Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: Розробка масштабованої високонавантаженої системи зберігання та обробки відеоданих

Виконав: студент IV курсу, групи СН-43
спеціальності 122 Комп’ютерні науки
(шифр і назва спеціальності)

_____ Бронецький А.І.
(підпис) (прізвище та ініціали)

Керівник _____ Литвиненко Я.В.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ Шимчук Г.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент _____ Тиш Є.В.
(підпис) (прізвище та ініціали)

Тернопіль - 2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
«__» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Бронецькому Андрію Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка масштабованої високонавантаженої системи зберігання та обробки відеоданих

Керівник роботи Литвиненко Ярослав Володимирович, д.т.н., проф.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «07» 05 2025 року № 4/7-444

2. Термін подання студентом завершеної роботи 21.06.2025 р.

3. Вихідні дані до роботи наукові літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ
1. Моніторинг існуючих систем відеоаналітики і концепції хмарних і кордонних обчислень.
2. Теоретична частина
3. Проектування та реалізація системи
4. Безпека життєдіяльності, основи охорони праці
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Титулка. 2. Актуальність роботи 3. Мета, задачі дослідження. 4. Порівняння систем. відеоаналітики. 5. Засоби для аналізу відеопотоку. 6. Алгоритми та засоби ідентифікації обличь на відеоданих 7. CNN. 8. Засоби програмної розробки. 9. Шаблон проектування 10. ER -діаграма БД. 11. Створення ресурсної групи та облікового запису зберігання даних. 12. Клас Azure Manager для взаємодії з хмарним сховищем та аналізу зображень. 13. Скріншоти роботи системи. 14. Висновки. Основні результати проведеного дослідження.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В.С., доц. каф. МТ		

7. Дата видачі завдання 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	08.06 – 09.05.25	*Виконано*
2.	Підбір джерел про зберігання та обробку відеоданих	10.05 – 13.05.25	*Виконано*
3.	Опрацювання джерел про системи зберігання та		
	обробку відеоданих	14.05 – 18.05.25	*Виконано*
4.	Виконання дослідження щодо розробки програмного		
	забезпечення для зберігання та обробки відеоданих	19.05 – 22.05.25	*Виконано*
5	Розроблення програмного коду	23.05 – 25.05.25	*Виконано*
6.	Оформлення розділу «Моніторинг існуючих систем		
	відеоаналітики і концепції хмарних і кордонних		
	обчислень»	24.05 – 25.05.25	*Виконано*
7.	Оформлення розділу «Теоретична частина»	26.05 – 27.05.25	*Виконано*
8.	Оформлення розділу «Проектування та реалізація		
	системи.»	28.05 – 30.05.25	*Виконано*
9.	Виконання завдання до підрозділу «Безпека		
	життєдіяльності, основи хорони праці»	31.05 – 03.06.25	*Виконано*
10.	Оформлення кваліфікаційної роботи	20.05 – 03.06.25	*Виконано*
11.	Нормоконтроль	01.06 – 05.06.25	*Виконано*
12.	Перевірка на плагіат	04.06 – 10.06.25	*Виконано*
13.	Попередній захист кваліфікаційної роботи	01.06 – 18.06.25	*Виконано*
14.	Захист кваліфікаційної роботи	28.06.25	

Студент _____ Бронецький А.І.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Литвиненко Я.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка масштабованої високонавантаженої системи зберігання та обробки відеоданих // Бронецький Андрій Ігорович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, кафедра комп'ютерних наук, група СН–43 // Тернопіль, 2025 // С. – 57, рис. – 22, табл. – 1, слайдів – 14, бібліогр. – 20.

Ключові слова: java, HOG, microsoft azure, відеоаналітика, згорткова мережа, метод віоли – джонса

Кваліфікаційна робота присвячена розробці системи, котра здатна проводити розпізнавання та ідентифікацію людей на зображеннях з подальшим збереженням результатів обробки у хмарному сховищі.

У розділі 1 аналізуються сучасні системи та комплекси відеоаналітики, проведено їх порівняльний аналіз. Встановлено, що більшість систем відеоаналітики використовують технологію хмарних обчислень, що має низку недоліків. Розглянуто технологію кордонних обчислень, що допомагає усунути недоліки хмарних обчислень, та наведено її основні переваги.

У розділі 2 наведено засоби та методи обробки відеопотоку, у тому числі алгоритми та засоби для ідентифікації осіб на зображеннях. Встановлено, що всі обчислення найкраще проводити за допомогою графічних процесорів при використанні згорткових нейронних мереж, в той же час Microsoft Cognitive Services є платною хмарною службою, тому як алгоритм для ідентифікації осіб на зображеннях в розроблюваній системі використовується HOG. Здійснено вибір програмних інструментів для реалізації системи.

У розділі 3 представлена розробка системи обробки відеопотоку з відеокамери, вбудованої в ноутбук, і програми-тригера, що ідентифікує обличчя на зображеннях і зберігає результати в хмарне сховище.

ANNOTATION

Development of a scalable high-load video data storage and processing system // Bronetskyi Andrii // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science // Ternopil, 2025 // P. - 57, Fig. - 22, Table - 1, Slide - 14, References - 20.

Keywords: java, HOG, microsoft azure, video analytics, convolutional network, viola-jones method

The thesis deals with development of a system capable of recognizing and identifying people in images with subsequent storage of the processing results in cloud storage.

In chapter 1, modern video analytics systems and complexes are analyzed, their comparative analysis is conducted. It is established that most video analytics systems use cloud computing technology, which has a number of disadvantages. The technology of edge computing is considered, which helps to eliminate the disadvantages of cloud computing, and its main advantages are given.

Chapter 2 presents the means and methods of processing the video stream, including algorithms and tools for identifying faces in images. It has been established that all calculations are best neural networks, at the same time Microsoft Cognitive Services is a paid cloud service, therefore HOG is used as the algorithm for identifying faces in images in the system. Software tools have been selected for system implementation.

Chapter 3 presents the development of a video stream processing system from a video camera built into a laptop and a trigger program that identifies people in images and stores the results in cloud storage.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

Azure Cognitive Services – набір хмарних сервісів, що дозволяють швидко та зручно створювати програми з використанням машинного навчання та когнітивних засобів штучного інтелекту (інтелектуальних алгоритмів).

CNN (convolutional neural network) – згорткова нейронна мережа.

IDE (integrated development environment) – інтегроване середовище розробки.

HOG (histogram of oriented gradients) – гістограма напрямлених градієнтів.

OpenCV (Open Source Computer Vision Library) – бібліотека комп'ютерного зору з відкритим кодом, містить функції та алгоритми комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом.

SVM (Support vector machine) – метод опорних векторів.

БД – база даних.

МН – машинне навчання.

ОЗЗ – обліковий запис зберігання.

ОС – операційна система.

ПЗ – програмне забезпечення.

СКУД – система контролю і управління доступом.

ШІ – штучний інтелект.

