стану відстеження: якщо *tracking* активно, воно деактивується, і навпаки.

Функція *toggle_recognition()*:

Перемикання стану розпізнавання облич. Якщо розпізнавання активовано (*enable_recognition.get()* повертає True), то воно деактивується, і навпаки.

Функція *close_program()*:



global running: Оголошення *running* як глобальної змінної.

running = False: Встановлення стану програми як неактивного, що призводить до зупинки циклу захоплення відео.

video_capture.release(): Звільнення ресурсів відеозахоплення.

root.quit(): Закриття графічного інтерфейсу користувача.

Також важливим аспектом коректної роботи інформаційної системи є моделі, що були навчені нами перед виконанням програми. У випадку відстеження, навчена модель, що працює правильно, не потребує ніяких змін у майбутньому. Щодо модуля розпізнавання, дані в ньому можуть оновлюватись та доповнюватись новими особами.

З технічної точки зору користувач не повинен перейматись подібними проблемами, тим не менш можлива ситуація в якій кінцевому споживачу необхідно терміново оновити базу даних без участі спеціаліста. Такий випадок також передбачено, тому реалізована можливість створення нових векторів лиць виконанням простого скрипту. Нижче описано логіку роботи цього алгоритму.

Представлений вище код реалізує функціональність для збору шляхів до папок з зображеннями, відсортованими за людьми, та підготовку структури даних для зберігання кодувань облич. Ось детальний опис його логіки:

Функція *get_people_folders(base_folder)*:

Функція приймає як аргумент *base_folder*, що є базовим каталогом, де зберігаються зображення.

people_folders = {}: Ініціалізація порожнього словника для зберігання пар ім'я папки — шлях до папки.

for name in os.listdir(base_folder): Цикл, що ітерує через всі елементи (папки та файли) у заданому *base_folder*. Функція *os.listdir* повертає список імен усіх папок та файлів у вказаній директорії.

folder_path = os.path.join(base_folder, name): Складання повного шляху до кожного елемента, що дозволяє звертатися до файлів і папок відносно базового шляху.

if os.path.isdir(folder_path): Перевірка, чи є елемент директорією (а не файлом).

people_folders[name] = folder_path: Якщо елемент є директорією, додаємо його назву та шлях до словника *people_folders*.

Функція повертає словник *people_folders*, де ключі — це імена папок (людей), а значення — шляхи до цих папок.

Ініціалізація базового каталогу:

base_folder = 'images/': Встановлення шляху до базового каталогу, де зберігаються папки з зображеннями різних людей.

Отримання словника з шляхами:

people_folders = get_people_folders(base_folder): Виклик функції *get_people_folders* із шляхом до базової папки для отримання словника, що містить шляхи до папок кожної особи.

Підготовка словника для кодувань облич:

face_encodings = {}: Ініціалізація порожнього словника, куди будуть зберігатися кодування облич. Цей словник буде використовуватися для асоціації імен людей із їхніми кодуваннями облич.

Код представлений вище виконує обробку зображень осіб, збирає кодування облич із цих зображень, зберігає їх у словнику та серіалізує для подальшого використання.

Обробка кожної папки, пов'язаної з особою:



for person, folder in people_folders.items(): Цей цикл ітерує через кожну пару ключ-значення у словнику `people_folders`, де `person` — це ім'я особи, а `folder` — шлях до папки з її зображеннями.

Ініціалізація списку для кодувань облич особи:

person_encodings = []: Цей список буде використовуватися для зберігання усіх кодувань облич, знайдених у зображеннях конкретної особи.

Ітерація по кожному файлу в папці:

for filename in os.listdir(folder): Цикл проходить по кожному файлу в папці, де зберігаються зображення особи.

if filename.endswith(('png', 'jpg', 'jpeg')): Перевірка, чи файл є зображенням (формати PNG, JPG, JPEG).

