

ДОДАТОК А

Висвітлення результатів кваліфікаційної роботи

Міністерство освіти і науки України



NURE

Харківський національний університет
радіоелектроніки

ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2024

(Випуск 2)

[електронне видання]



<http://nure.ua/department/kafedra-komp-yuterno-integrovanih-tehnologiy-avtomatizatsiyi-ta-mehatroniki-kitam>



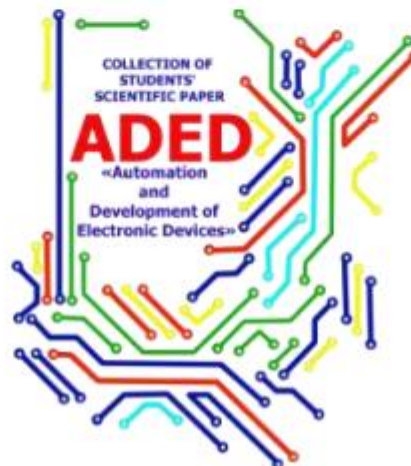
<http://itez.zntu.edu.ua/>



<http://kafea.kdu.edu.ua>

Харків 2024

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(КІТАР)



ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2024

(Випуск 2)

[електронне видання]

Харків 2024

Головний редактор	Невлюдов Ігор Шакирович , доктор технічних наук, професор, завідувач кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки.
Редакційна колегія:	<p>Филипенко Олександр Іванович, доктор технічних наук, професор, декан факультету Автоматики та комп'ютеризованих технологій, Харківського національного університету радіоелектроніки.</p> <p>Цимбал Олександр Михайлович, доктор технічних наук, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки.</p> <p>Андрусевич Анатолій Олександрович, доктор технічних наук, професор, начальник Криворізького коледжу національного авіаційного університету</p> <p>Косенко Віктор Васильович, доктор технічних наук, професор, зам. директора Державного підприємства «Південний державний проектно-конструкторський та науково-дослідний інститут авіаційної промисловості».</p> <p>Замірець Микола Васильович, доктор технічних наук, професор, директор Державного підприємства Науково-дослідного технологічного інституту приладобудування.</p> <p>Свищ Володимир Митрофанович, доктор технічних наук, професор, радник директора Державне науково-виробниче підприємство «Об'єднання Комунар».</p> <p>Фомовська Олена Владиславівна, кандидат технічних наук, доцент завідувач кафедри «Електронних апаратів» Кременчуцького національного університету імені Михайла Остроградського.</p> <p>Кухаренко Дмитро Володимирович, кандидат технічних наук, доцент кафедри «Електронних апаратів» Кременчуцького національного університету імені Михайла Остроградського</p> <p>Демська Наталія Павлівна, кандидат технічних наук, доцент кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки.</p> <p>Фурманова Наталія Іванівна, кандидат технічних наук, доцент, декана факультета Радіоелектроніки і телекомунікацій, Національного університету «Запорізька політехніка».</p>
Відповідальний редактор:	Євсєєв Владислав В'ячеславович , доктор технічних наук, професор кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки, Харківського національного університету радіоелектроніки.

Автоматизація та Приладобудування («Automation and Development of Electronic Devices» ADED-2024) [Електронний ресурс]: збірник студентських наукових статей / Харківський національний університет радіоелектроніки; [редкол.: І.Ш. Невлюдов та ін.]. – Харків : ХНУРЕ, 2024. – Вип. 2. – 290с.

Collection of Students' Scientific Paper «Automation and Development Of Electronic Devices» ADED-2024 Part 2 (Key infrastructure 2024) - Kharkiv/ The Editorial.: Nevlyudov I.Sh. (head), that all. Kharkiv: Kind of Kharkiv National University of Radio Elektronik [electronic edition], 2024. – 290p with.