ЗМІСТ

ВСТУП

Актуальність теми. На сьогоднішній день галузь відеоспостереження поширена і застосовується як для особистих, так і професійних цілей. Забезпечення безпеки об'єктів – одна з основних цілей застосування камер відеоспостереження. Однак надто велика кількість камер ускладнює роботу оператора, а часом вона може стати нездійсненною. Наприклад, один оператор не впорається з контролем сотень камер, тому доводиться наймати цілий штат співробітників. У подібних ситуаціях на допомогу приходить відеоаналітика.

Відеоаналітика є інтелектуальним аналізом даних, прийнятих від камери у вигляді сукупності відеозображень або потоку відеоданих, і автоматичне детектування заздалегідь запрограмованих подій. Аналітика може проводитися як у режимі реального часу (онлайн-режим), так і під час роботи з архівом (офлайн-режим). Результати роботи - події, які можна передати оператору у вигляді повідомлень або записати до архіву для подальшого ведення пошуку по них та складання звітів. У відеоаналітиці виділяють кілька самостійних напрямків, серед яких можна виділити базові та сервісні детектори.

Базові та сервісні детектори - одні з найпростіших та найефективніших детекторів руху, зміщення фокусу, засвітки, закриття камери, які на сьогоднішній день або присутні майже в кожному програмному забезпеченні відеоспостереження, або при необхідності можуть бути легко вбудовані у функціонал камери. Детектор розпізнавання – приклад базового детектора. Основна його мета - розпізнати обличчя людини з появою його в кадрі, відрізняючи його від інших об'єктів, проводити пошук в архівах, що містять великі обсяги інформації.

Сучасна якісна система відеоспостереження повинна вміти проводити низку аналітичних функцій, а не лише записувати та виводити на екран зображення. До таких функцій відноситься розпізнавання та ідентифікація людей (осіб), яка широко застосовується у таких випадках:

– протидія крадіжкам;

– СКУД;

– пропускний контроль у нічних клубах та різних розважальних закладах.

У випадках з крадіжками класична система відеоспостереження, яка встановлена в магазині, зазвичай використовується для отримання доказів, коли шкода вже була завдана. У більшості випадків дрібні злочини, які приносять закладам не тільки фінансову шкоду, а й знижують їхню репутацію, роблять одні й ті самі люди. Створення бази даних, що містить особи цих людей, та використання засобів відеоаналітики допоможуть виявити зловмисників ще при вході до магазину та ретельніше контролювати їх дії.

У випадках зі СКУД, аналітика може бути інтегрована в систему безпеки та управління контролерами на турнікетах. Перевага такого підходу – мінімізація людського фактора та підвищення трудової дисципліни.

У випадках з розважальними клубами використання системи відеоаналітики з виведенням тривожного оповіщення на віддалений пристрій начальника клубу (наприклад, телефон або планшет) допоможе знизити зловживання персоналу.

Мета роботи – вивчення технологій хмарних та кордонних обчислень та розробка масштабованої високонавантаженої системи зберігання та обробки відеоданих.

Для досягнення мети виділено ряд завдань:

– вивчити існуючі системи та комплекси відеоаналітики та компонентів Microsoft Azure;

– дослідити кордонних та хмарних обчислень;

– розглянути сучасні алгоритми та засоби з обробки відеопотоку;

– порівняти вивчені алгоритми та їх застосування при розробці системи;

– спроектувати та створити програму зі збирання, зберігання та читання відеоданих з відеокамери та додатка - тригера для обробки відеоданих з подальшим збереженням результатів у хмарному сховищі.

Практичне значення. Система може бути використана в дрібних та великих закладах для забезпечення безпеки та контролю.

РОЗДІЛ 1. ОГЛЯД НАЯВНИХ СИСТЕМ ВІДЕОАНАЛІТИКИ ТА КОНЦЕПЦІЇ ХМАРНИХ І КОРДОННИХ ОБЧИСЛЕНЬ

1.1 Огляд існуючих систем та комплексів відеоаналітики

1.1.1 CasRetail

Є потужним відеоаналітичним засобом, котрий створений для контролю та аналізування. Є масштабованим інструментом для комплексних об'єктів, котре функціонує без обмеження у числі каналів із підтримкою віртуалізації. CasRetail зручно застосовувати як для малих магазинчиків, так і для потужних торгових центрів. Варто навести корисні функції інтурменту:

– опрацювання відеоархіву через власний кабінет юзера;

– побудова карт траєкторії рухів та теплової, котрі демонструють найпопулярніші зони користувачів;

– ОС Linux, котра дає змогу достатньо зменшити затрати на інтеграцію та, водночас, підвищити відмовостійкість.

Опрацьовувати відеоархів можна в онлайн режимі, застосовуючи будь-який веб-браузер, котрий дає змогу контролювати функціонування персоналу та проводити стеження за всім, що трапляється на об'єкті, не потрібно використовувати будь-яке стороннє ПЗ. Разом із тим завдяки використанню технології власне структурованого покадрового презентування відео є можливість майже миттєво здійснити оцінку поточної ситуації у торгівельному закладі без завчасного завантаження архіву.

На додачу, корисними функціями системи також є: можливість оптимізації навантаження працівників, аналіз та планування рекламних акцій, збирання даних з метою здійснення прогнозу прибутку. Як приклад, при допомозі аналізу поведінки у черзі єх можливість оптимізування праці касирів. Тут двома основними показниками є розмір черги та час, який проводять у ній покупці. В основі аналізу покладено детектор, котрий здатен розпізнавати людські голови, причому всі інші об'єкти ігноруються. Варто зазначити, що за остаточного підрахунку людей у черзі зі статистики також виключаються діти.

1.1.2 VOCORD FaceControl

Є системою ідентифікації людини за обличчям. Дане рішення успішно застосовується там, де можливе масове нагромадження людей, як то стадіони, метро, вокзали, аеропорти і т.п., тобто у громадських місцях, в яких люди напряму не взаємодіють із системою для проведення біометричної ідентифікації.

Відмінною рисою системи є її функціонування у некооперативному форматі. Це свідчить про те, що VOCORD FaceControl не потрібна безпосередня взаємодія із людиною для її детектування. Контрольні точки забезпечуються спеціалізованими камерами з метою детектування людей. За час, поки людина проходить крізь зону контролю, VOCORD FaceControl виконує декілька її фото, у тому числі для наступного аналізування беруться найкращі. У подальшому проводиться порівняння виділеної особи на фото із особами-еталонами з БД та їх on-line розпізнавання. Спеціальний архів містить всі результати аналізу, в ньому зручно проводити пошук і формувати звіти.

Дане рішення також інтегрують у системи контролю керування доступом для підвищення ефективності та широко використовують у магазинах. У рітейлі цей продукт дає змогу формувати спеціальні списки клієнтів, саме тому персонал буде попередньо сповіщений про появу в закладі VIP-особи чи ймовірного зловмисника.

Імовірність коректного розпізнавання у «польових» умовах у середньому становить 90%.