Завантаження зображення та детектування кодувань облич:

image_path = os.path.join(folder, filename): Формування повного шляху до зображення.

image = face_recognition.load_image_file(image_path): Завантаження зображення за допомогою бібліотеки `face_recognition`.

encodings = face_recognition.face_encodings(image): Отримання кодувань облич з зображення. Якщо на зображенні виявлено обличчя, функція повертає список кодувань.

Додавання кодування облич до списку:

if encodings: Перевірка, чи були знайдені кодування на зображенні.

person_encodings.append(encodings[0]): Додавання першого кодування облич зі списку до `person_encodings`.

Збереження кодувань облич для особи в словнику:

face_encodings[person] = person_encodings: Асоціювання імені особи з її кодуваннями облич у словнику `face_encodings`.

Серіалізація кодувань облич у файл за допомогою `pickle`:

with open('face_encodings.pkl', 'wb') as f: Відкриття файлу `face_encodings.pkl` у бінарному режимі для запису.

pickle.dump(face_encodings, f): Збереження словника `face_encodings` у файл за допомогою серіалізації `pickle`.

Виведення результатів:

for person, encodings in face_encodings.items(): Цикл по словнику для виведення кількості кодувань облич, асоційованих з кожною особою.

print(f"{person} has {len(encodings)} face encodings."): Виведення імені особи та кількості її кодувань облич.

Опис основних характеристик і особливостей системи

Використання даної програми дозволяє значно скоротити час виконання задач та автоматизувати процеси, що раніше виконувались вручну. Програма обробляє дані швидше та ефективніше, ніж людина, що призводить до значного підвищення продуктивності.

Система можна працювати як і в цілодобовому режимі, так і штатному в залежності від потреб користувача.

Система може зберігати записи результатів своєї роботи для забезпечення контролю та якості виконання. Оскільки система використовує штучний інтелект, для кожного розпізнавання надається відсоток, що вказує з якою імовірністю програма впевнена що бачить саме необхідний об'єкт.

Варто зазначити, що використання подібного програмного забезпечення може йти в розріз із законами про безпеки персональних даних та конфіденційність, тому перед її інтеграцією у власну справу варто провести консультацію із юристом для дотримання усіх необхідних норм та прав людини.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для демонстрації роботи програми проведемо тестування усіх функцій та відобразимо результати.

Після відкриття програми користувач може побачити власну веб-камеру та інтерфейс. Веб-камера відображається оскільки вона за замовчуванням вказано в конфігураційному файлі ПЗ (рис. 12).

Після натиснення кнопки Start Recording її статус зміниться на Stop recording, а система почне записувати відео (рис. 13). Відео буде збережене в окремому файлі, який пізніше можна переглянути (рис. 14).

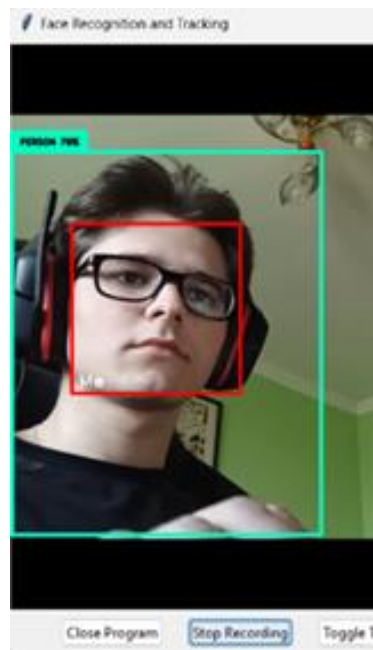


Рис. 12 Перевірка функції запису відео

Програмне забезпечення працює і виділяє навіть невідомих людей (рис. 13).