Рекомендовано рішенням
Науково-технічної ради
Харківського національного
університету радіоелектроніки
протокол №6 від 29.11.2018

Рекомендовано рішенням Вченої ради
факультету Автоматики і комп'ютеризованих технологій
Харківського національного
університету радіоелектроніки
протокол № 4 від 26.12.2024

Збірник містить наукові статті здобувачів першого (бакалаврського), другого (магістерського) рівнів вищої освіти кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки (КІТАР) Харківського національного університету радіоелектроніки, кафедри Інформаційних технологій електронних засобів (ІТЕД) Запорізького національного технічного університету та кафедри Електронних апаратів (ЕА) Кременчуцького національного університету ім. М. Остроградського які навчаються за спеціальностями: 151 Автоматизація та комп'ютерно-інтегровані технології, 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка; 172 Телекомунікації та радіотехніка, 171 Електроніка та 163 Біомедична інженерія. Статті надані в авторській редакції.

©ХНУРЕ, 2024 рік

<i>Александрович Д.П.</i> Розроблення автоматизованої системи віддаленого керування аварійним електропостачанням на виробничому підприємстві	138
<i>Васенко А.В.</i> Аналіз розвитку систем автоматичного розпізнавання автомобільних номерів	145
<i>Водяницький М.А.</i> Розробка системи розумного доступу до виробничого приміщення з використанням технологій комп'ютерного зору	147
<i>Глушенко О.Г.</i> Аналіз ефективності інфрачервоних нагрівачів для монтажу та демонтажу SMD та BGA компонентів	152
<i>М.С. Греков</i> Безпілотна робототехнічна мобільна платформа для надання гуманітарної допомоги...	157
<i>Жуков А.І.</i> Підсистема для оптимізації взаємодії між державними органами та людьми з обмеженими можливостями	164
<i>Жукова Л.С.</i> Автоматизована підсистема розрахунку компенсацій і пільг для працівників промислових підприємств	170
<i>Редькін К.С.</i> Інтеграція газових котлів з системою сучасного теплозабезпечення України	176
<i>Карпенко А.</i> Overview at modern mine detecting robots	181
<i>Краснопольоров М.Р., Казановська К.А.</i> Автоматизація логістичних систем з використанням кіберфізичних підходів	186
<i>Кривенко Д.</i> Автоматизація ідентифікації вантажів на бондових складах	191
<i>Мірошниченко Ю.М.</i> Аналіз сучасних робототехнічних комплексів	196
<i>Олінкевич Я.В.</i> SRM-система в сучасному підприємстві: ефективне управління бізнес-процесами	202
<i>Погребняк В.В.</i> Дослідження методів обробки зображень за допомогою бібліотеки OPENCV для пошуку дефектів на поверхні друкованих виробів за технологією FDM/FFF	207
<i>Ісмайлов Т.В.</i> Розробка алгоритму підвищення точності локалізації та навігації рухомих об'єктів	214
<i>Ilya Karpenko</i> analysis of limitations on the design of a small-dimensional robot for investigating damage to panel buildings	219
<i>Дмитрієв Д.В.</i> Розробка реконфігурованого мобільного робота	223
<i>Бельков Д.О.</i> Інтелектуальна система управління мікрокліматом у складському приміщенні	285

УДК 004.93

РОЗРОБКА СИСТЕМИ РОЗУМНОГО ДОСТУПУ ДО ВИРОБНИЧОГО ПРИМІЩЕННЯ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ

М.А. Водяницький

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки 14

E-mail: mykhailo.vodianytskyi@nure.ua

Анотація: у статті розглядаються сучасні системи розумного доступу до приміщення, що базуються на технології комп'ютерного зору. Проаналізовано основні положення теорії розпізнавання образів, на основі теоретичних відомостей описані методи розпізнавання обличчя та способи їх реалізації на базі ESP32-CAM.

Ключові слова: комп'ютерний зір, автоматизація, доступ, ESP32-CAM, розпізнавання лиця.