Із цією системою можна застосовувати необмежене число БД. Якщо в кадрі було виявлено злочинця, то система видасть оператору повідомлення з позначкою, де виявлено злочинця, та коментарі, за умови, що вони були початково додані.

1.1.3 VideoNet

Є цифровою системою безпеки для здійснення автоматизації властиво процесу відеоспостереження та забезпечення охорони складних і масштабованих об'єктів. Система дає змогу не тільки фіксувати події, котрі мають місце на об'єкті, котрий охороняється, але також і детектувати ймовірні небезпечні подіє

завдяки інноваційним алгоритмам опрацювання відеопотоку і ефективних відеоаналітичних методів, котрі є базою програмного рішення.

VideoNet успішно працює на багатьох об'єктах, в т.ч. промислових та військових, банки та фінансові установи, митні та прикордонні термінали, великі торгові центри та транспорті об’єкти. Головними переваги VideoNet є:

– забезпечення підтримки різноманітних IP –пристроїв під дією спеціалізованої платформи Total.IP (підтримка більше трьох тисяч IP -пристроїв від різних вендерів);

– відеозображення максимальної якості. Алгоритм DVPack 2 дає змогу отримати високий рівень стискання (значно менший розмір кадру при такій самі візуальній якості) та якість відео, покращити швидкодію [5]. Даний підхід дозволяє заощаджувати на обладнанні, необхідному для організації відеоархіву, та на існуючій конфігурації отримати більшу глибину архіву;

– широкі можливості інтеграції із іншими системами. Наприклад, із системами безпеки Orion Pro та Alpha, системою детектування автономерів та багатьма іншими;

– швидкий та надійний архів. Рішення дозволяє робити запис аж до тисячі кадрів повнокадрового відео безпосередньо, швидке відшукання даних в архіві та перехід між інформацією вперед/назад. Функціонує система для запобігання втрачанню кадрів за появи пікових навантажень чи некоректного закінчення роботи;

– вельми зручна оболонка. Будь-який користувач здатен самостійно виконати налагодження та конфігурування інтерфейс для своїх потреб, щоб досягти максимально зручної та ефективної роботи з системою;

– новітні функції відеоаналітики Наявний набір інтелектуальних детекторів, котрі можуть повністю автоматично детектувати можливі небезпечні і підозрілі моменти на об'єкті контролю.

1.1.4 ITV | Axxon Next

Дане рішення об'єднує нейромережну відеоаналітику та підтримує понад 10 тисяч моделей різноманітних IP -пристроїв.

Система є достатньо надійною і доволі простою у використанні завдяки зручному інтерфейсу користувача. Продукт зрцчний для розв’язування завдань різної складності – від невеликих фірм і аж до потужних масштабованих об'єктів. Варто відмітити ключові переваги Axxon Next:

– підтримка широкого спектра IP -камер (навіть аналогових);

– застосування мікромодульної архітектури, котра надає високу стабільність та надійність системи;

– зручний інтерфейс користувача;

– віддалена робота з системою під дією мобільних пристроїв та веб-інтерфейсу;

– чудова масштабованість;

– зручний відеоархів із можливістю «інтелектуального пошуку» (осіб та номерів автомобілів);

– апаратне прискорення: система може використовувати апаратні прискорювачі для виконання ресурсомістких обчислень нейромережевої аналітики;

– інтерактивна SD -карта, що відображає відео та розташування камер у визначеному місці для миттєвого спостереження;

– інтеграція із зовнішніми пристроями та системами;

– підключення резервного сервера у разі втрати зв'язку з основним;

– безкоштовні оновлення та підтримка;

– офлайн-аналітика: Axxon Next дає змогу проводити аналіз попередньо імпортованого відео та використовувати при цьому функцію «інтелектуального пошуку».

1.1.5 GoalCity Instinct 2.0

GoalCity Instinct 2.0 - апаратно-програмний комплекс, що легко масштабується, що дозволяє працювати з IP та й з аналоговими камерами, а також вирішувати будь-які задачі щодо забезпечення безпеки об'єктів. Головними особливостями цього рішення є наявність модульної архітектури та розподілено-мережевого принципу побудови, тому воно і є всебічною системою

безпеки.

Базою рішення є універсальна архітектура мережі, котра дає змогу об'єднати всі типи об'єктів у єдину власне структуру із застосуванням різноманітних засобів зв’язку. Із використанням сучасного мобільного ґаджету можна керувати системою властиво в режимі реального часу.

1.1.6 Порівняння систем відеоаналітики

У табл. 1.1 представлено результати порівняння основних параметрів та функцій систем, розглянутих у попередніх підрозділах.

Таблиця 1.1 – Результати порівняння досліджених систем

Функція (Критерій порівняння)	CasRetail	VOCORD FaceControl	VideoNet	ITV| Axxon Next	GoalCity Instinct 2.0
Детектор руху	+	+	+	+	+
Виділення обличчя та ідентифікація	+	+	+	+	+
Детектор активності персоналу	+	+	+	+	+
Тепловізор	+	-	+	+	+
Хмарні обчислення	+	+	+	+	-
Контроль сліпих зон	+	+	+	-	+
Спрямований рух	+	+	+	+	+
Класифікатор об'єкта	+	+	+	+	+
Мобільний застосунок та оповіщення	+	+	+	+	+

Розглянувши існуючі системи та комплекси відеоаналітики можна зробити

висновок, що в сучасних умовах систем відеоаналітики є достатня кількість, кожна з них має свої індивідуальні переваги. Впровадження таких систем у дрібні та великі заклади підвищує їхню безпеку та захищеність. Необхідно констатувати, що більшість систем відеоаналітики володіють можливостями хмарних обчислень, що часто буває невигідно через те, що такий підхід має низку недоліків.

1.2 Хмарні обчислення, їх переваги та недоліки

Хмарні обчислення є концепцією, за якою всі обчислення проводяться на віддаленому комп'ютері (у так званій «хмарі»), тоді як результати роботи демонструються у вікні веб-браузера на персональному комп'ютері. Всі додатки та дані також розташовані в «хмарі».

Основні переваги:

– зменшення вимог до власне обчислювальної потужності (ПЗ) персональних комп'ютерів;

– висока швидкість обробки даних;

– економія ресурсів локальних комп'ютерів;

– зниження витрат за електроенергію.

Однак такий підхід має і свої недоліки:

– необхідне стале Інтернет-з'єднання. Багато «хмарних» ресурсів потребують високошвидкісного Інтернет з'єднання із достатньо високою, власне, пропускною спроможністю;

– не всі програми можуть функціонувати віддалено;

– велика завантаженість хмарних машин здатна призвести до збільшення часу виконання обчислень;

– можлива загроза безпеці даних клієнтів. Тут варто відзначити, хто забезпечує хмарні послуги. Коли дані клієнтів піддаються шифруванню, виконуються операції із створення і зберігання резервних копій таких даних, тоді загрози безпеці даних клієнтів може і не трапитися ніколи;

– такі обчислення є дорогими.