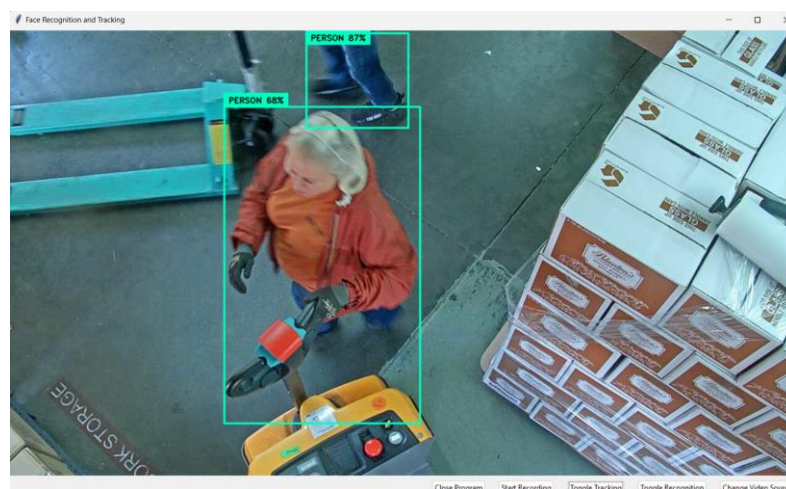


Рис. 13. Програмне забезпечення з новим джерелом відео



ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Реалізовано інформаційну систему відстеження та розпізнавання рухомих об'єктів. Система розділена на три основні модулі: розпізнавання облич, відстеження людей і збереження результатів розпізнавання.

Описано використання сучасних технологій і алгоритми машинного навчання в роботі, зокрема YOLOv7 для відстеження людей та бібліотеку Face Recognition для розпізнавання облич.

Спроектвана інформаційна система дозволяє без технічних знань, швидко та безкоштовно впроваджувати на підприємствах сучасну технологію відстеження та розпізнавання.

Досліджено структуру, функціональність та процеси інформаційної системи. За допомогою дерева цілей визначено функціональність системи, яка необхідна для досягнення мети. Декомпозиція процесів дозволила детально розглянути кожен з ключових аспектів системи та розкрити послідовність кроків та потоків даних, необхідних для їх реалізації. Побудова діаграми ієрархії процесів дала можливість якісно відобразити взаємозв'язки між усіма процесами та підпроцесами системи, демонструючи логічну послідовність їх виконання. ER-діаграма визначила структуру бази даних, що була використана для зберігання інформації про осіб. Системний аналіз заклав основу для подальшого проектування та розробки інформаційної системи відслідковування та розпізнавання людей. Він дозволив визначити основні функціональні вимоги, структуру системи та взаємозв'язки між її компонентами.

Використано мову програмування Python разом із бібліотеками YoloV7 та Face Recognition. Вибір було зумовлено ефективністю представлених варіантів, а також їх відповідністю до поставлених нами вимог.

Було створено інструкцію користувача, що дозволяє зрозуміти основні особливості системи, та необхідні умови для її використання. Було сформульовано рекомендовані технічні характеристики та детально описано усі аспекти системи. Продемонстровано її роботу та проаналізовано контрольний приклад, де ми перевірили усі функції системи та їх роботоздатність.

ПОДЯКА

Дослідження виконано за підтримки Міністерства освіти і науки України «Методи та засоби ідентифікації бойових машин на основі технологій глибинного навчання для автоматизованого керування цілерозподілом» за проектом № 0124U000925.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zhu, H., Zheng, W., Zheng, Z., & Nevatia, R. (2024). Sharc: Shape and appearance recognition for person identification in-the-wild. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 6290–6300.
2. Zhu, H., Budhwant, P., Zheng, Z., & Nevatia, R. (2024). SEAS: ShapE-Aligned Supervision for Person Re-Identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 164–174.
3. Saad, R. S. M., Moussa, M. M., Abdel-Kader, N.S., et al. (2024). Deep video-based person re-identification (Deep Vid-ReID): comprehensive survey. *EURASIP J. Adv. Signal Process.*, 2024(63). <https://doi.org/10.1186/s13634-024-01139-x>