DEVELOPMENT OF A SYSTEM FOR SMART ACCESS TO PRODUCTION FACILITIES USING COMPUTER VISION TECHNOLOGIES

M. Vodianytskyi

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, Nauky av., 14

E-mail: mykhailo.vodianytskyi@nure.ua

Annotation: In the article deals with modern systems of smart access to premises based on computer vision technology. The main provisions of the theory of pattern recognition are analysed, and based on theoretical information, face recognition methods and methods of their implementation on the basis of Arduino ESP32-CAM are described.

Key words: computer vision, automation, access, ESP32-CAM, face recognition.

АКТУАЛЬНІСТЬ РОБОТИ. Безпека виробничих приміщень завжди була актуальною проблемою. З розвитком технологій, коли мега корпорації витрачають мільярди доларів на побудову виробництв, найму працівників, створення технологій – одне виробниче приміщення може зберігати таємниці на мільйони доларів. Тож важливою задачею є охорона таких місць. Одним з методів може бути розробка системи розумного доступу на базі комп'ютерного зору. Метою роботи є підвищення безпеки доступу на підприємстві, мінімізуючи потребу у фізичному контролі шляхом розробки системи розумного доступу на основі технологій комп'ютерного зору. Задачею роботи є розробка такого рішення на базі ESP32.

МАТЕРІАЛ І РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ. Розвиток мікроелектроніки у двадцять першому сторіччі, у сфері відеокарт, дав великий поштовх у розвитку систем комп'ютерного зору. Комп'ютерний зір (computer vision) – теорія та технологія створення машин, які можуть проводити виявлення, відстежування та визначення об'єктів. Як наукова дисципліна комп'ютерний зір належить до теорії та технології створення штучних систем, які отримують інформацію у вигляді зображень. Головним користувачем даної теорії у наш час є нейронні мережі. Збільшення потужності відеокарт дозволило навчати мережі швидше та на більшій кількості вхідних даних. Завдяки цьому нейронні мережі можуть зчитувати інформацію з зображень за доли секунди, оброблювати її та видавати відповідь. Наприклад можна надіслати мережі фото листку з задачею по фізиці і мережа знайде рішення. Але до буму їх розвитку, інформація з зображення зчитувалась іншими технологіями, які підпорядковуються теорії розпізнавання образів.

Теорія розпізнавання образів – розділ кібернетики, що розвиває теоретичні основи й методи класифікації і ідентифікації предметів, явищ, процесів, сигналів, ситуацій і об'єктів, які характеризуються скінченним набором властивостей і ознак. Розпізнавання образів включається в ширшу наукову дисципліну – теорію машинного навчання, метою якої є розробка методів побудови алгоритмів, що здатні навчатися. Прикладом задачі, що потребує такого підходу, може служити система безпілотного керування автомобілем від компанії Tesla. Ця система повинна зчитувати велику кількість зовнішньої інформації та приймати миттєві рішення: колір світла світлофору на перехресті, дорожні знаки, інші машини, стан дорожнього покриття, погодні явища, стан автомобілю. Наш мозок виконує такі задачі автоматично, шляхом спрощення вхідних даних та зведення їх до чогось подібного, тобто наш мозок автоматично проводить класифікацію явищ навколишнього світу на групу схожих явищ. Комплекс апаратних і програмних рішень, здатних симулювати цей процес, називається системою розпізнавання образів.

Система розпізнавання образів – це комплекс апаратних і програмних засобів, призначених для ідентифікації або класифікації об'єктів (образів) на основі їх характеристик, здатний моделювати розумові процеси, властиві під час прийняття рішень із метою виявлення аналогій серед досліджуваних об'єктів. Система має велику кількість методів роботи із даними, тому для більшої зручності вони класифікуються за спільними ознаками. Самі системи розпізнавання поділяються на складні та прості. Прості характеризуються єдиною фізичною природою ознак. Наприклад тільки геометричні розміри для ідентифікації об'єкту на екрані, або тільки колір певний звук. Складні системи мають неоднорідні фізичні ознаки – автопілот в автомобілі.