1.3 Технологія кордонних обчислень

1.3.1 Концепція кордонної аналітики Edge Analytics

Кордонні чи периферійні обчислення (Edge computing) було створено з метою усунення недоліків власне хмарних обчислень [2]. В основі концепції кордонної аналітики Edge Analytics знаходиться збирання, обробка та аналіз даних на кордонних пристроях мережі, іншими словами безпосередньо близько до джерела даних.

Edge computing є переходом від хмарних обчислень властиво централізованих до, власне, периферійних чи кордонних обчислень. Самі ж кордонні обчислення є комплексом обчислювальних ресурсів (як то ПЗ, сервери та мережеві під'єднання), котрі розгортаються по периметру організацій. Отож інформація за своїм максимумом опрацьовується на локальних машинах, а хмарні властиво ж ресурси можна використовувати, наприклад, як сховище результатів виконаних розрахунків (як варіант, для збереження вже створених раніше результатів та звітів).

Головна функція кордонних обчислень – це є плавне об'єднання хмарних ресурсів та обчислень з кордонними пристроями та двосторонній обмін інформацією (рис. 1.1).

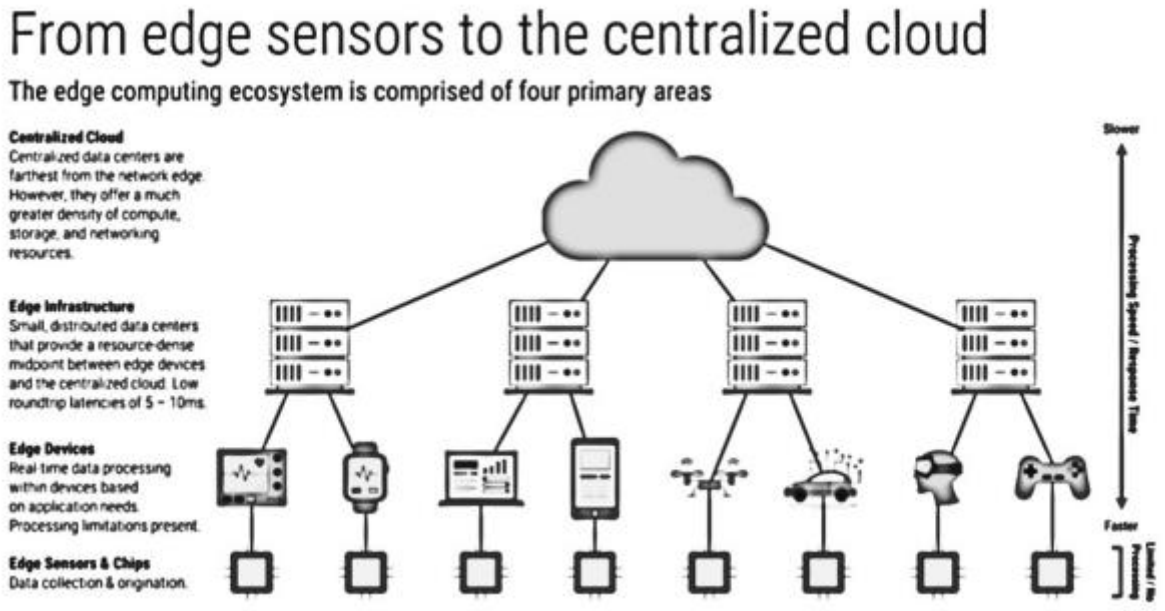


Рисунок 1.1 – Схема реалізації кордонних обчислень [2]

Рішення, яке використовує кордонні обчислення, надає можливість обробляти та здійснювати онлайн аналіз ключових даних поблизу. «Периферія мережі» містить у собі модулі прийняття рішень та модулі тимчасового збереження даних, котрі в перспективі не використовуватимуться, тобто такі дані є неістотними.

1.3.2 Основні переваги кордонних обчислень

До визначених переваг периферійних обчислень (рис. 1.2) можна віднести такі:

– мінімізація проблем із зв'язком;
– зниження затримок;
– забезпечення конфіденційності суттєвої інформації.

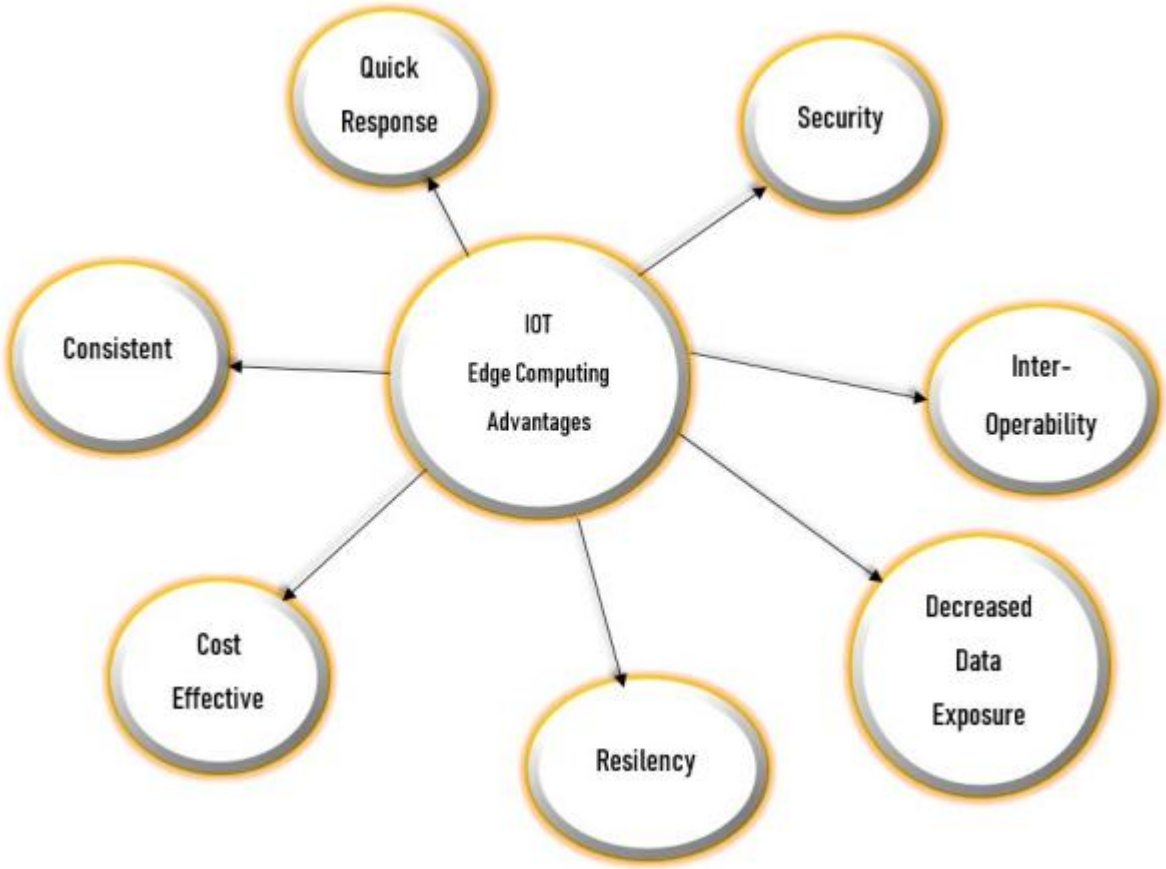


Рисунок 1.2 – Переваги кордонних обчислень

Для першого моменту при застосуванні концепції кордонних обчислень абсолютна більшість обчислень виконуються на локальних машинах. Це і є

гарантією того, що роботу не буде перервано у випадках поганого власне Інтернет-з'єднання. Якраз це і є особливо важливим у місцях, де немає постійного та стабільного мережного підключення.