4. Dong, N., Yan, S., Tang, H., Tang, J., & Zhang, L. (2024). Multi-view information integration and propagation for occluded person re-identification. *Information Fusion*, 104, 102201.
5. Nguyen, V. D., Mirza, S., Zakeri, A., Gupta, A., Khaldi, K., Aloui, R., & Merchant, F. (2024). Tackling Domain Shifts in Person Re-Identification: A Survey and Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4149–4159.
6. Nazarkevych, M., Petrov, A., Onopriyчук, O., Oleksiv, N., & Kis, Y. (2022). Development of A Fingerprint Pattern Matching Method Using K-Means. *Electronics and information technologies*, 19, 58–65. <http://dx.doi.org/10.30970/eli.19.5>
7. Nazarkevych, M., Voznyi, Y., & Nazarkevych, H. (2021). Development of machine learning method with biometric protection with new filtration methods. *Electronic Professional Scientific Journal «Cybersecurity: Education, Science, Technique»*, 3(11), 16–30. <https://doi.org/10.28925/2663-4023.2021.11.1630>
8. Hrytsyk, V., Nazarkevych, M., Dyshko, A. (2021). Comparative Analysis of Image Recognition Methods Obtained From Sensors Of The Visible Spectrum. *Electronic Professional Scientific Journal «Cybersecurity: Education, Science, Technique»*, 4(8), 149–164. <https://doi.org/10.28925/2663-4023.2020.8.149164>
9. Python, W. (2021). Python. *Python releases for windows*, 24.
10. Diwan, T., Anirudh, G., & Temburne, J. V. (2023). Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, 82(6), 9243–9275.
11. Du, J. (2018). Understanding of object detection based on CNN family and YOLO. In *Journal of Physics: Conference Series*, 1004.
12. Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). Fast YOLO: A fast you only look once system for real-time embedded object detection in video. *arXiv preprint arXiv:1709.05943*.
13. Zhang, Z., Lu, X., Cao, G., Yang, Y., Jiao, L., & Liu, F. (2021). ViT-YOLO: Transformer-based YOLO for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2799–2808.
14. Li, S., Li, Y., Li, Y., Li, M., & Xu, X. (2021). YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection. In *IEEE Access*, 9, 141861–141875. <https://doi.org/10.1109/ACCESS.2021.3120870>

**Nazar Ohonoik**

Student

Information Systems And Networks Department,
Lviv Polytechnic National University, Lviv, Ukraine

ORCID ID: 0009-0001-1934-7632

nazar.ohoniuk.itis.2020@lpnu.ua**Mariia Nazarkevych**Professor, Information Systems And Networks Department,
Lviv Polytechnic National University, Lviv, Ukraine

ORCID ID: 0000-0002-6528-9867

mariia.a.nazarkevych@lpnu.ua**Yurii Myshkovskiy**

Postgraduate

Information Systems And Networks Department,
Lviv Polytechnic National University, Lviv, Ukraine

ORCID ID: 0009-0004-0051-026X

yurii.i.myshkovskiy@lpnu.ua**Nazar Nakonechnyi**

Postgraduate

Information Systems And Networks Department,
Lviv Polytechnic National University, Lviv, Ukraine

ORCID ID: 0009-0000-2456-3498

Nazar.I.Nakonechnyi@lpnu.ua**Romanchuk Roman**

Postgraduate

Information Systems And Networks Department,
Lviv Polytechnic National University, Lviv, Ukraine