Доволі довгий час, системи розпізнавання лише умовно були симуляціями роботи мозку. Головною причиною було складність програмування великих програм та мала обчислювальна потужність систем. У разі роботи із письмовими даними системи частіше за все будувалися за структурними правилами – використання спеціальних граматик, що породжують мови, де кожне речення описує об'єкти конкретного класу. У випадку необхідності класифікувати зображення або реальні об'єкти через комп'ютерний зір, використовували детерміновані методи побудови систем, що базуються на геометричних методах. Цей підхід передбачав створення еталонних класів з якими порівнювались вимірюванні відстані на об'єкті.

Для спрощення програмування детермінованих систем, зазвичай використовували математичні методи фільтрації, що дозволяли виділити на зображенні потрібні ознаки, без обробки зайвого масиву даних. Найбільш простими та ефективними методами, що використовуються і у сучасних системах є бінаризація по порозу, фільтри Гаусса та Габора, оператор Кенні для визначення меж та контурів.

Але, як вже було зазначено, були доволі далекі від симуляції роботи мозку людини, вони не виконували необхідну умову – спрощення вхідних даних та зведення їх до чогось подібного в автоматичному режимі, без використання додаткової перед обробки даних. Довгий час ця проблема не могла бути вирішена через недостатню обчислювальну потужність комп'ютерів. Але справжня революція сталася у 2012 році, коли на міжнародних змагання ILSVRS, присвячених комп'ютерному зору та штучному інтелекту, була представлена система AlexNet. Це була згортовка нейронна мережа (CNN), що складалась з восьми шарів, п'яти згорткових шарів (convolutional layers) та трьох повнозв'язних шарів (fully connected layers).

Нейронні мережі, зокрема CNN, є революційними інструментами для розпізнавання і обробки зображень, завдяки своїй здатності самостійно виділяти важливі ознаки. CNN мережа складається з трьох типів шарів: перший шар вхідний, другий шар прихований і вихідний шар. Згорткові шари витягують особливості із вхідних даних, вони використовують фільтри, які «ковзають» по зображенню, виконуючи операцію згортання. У результаті формуються карти

ознак (feature maps), які показують, де виявляються певні характеристики. Після згорткового шару дані потрапляють до прихованого шару, або шару об'єднання. Його головна задача зменшити розмірність зображення, для того аби зменшити кількість навчальних параметрів, що знижує обчислювальні витрати. На завершальному етапі CNN обробляє дані через повнозв'язні шари, де кожен нейрон з'єднаний з усіма нейронами попереднього шару. Це об'єднує інформацію, створюючи узагальнену картину, яка може бути використана для класифікації.

CNN можуть виділяти ознаки без втручання людини, що робить їх дуже ефективними. Обробляючи локальні області зображення, зменшують обсяг оброблюваної інформації, дозволяючи мережі працювати швидше. Мають чудові показники точності, оскільки вони налаштовуються на розпізнавання різноманітних ознак об'єкта.

У топі помилок AlexNet досягла 15,3% (проти 26% у найближчого конкурента). Це стало першою демонстрацією того, як глибокі нейронні мережі можуть перевершувати інші алгоритми в завданнях класифікації зображень. Цей результат ставку початком «буму» нейромереж, і в наш час кожна нова система розпізнавання будується на їх базі.

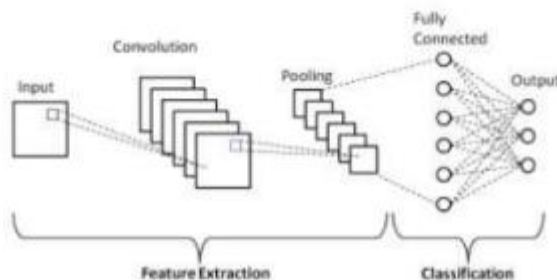


Рисунок 1 – Схема нейронної мережі типу CNN

Швидкий розвиток мікроелектроніки дав поштовх не лише у сфері нейромереж. Ми отримали таку систему як Arduino. Arduino – це відкрита апаратно-програмна платформа, розроблена для швидкого створення електронних пристроїв. Вона популярна серед аматорів, інженерів і розробників завдяки простоті використання та великій спільноті. На базі цих мікрочипів створюють системи розумного дому, системи відеоспостереження та інше. Одна з можливостей це спільне використання з модулем ESP32-CAM. Цей модуль представляє собою плату із чіпом ESP32S та камерою, який підтримує функції комп'ютерного зору, включно з розпізнаванням облич. Ця функція надається бібліотекою ESP-WHO, яка розроблена Espressif спеціально для комп'ютерного зору. У бібліотеці доступні приклади коду для розпізнавання облич, які можна використовувати з ESP32-CAM.