По-друге, на сьогоднішній день достатньо багато осіб застосовують хмарні обчислення, надсилаючи у «хмару» значні об’єми інформації для наступного її опрацювання. Саме тому і здатні траплятися суттєві затримки при очікуванні на результати хмарного сервісу, це здатне погано вплинути на функціонування організацій.

По-третє, всі конфіденційні дані варто невідкладно опрацьовувати «біля», а до хмарного сервісу надсилати тільки ті дані, котрі узгоджуються із політикою конфіденційності, для уникнення витоку суттєвої інформації.

РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА

2.1 Засоби для аналізу відеопотоку

2.1.1 Бібліотека OpenCV

Є бібліотекою алгоритмів комп'ютерного зору та МН, створена на C/C++. Дає змогу проводити обробку, класифікацію та аналіз зображень із застосуванням різноманітних чисельних алгоритмів із відкритим вихідним кодом. Особливо популярна для використання у мові програмування Python, але також застосовується і у C, C++, Java.

Бібліотека містить велику сукупність алгоритмів, котрі призначені для вирішення різних задач, як то:

- розпізнавання рукописного та друкованого тексту;
- видалення спотворень зображення;
- спостереження за переміщенням об'єкта;
- розпізнавання різних рухів, жестів;
- розпізнавання об'єктів у відеоданих;
- встановлення подібності та форми об'єктів

2.1.2 Метод Віоли-Джонса

Для відшукання об'єктів із застосуванням їх ключових ознак, застосовується метод П. Віоли та М. Джонса. Базою підходу є 4 складові:

- функції Хаара (прості прямокутні функції);
- інтегральне представлення зображення;
- AdaBoost;
- каскадний класифікатор.

Метод застосовує підхід «ковзного вікна» – рамки, розмір якої менше, ніж розмір вихідного зображення, котра переміщається по зображенню із визначеним кроком і детектує присутність осіб на зображенні через каскад "слабких" класифікаторів. Дана техніка ефективно застосовується в сфері комп'ютерного зору з метою виявлення різних об'єктів на різноманітних

зображеннях [4].

Характерні особливості, котрі використані в даному методі, спираються на вейвлет Хаара– прямокутні хвилі з однаковою довжиною [1]. Для двовимірного простору такими прямокутними хвилями є пара сусідніх прямокутників (світлого та темного відтінків), котрі називаються примітивами Хаара чи маскою (рис. 2.1).

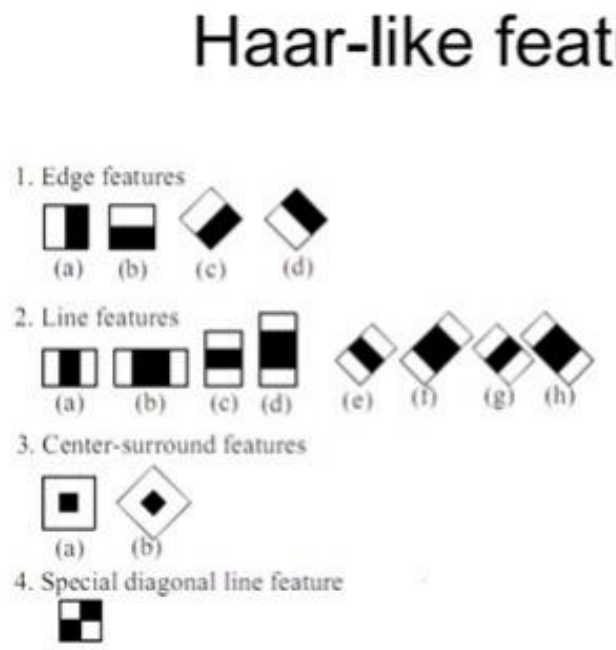


Рисунок 2.1 – Примітиви Хаара [4]

Присутність функції ймовірно визначити при допомозі примітивів Хаара. При накладенні примітивів на вихідне зображення можна одержати якесь числове значення - різницю суми яскравості пікселів, котрі попали у білу частину примітиву, та суми тих пікселів, котрі попали до їх чорної частини (2.1):

$$FHaar = Bwhite - Bblack. \quad (2.1)$$

Тут Bwhite і є тією сумою яскравості пікселів, котрі потрапили в білу область примітиву, Bblack - у чорну область відповідно.

У подальшому отримане число порівнюють із граничним значенням, яке попередньо встановлюється на етапі навчання: за умови, що це число перевищує

граничне значення, то функція Хаара наявна.

Технологія інтегрального подання зображення застосовується для ефективного визначення великої кількості функцій Хаара. Яскравість будь-якого пікселя і є, властиво, його вагою. Інтегральна ж величина кожного пікселя є сумою величин яскравості тих пікселів, котрі містяться вище і лівіше біжучого пікселя, і вага пікселя, що розглядається [6]. У комп'ютерних програмах для розрахунку інтегрального значення пікселя можна використати формулу (2.2):

$$L(x,y)=\sum_{i=0,j=0}^{i\le x,j\le y} I(i,j). \qquad (2.2)$$

Тут *I (i,j)* - яскравість пікселя із координатами *(i,j)* вихідного зображення.

Інтегральне подання зображення є матрицею, розміри котрої збігаються з розмірами зображення, кожен елемент котрої відповідно є також інтегральним значенням відповідного пікселя на зображенні. Якщо починати обхід зображення від верхнього кута зліва і йти праворуч і вниз, зображення може бути швидко «проінтегрованим».