ORCID ID: 0009-0004-4352-1073

roman.v.romanchuk@lpnu.ua

BUILDING A MOVING OBJECT IDENTIFICATION SYSTEM BASED ON MACHINE LEARNING TECHNOLOGIES

Abstract. The study is devoted to the construction of a system for identifying moving objects in a video stream based on machine learning technologies. Tracking and recognizing moving objects is an urgent task of our time. It is important to recognize objects in motion and identify them based on artificial intelligence. The system is divided into three main modules: face recognition, people tracking, and saving of recognition results. The use of modern technologies and YOLOv7 machine learning algorithms for tracking people and the Face Recognition library for face recognition is described. A contextual Data flow diagram is created, which shows the sequence of steps required to convert the input video stream into normalized face images that are ready for further recognition. The hierarchy of processes of the moving object identification system is built. The video processing process decomposition diagram shows the logical sequence of stages and data flows required to prepare face images. Behavior classification associates detected motion patterns with specific types of behavior. The system uses facial identification data and information about their previous behavior to classify movement patterns. The process decomposition allowed us to consider in detail each of the key aspects of the system and reveal the sequence of steps and data flows required for their implementation. Building a process hierarchy diagram made it possible to qualitatively display the relationships between all processes and subprocesses of the system, demonstrating the logical sequence of their execution. The ER diagram defined the structure of the database used to store information about individuals. The system analysis laid the foundation for the further design and development of the information system for tracking and recognizing people. It allowed us to determine the main functional requirements, the structure of the system and the relationships between its components. particular importance is the ability to use the software to prevent terrorist



and sabotage threats. Thanks to such information systems, it is possible to improve the economic situation of both individual facilities and the country as a whole.

Keywords: video stream; information system; recognition; identification.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Zhu, H., Zheng, W., Zheng, Z., & Nevatia, R. (2024). Sharc: Shape and appearance recognition for person identification in-the-wild. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 6290–6300.
2. Zhu, H., Budhwant, P., Zheng, Z., & Nevatia, R. (2024). SEAS: ShapE-Aligned Supervision for Person Re-Identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 164–174.
3. Saad, R. S. M., Moussa, M. M., Abdel-Kader, N.S., et al. (2024). Deep video-based person re-identification (Deep Vid-ReID): comprehensive survey. *EURASIP J. Adv. Signal Process.*, 2024(63). <https://doi.org/10.1186/s13634-024-01139-x>
4. Dong, N., Yan, S., Tang, H., Tang, J., & Zhang, L. (2024). Multi-view information integration and propagation for occluded person re-identification. *Information Fusion*, 104, 102201.
5. Nguyen, V. D., Mirza, S., Zakeri, A., Gupta, A., Khaldi, K., Aloui, R., & Merchant, F. (2024). Tackling Domain Shifts in Person Re-Identification: A Survey and Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4149–4159.
6. Nazarkevych, M., Petrov, A., Onopriyuk, O., Oleksiv, N., & Kis, Y. (2022). Development of A Fingerprint Pattern Matching Method Using K-Means. *Electronics and information technologies*, 19, 58–65. <http://dx.doi.org/10.30970/eli.19.5>
7. Nazarkevych, M., Voznyi, Y., & Nazarkevych, H. (2021). Development of machine learning method with biometric protection with new filtration methods. *Electronic Professional Scientific Journal «Cybersecurity: Education, Science, Technique»*, 3(11), 16–30. <https://doi.org/10.28925/2663-4023.2021.11.1630>
8. Hrytsyk, V., Nazarkevych, M., Dyshko, A. (2021). Comparative Analysis of Image Recognition Methods Obtained From Sensors Of The Visible Spectrum. *Electronic Professional Scientific Journal «Cybersecurity: Education, Science, Technique»*, 4(8), 149–164. <https://doi.org/10.28925/2663-4023.2020.8.149164>
9. Python, W. (2021). Python. *Python releases for windows*, 24.
10. Diwan, T., Anirudh, G., & Tembhrne, J. V. (2023). Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, 82(6), 9243–9275.
11. Du, J. (2018). Understanding of object detection based on CNN family and YOLO. In *Journal of Physics: Conference Series*, 1004.
12. Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). Fast YOLO: A fast you only look once system for real-time embedded object detection in video. *arXiv preprint arXiv:1709.05943*.
13. Zhang, Z., Lu, X., Cao, G., Yang, Y., Jiao, L., & Liu, F. (2021). ViT-YOLO: Transformer-based YOLO for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2799–2808.
14. Li, S., Li, Y., Li, Y., Li, M., & Xu, X. (2021). YOLO-FIRI: Improved YOLOv5 for Infrared Image Object Detection. In *IEEE Access*, 9, 141861–141875. <https://doi.org/10.1109/ACCESS.2021.3120870>