Камера ESP32-CAM робить знімок у режимі реального часу. Спочатку застосовується алгоритм виявлення облич, наприклад Haar Cascade або MTCNN. На зображенні виділяються області, які містять обличчя. Після виявлення обличчя мережа витягує ключові особливості (наприклад, відстань між очима, форму носа) і перетворює ці дані в числовий вектор. Вектор обличчя порівнюється зі збереженими в базі даних векторами, через алгоритм k-найближчих сусідів або відстань косинусної близькості.



Рисунок 2 – Розпізнавання обличчя бібліотекою ESP-WHO

Але такий підхід доволі обмежений у своїх можливостях. Потужності чіпу не дають змоги використовувати великі масиви даних для порівняння, не дають змоги використовувати нейронної мережі. Вирішенням цієї проблеми є використання функції ESP32 WebServer, що надає змогу передавати відео або фото на окремий сервер, де вже відбувається обробка. Створити таку систему можливо власноруч, або за допомогою готових програмних рішень типу Home Assistant. Home Assistant запускає сервер на Python і використовує базу даних для зберігання інформації, що працює на локальному пристрої, це може бути Raspberry Pi, окремий сервер на Linux чи віртуальна машина. Сервер зв'язується з компонентами розумного дому (датчики руху, температури, світла), або IP-камерами що передають відео формату RTSP, якою є ESP32. Сам сервер працює як центральний вузол, що зв'язую усі компоненти і застосовує для пристроїв сценарії роботи. Тож у випадку коли потрібно ідентифікувати обличчя використовуються сторонні інтеграції, наприклад Microsoft Azure Face API. Сервер передає зображення для ідентифікації через Azure API на сервери Microsoft, які повертають результат – на зображенні ідентифіковано людину. Якщо обличчя відоме сервер Home Assistant може, наприклад, надати команду відкрити двері, або навпаки – обличчя не відоме, відправити сповіщення.

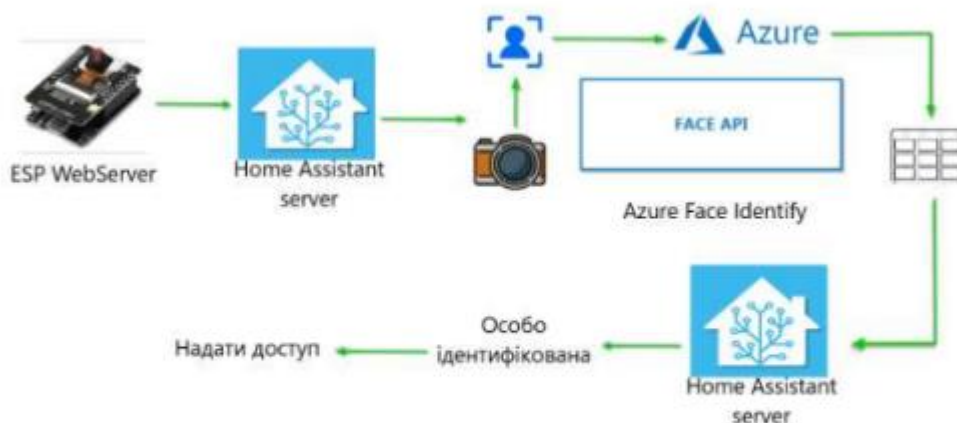


Рисунок 3 – Загальний принцип системи ідентифікації на базі Home Assistant

ВИСНОВКИ. У підсумку: був проведений аналіз положень теорії розпізнавання образів, описані базові типи систем розпізнавання та наданий стислий опис їх роботи із основними особливостями, описано принципи роботи мережі CNN, її переваги та вплив на подальший розвиток галузі, описані загальні методи роботи з ESP32-CAM та методи реалізації ідентифікації особи сторонніми сервісами.