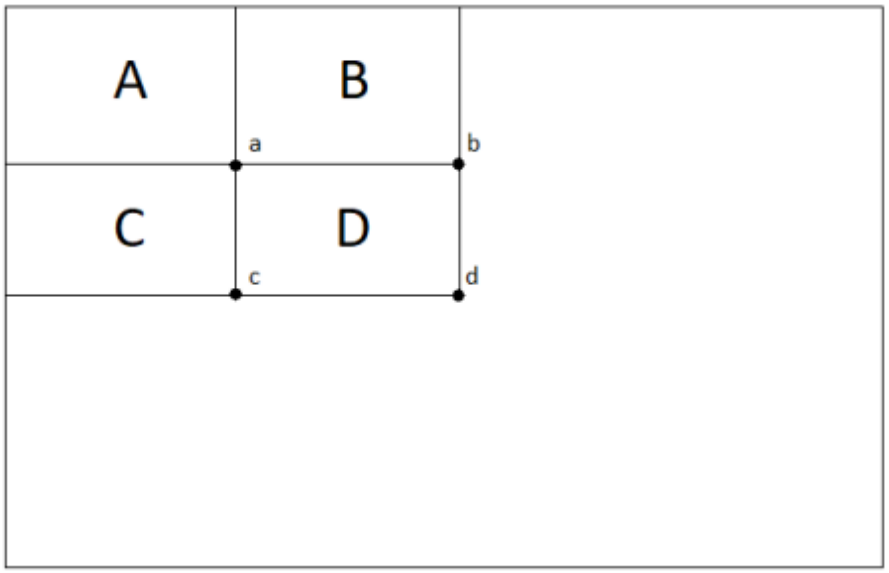


Рисунок 2.2 – Розглядувана область *D*

Наприклад, $(x_a, y_a), (x_d, y_d), (x_c, y_c), (x_b, y_b)$ - інтегральні значення для точок A, D, C, B відповідно, тоді для подання на рис. 2.2 області D інтегральне значення можна підрахувати за наступною формулою (2.3):

$$I(D) = (x_a, y_a) + (x_d, y_d) - (x_c, y_c) - (x_b, y_b). \quad (2.3)$$

Призначення AdaBoost у методі Віоли-Джонса – це визначення порогових значень та вибору конкретних функцій Хаара, що використовуються. Основною метою AdaBoost є формування «сильного» класифікатора через комбінації «слабких». Під власне «слабким» класифікатором мається на увазі той класифікатор, котрий робить у роботі достатньо багато помилок і видає правильний результат не частіше, як випадкове угадування, тобто він не здатен чітко і точно дати відповідь. Головна ідея такого процесі в тому, що за умови наявності сукупності «слабких» класифікаторів (важливо, що кожен окремий класифікатор робить помилку незалежно від інших і одержує відповідь із якоюсь помилкою), тоді комбінація усіх таких класифікаторів має видати меншу імовірність помилки у порівнянні із результатом окремого властиво класифікатора множини. Метод AdaBoost вибирає класифікатори та кожному класифікатору ставить у відповідність якесь число – вагу. Лінійна ж така комбінація класифікаторів, попередньо помножених на власну вагу, - це «сильний» класифікатор, який можна представити формулою (2.4):

$$f(x) = \sum_{k=1}^{m} \alpha_k * r_k(x). \quad (2.4)$$

Тут $f(x)$ - «сильний» класифікатор, $r_k(x)$ - k-ий «слабкий» класифікатор, α_k - вага k-того «слабкого» класифікатора, x - деякий шаблон для класифікації.

Властиво каскадний класифікатор є моделлю, котра містить послідовні рівні із фільтрами. Такий фільтр є класифікатором AdaBoost із малою кількістю «слабких» класифікаторів. Фільтри розташовуються за рівнями, на початкових

рівнях містяться найважчі такі фільтри для відсіювання областей, котрі містять аналізований об'єкт. Варто відмітити, що зображення проходить один за одним усі фільтри. Відповідно на кожному рівні фільтр здатен чи пропустити чи не пропустити якусь частину зображення. Коли фільтр не пропускає якусь частину зображення, вонj розглядається як «не обличчя» і надалі не опрацьовується. Отож, пропускаючи все зображення через послідовність таких фільтрів, ймовірно чи розпізнати всі обличчя на фотографіях, або зробити висновок про те, що на фотографії обличчя відсутні. Приклад ймовірної фільтрації області зображення наведено на рис. 2.3.

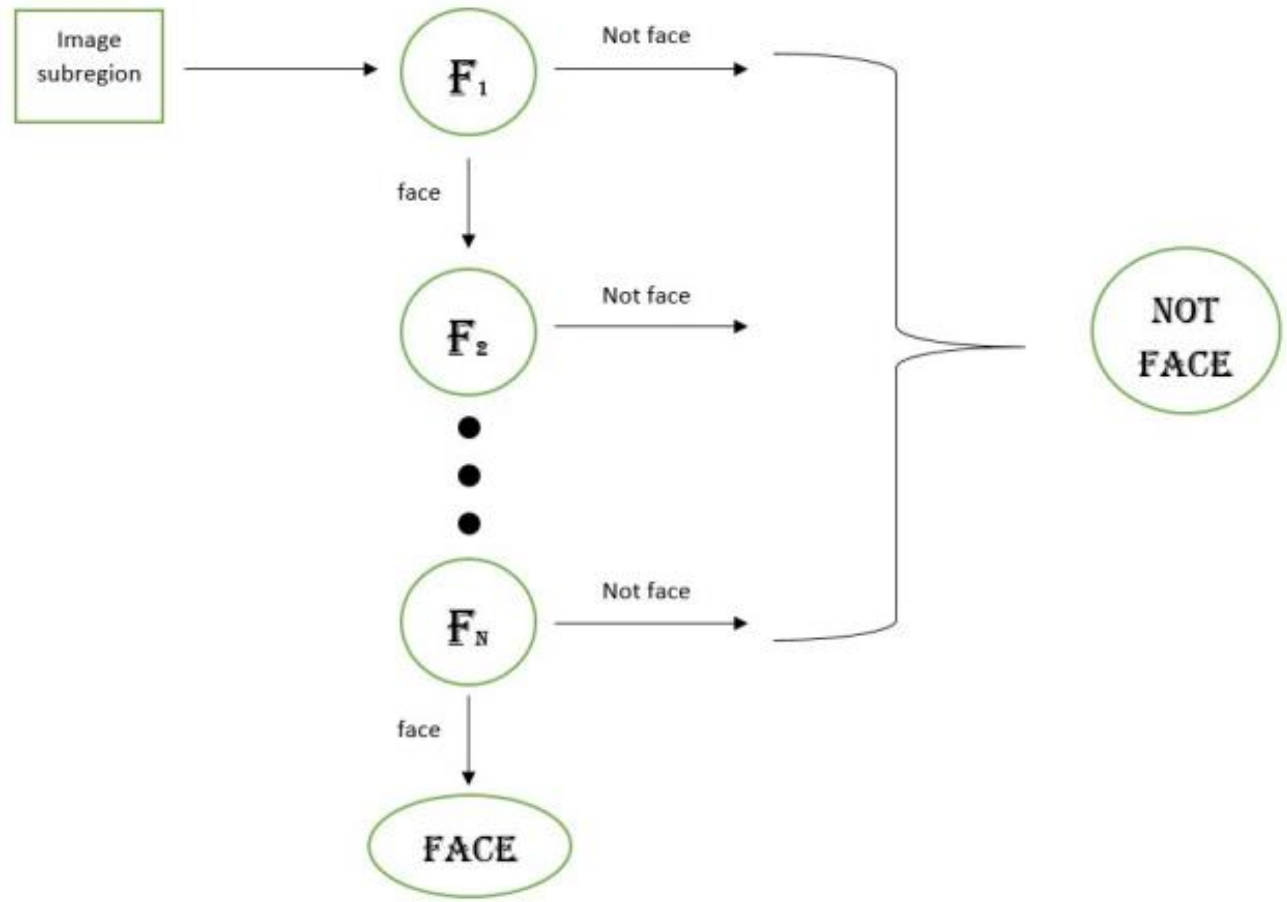


Рисунок 2.3 – Приклад можливої фільтрації області зображення

Метод Віоли-Джонса є достатньо ефективним методом розпізнавання обличь на зображеннях, але цей метод має свої недоліки:

– необхідне перенавчання завдяки високому рівню шуму даних;

– велика навчальна вибірка та тривале навчання.