ЛІТЕРАТУРА.

1. What is computer vision? // Сайт IBM.com URL: <https://www.ibm.com/topics/computer-vision>.
2. Основи теорії розпізнавання образів, навчальний посібник у двох частинах, А.С. Довбиш, І.В. Шелехов // Сайт НТУ Національний транспортний університет URL: <http://kist.ntu.edu.ua/textPhD/tro2.pdf>.
3. Теорія розпізнавання образів, навчальний посібник, В.В. Гавриленко, Г.Ф. Іванченко, Є.Г. Шевченко // Сайт studfile URL: <https://studfile.net/preview/5607072/>.
4. What is Thresholding in Image Processing? A Guide. // Сайт Roboflow Blog URL: <https://blog.roboflow.com/image-thresholding/>.
5. Face Recognition Techniques – A Review // Сайт Ajay Kumar Garg Engineering College URL: <https://www.akgec.ac.in/wp-content/uploads/2019/06/2-Face-Recognition-Techniques-Bhaskar-Gupta.pdf>.
6. Face Recognition for Beginners // Сайт Medium URL: <https://towardsdatascience.com/face-recognition-for-beginners-a7a9bd5eb5c2>.
7. Convolution Neural Network (CNN) // Сайт Analytics Vidhya URL: <https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-networkcnn/>.
8. Azure Face API: Face Detection & Recognition // Сайт Medium URL: <https://medium.com/codenx/azure-face-api-face-detection-recognition-b2b954e3e939>
9. Build ESP32-CAM Projects // Сайт MyArduino URL: <https://myarduino.dk/ESP32-CAM.pdf>
10. Yevsieiev, V., & Demska, N. (2024). *A Model of Using Computer Vision to Monitor the Environment of a Collaborative Manipulator Robot* (Doctoral dissertation).
11. Nevludov, I., Yevsieiev, V., Maksymova, S., Demska, N., Starodubcev, N., & Klymenko, O. (2023, September). Monitoring System Development for Equipment Upgrade for IioT. In *2023 IEEE 5th International Conference on Modern Electrical and Energy System (MEES)* (pp. 1-5). IEEE.
12. Nevludov, I. S., Yevsieiev, V. V., Demska, N. P., & Kostrova, H. Y. (2023). Development and Improvement of the Design of a Lightweight Mobile Robot Manipulator Using Generative Design.
13. Yevsieiev, V., & Demska, N. (2023). *Application of Generative Design Methods for Improving Manipulator Designs for Mobile Robots* (Doctoral dissertation, Черкаського національного університету імені Богдана Хмельницького).
14. Nevludov, I., Yevsieiev, V., Maksymova, S., Demska, N., Kolesnyk, K., & Miliutina, O. (2022, September). Object Recognition for a Humanoid Robot Based on a Microcontroller. In *2022 IEEE XVIII International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH)* (pp. 61-64). IEEE.
15. Євсєєв, В. В., Демська, Н. П., & Олександров, Ю. М. (2022). *Моделювання виробничої лінії SMT-монтажу в кібер-фізичних виробничих системах* (Doctoral dissertation, Кременчуцький національний університет імені Михайла Остроградського).

Науковий керівник: Гурін Дмитро Валерійович, старший викладач кафедри КІТАР, Харківський національний університет радіоелектроніки

ДОДАТОК Б

Код реалізації програми OpenCV

Файл collect.py

```
import cv2

user_id: int = int(input("Enter user ID: "))

face_cascade = cv2.CascadeClassifier(cv2.data.harcascades +
"haarcascade_frontalface_default.xml")

cap = cv2.VideoCapture(0)

cv2.namedWindow("live transmission", cv2.WINDOW_AUTOSIZE)

count: int = 0

max_count: int = 500

while count < max_count:
    ret, frame = cap.read()
    if not ret:
        print("Failed to capture frame, trying again...")
        continue # Move to the next iteration of the loop
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray_frame, scaleFactor=1.1,
minNeighbors=5)
    for (x, y, w, h) in faces:
        count += 1
        cv2.imwrite('DataFaces/User.' + str(user_id) + "." + str(count) + ".jpg",
gray_frame[y:y + h, x:x + w])
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
    if count >= max_count:
        print("Dataset collection done.....")
        break
    cv2.imshow('live transmission', frame)
    key = cv2.waitKey(5)
    if key == ord('q'):
```

```

        break
cap.release()
cv2.destroyAllWindows()

Файл train.py
import cv2
import numpy as np
from PIL import Image
import os
recognizer = cv2.face.LBPHFaceRecognizer_create()
path="DataFaces"
def getImageID(path):
    imagePath = [os.path.join(path, f) for f in os.listdir(path)]
    faces=[]
    ids=[]
    for imagePaths in imagePath:
        faceImage = Image.open(imagePaths).convert('L')
        faceNP = np.array(faceImage)
        Id= (os.path.split(imagePaths)[-1].split(".")[1])
        Id=int(Id)
        faces.append(faceNP)
        ids.append(Id)
        cv2.imshow("Training",faceNP)
        cv2.waitKey(1)
    return ids, faces
IDs, facedata = getImageID(path)
recognizer.train(facedata, np.array(IDs))
recognizer.write("LBPH_Trainer_500.yml")
cv2.destroyAllWindows()
print("Training Completed.....")

```

Файл main.py

```

import cv2
import time
import timeit

video = cv2.VideoCapture(0)
face_cascade = cv2.CascadeClassifier(cv2.data.harcascades +
"haarcascade_frontalface_default.xml")
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read("LBPH_Trainer_500.yml")
name_list = ["", "User 1", "User 2"]

while True:
    ret, frame = video.read()
    if not ret:
        print("Failed to capture frame, trying again...")
        continue # Move to the next iteration of the loop
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    face_detection_start = timeit.default_timer()
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    face_detection_time = (timeit.default_timer() - face_detection_start) * 1000
    for (x, y, w, h) in faces:
        recognition_start = timeit.default_timer()
        serial, conf = recognizer.predict(gray[y:y + h, x:x + w])
        recognition_time = (timeit.default_timer() - recognition_start) * 1000
        print(f"onf: {conf}")
        print(f"Face Detection Time: {face_detection_time:.5f} mseconds")
        print(f"Face Recognition Time: {recognition_time:.5f} mseconds")
        if conf < 60:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 1)

```

```

cv2.rectangle(frame, (x, y), (x + w, y + h), (50, 50, 255), 2)
cv2.rectangle(frame, (x, y - 40), (x + w, y), (50, 50, 255), -1)
cv2.putText(frame, name_list[serial], (x, y - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (255, 255, 255), 2)
else:
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 1)
    cv2.rectangle(frame, (x, y), (x + w, y + h), (50, 50, 255), 2)
    cv2.rectangle(frame, (x, y - 40), (x + w, y), (50, 50, 255), -1)
    cv2.putText(frame, "Unknown", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.8, (255, 255, 255), 2)
    cv2.imshow("Frame", frame)

k = cv2.waitKey(1)
if k == ord('o') and conf > 50:
    time.sleep(10)
if k == ord("q"):
    break
video.release()
cv2.destroyAllWindows()