2.2 Алгоритми та засоби ідентифікації обличь на відеоданих

2.2.1 Azure Cognitive Services

Використовуючи Azure Cognitive Services, отримуємо доступ до засобів ШІ. Варто зауважити, що для якісної роботи із такими засобами не потрібно бути професіоналом у галузі МН. Завдяки зручному API, котрий забезпечує Cognitive Services, можна додати до програм різні функціональності: «бачити», «чути», «спілкуватися», «розуміти інформацію» та «швидко приймати рішення» (рис. 2.4):

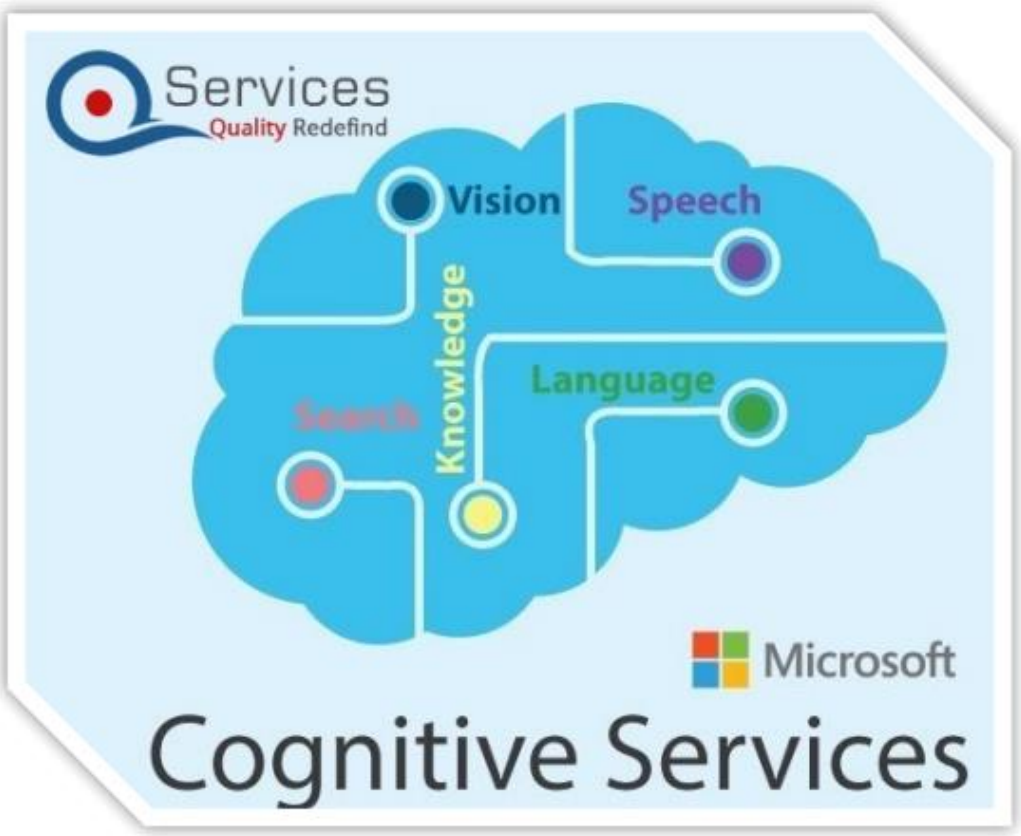


Рисунок 2.4 – Azure Cognitive Services [3]

Перевагами служби є:

– використання засобів ШІ при розробці різних застосунків;

– достатньо швидке розгортання служби із застосуванням контейнерів у

хмарному середовищі та локально;

– спрощена робота із засобами ШІ, які не потебують спеціальних знань.

2.2.2 API Ідентифікації обличь

Azure Face є хмарною служба Azure, котра дає змогу на зображеннях детектувати та аналізувати обличчя. До неї входить API ідентифікації осіб, що дає змогу за наявною БД знайомих осіб ідентифікувати особу на знімку [15].

Створивши заздалегідь БД із знайомими обличчями та навчивши на ній сервіс, у подальшому можна на знімках здійснювати ідентифікацію особи через порівняння об'єкта – обличчя з обличчями із БД. На рис. 2.5 наведено приклад БД з обличчями "known people".

Рисунок 2.5 – БД "known people"

Варто зазначити, що у складі кожної групи може бути до 1 мільйона об'єктів, які відповідають різним власне людям. Зі свого боку, для кожного об'єкта, який відповідає визначеній людині, ймовірно зареєструвати до 248

обличь. Якщо при порівнянні з БД обличчя було упізнано, то повертається об'єкт, котрий відповідає цій людині.

2.2.3 HOG

Є одним із найбільш популярних методів розпізнавання об'єктів, котрий ґрунтується на дескрипторах особливих точок і працює досить швидко та надійно. Такі HOG -дескриптори застосовуються для рішення різних задач розпізнавання: пішоходів на нерухомих зображеннях, обличь та різних частин тіла, жестів рук для перекладу на мову жестів.

HOG - дескриптори допомагають спрощувати зображення, витягуючи з нього лише найціннішу інформацію [11].

Вихідне зображення, як правило, виконують чорно-білим оскільки інформація про колір жодним чином не бере участь у детектуванні обличь. Після цього береться до уваги оточення кожного пікселя. Ця дія потрібна, щоб визначити, наскільки темним є піксель щодо його сусідів [8]. Після проведення такої операції для кожного пікселя знімка, потім пікселі здатні бути заміненими на стрілку, що показано на рис. 2.6.

Рисунок 2.6 – Вихідне зображення та його HOG - подання

Така стрілка є градієнтом, котрий проходить від світлих пікселів до темних, тобто напрямлений у бік затемнення знімку. Саме так одержують

шаблони.

За умови збереження в пам'яті машини градієнта для кожного пікселя зображення, це займатиме надто багато її пам'яті. З метою оптимізації пам'яті вихідне зображення можна розбити на комірки розміром 16 на 16 пікселів кожна, яка, зі свого боку, з яких може бути поділена на ще більш дрібні комірки 8 на 8 пікселів. У такій кожній маленькій комірці кожен власне піксель і замінюється стрілкою та проводиться підрахунок стрілок, котрі мають однаковий напрямок. Далі комірка із розміром 8 на 8 пікселів замінюється стрілкою, котра напрямлена туди ж, куди і більшість маленьких градієнтів. Точно такі самі дії проводяться для блоків із розміром 16 на 16 пікселів.