```

ДОДАТОК В

Код реалізації програми Face Recognition

Файл data_collect.py

```
import cv2
import face_recognition
import os
username = input("Enter username: ")
max_count = int(input("Enter the number of photos to collect: "))
count: int = 0
user_dir = f"dataset/{username}"
os.makedirs(user_dir, exist_ok=True)
cap = cv2.VideoCapture(0)
while count < max_count:
    ret, frame = cap.read()
    if not ret:
        print("Failed to capture frame, trying again...")
        continue # Move to the next iteration of the loop
    face_locations = face_recognition.face_locations(frame)
    if face_locations:
        top, right, bottom, left = face_locations[0]
        face_img = frame[top:bottom, left:right]
        filename = os.path.join(user_dir, f"{username}_{count}.jpg")
        cv2.imwrite(filename, face_img)
        cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)
        cv2.putText(frame, f"Photos: {count + 1}/{max_count}",
                    (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        count += 1
    if count >= max_count:
        print("Dataset collection done.....")
```

```

    break
cv2.imshow('Face Collection', frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
cap.release()
cv2.destroyAllWindows()

```

Файл main.py

```

import cv2
import face_recognition
import urllib.request
import numpy as np
import os
import pickle
import timeit

def encode_known_faces():
    known_faces = []
    known_names = []
    # Цикл проходу по папкам користувачів
    for username in os.listdir("dataset"):
        user_dir = os.path.join("dataset", username)
        # Цикл проходу по файлам в папці користувача
        for filename in os.listdir(user_dir):
            image_path = os.path.join(user_dir, filename)
            try:
                # Завантаження і кодування
                image = face_recognition.load_image_file(image_path)
                encodings = face_recognition.face_encodings(image)
                # Якщо кодування успішне завантажуюмо

```

```

    if encodings:
        known_faces.append(encodings[0])
        known_names.append(username)
        print(f"Face successfully encoded: {image_path}")
    else:
        print(f"Could not find face in image: {image_path}")
# Помилка на файлі №
except Exception as e:
    print(f"Processing error {image_path}: {e}")
# Збереження закодованих обличь
with open('known_faces.pkl', 'wb') as f:
    pickle.dump({
        'encodings': known_faces,
        'names': known_names
    }, f)
print(f"Encoded persons: {len(known_faces)}")
return known_faces, known_names

def load_encoded_faces():
    try:
        with open('known_faces.pkl', 'rb') as f:
            data = pickle.load(f)
            return data['encodings'], data['names']
    except FileNotFoundError:
        print("Файл с кодировками не найден. Выполняется кодирование...")
        return encode_known_faces()

def recognize_faces():
    # Загрузка закодованных значений через виклик функції
    known_faces, known_names = load_encoded_faces()

```

```

# Перевірка наявності відомих обличь
if not known_faces:
    print("No known faces to recognize!")
    return

# Захоплення камери
cap = cv2.VideoCapture(0)
recognition_times = []
while True:
    # Захоплюємо кадр
    ret, frame = cap.read()
    # Знаходимо в кадрі обличчя
    face_locations = face_recognition.face_locations(frame)
    face_encodings = face_recognition.face_encodings(frame, face_locations)
    # Обробка знайдених облич
    for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):
        # Сравнение с известными лицами
        recognition_start = timeit.default_timer()
        matches = face_recognition.compare_faces(known_faces, face_encoding)
        recognition_time = (timeit.default_timer() - recognition_start) * 1000
        name = "Unknown"
        recognition_times.append(recognition_time)
        print(f"Face Recognition Time: {recognition_time:.5f} ms")
        # Визначаємо ім'я якщо особа розпізнана
        if True in matches:
            first_match_index = matches.index(True)
            name = known_names[first_match_index]
        # Прямокутник і ім'я
        cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)
        cv2.putText(frame, name, (left, top - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)

```

```
# Поточний кадр
cv2.imshow('Face Recognition', frame)
# Вихід за клавішею 'q'
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
# Звільняємо ресурси
cap.release()
cv2.destroyAllWindows()

def prepare_face_recognition():
    # Перевірка наявності файлу кодування
    if not os.path.exists('known_faces.pkl'):
        print("Encoding file not found. Initial encoding in progress...")
        encode_known_faces()

def main():
    prepare_face_recognition()
    recognize_faces()

if __name__ == "__main__":
    main()
```

ДОДАТОК Г
Демонстраційний матеріал