Класифікацію HOG-дескрипторів можна проводити за допомогою SVM. Його ідея: за допомогою роздільної гіперплощини відокремити вектори ознак об'єкта та вектори ознак фону [7]. На рис. 2.7 показано приклад розділу гіперплощиною.

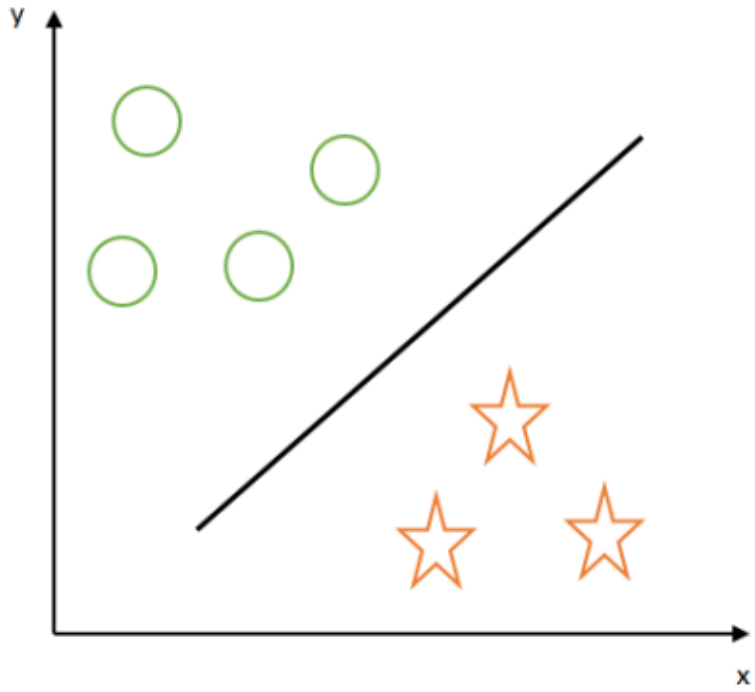


Рисунок 2.7 – Приклад гіперплощини, яка відділяє кола від зірочок

Результат роботи SVM - два образи об'єкта з негативними та позитивними вагами опорних векторів. Тут позитивність значить належність ознаки об'єкту (у

роботі об'єкт - обличчя), а негативніть вказує, що ознака належить фону.

Власне на вихідному шаблоні достатньо добре проглядається базова структура обличчя, що дозволяє ефективно виявляти обличчя нових зображеннях, застосовуючи уже знайомі шаблони.

2.2.4 CNN

Є нейромережевою архітектурою, котра використовується для класифікації зображень та графічних образів (рис. 2.8):

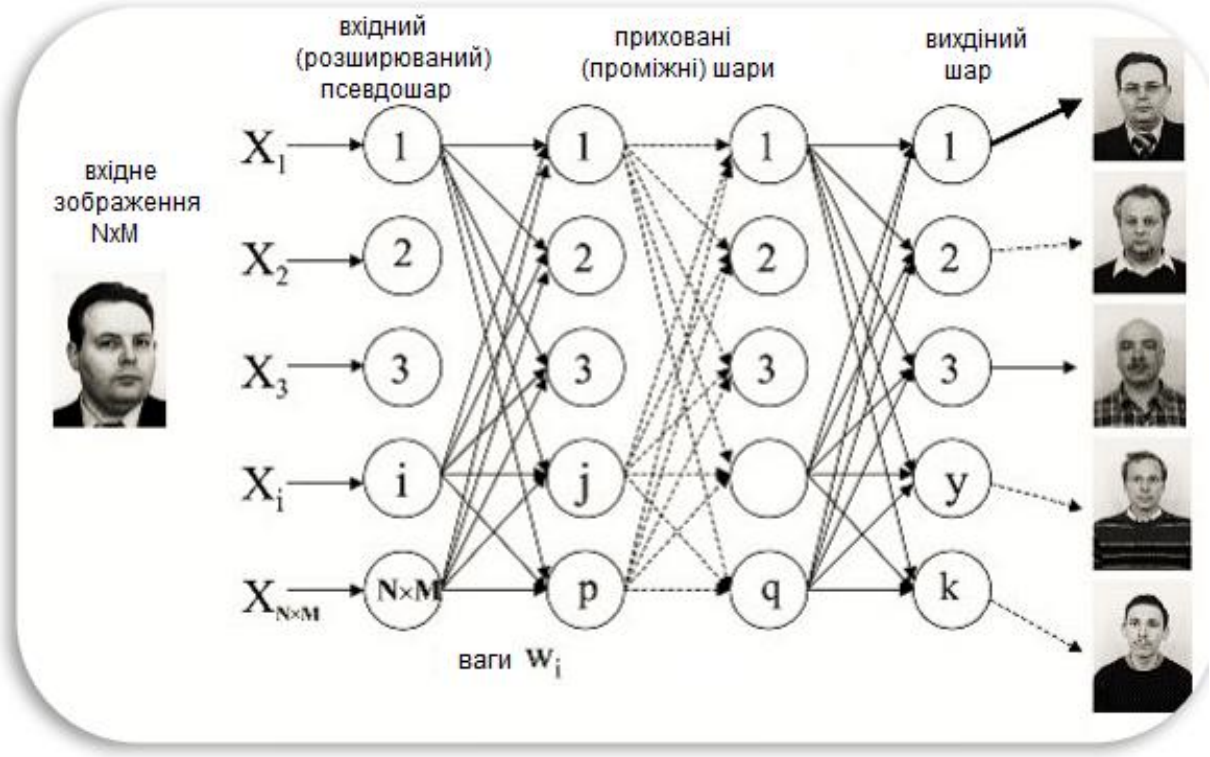


Рисунок 2.8 – CNN для класифікації зображень [13]

CNN популярна в галузі МН та комп'ютерного зору для рішення різних задач.

Навчаються CNN на зображеннях із застосуванням шаблонів (паттернів) для класифікації, ось чому не потрібно отримувати ознаки вручну. Один із найбільш поширених методів навчання є властиво метод зворотного поширення помилки. Власне назву CNN отримала як наслідок застосування спеціальної операції – згортки, є операцією над матрицями A (розміру n_x на n_y) і B (розміру

m_x на m_y), результатом якої є матриця C (розміру m_x - n_x + *1* на m_y - n_y + 1), елементи якої обчислюється за формулою (2.5):

$$C_{k,m}=\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} x_{i+k,\,j+m}*\omega_{ij}+\omega_0, k=\overline{1,m_x-n_x+1}, m=\overline{1,m_y-n_y+1}. \quad (2.5)$$

Тут $x_{i,j}(i=\overline{1,n_x}, j=\overline{1,n_y})$ - елементи матриці B, ω_{ij} $(i=\overline{1,m_x}, j=\overline{1,m_y})$ – елементи матриці A, ω_0 - якесь число (bias або нейрон зміщення).

На рис. 2.9 показано ймовірний приклад виконання операції згортки.

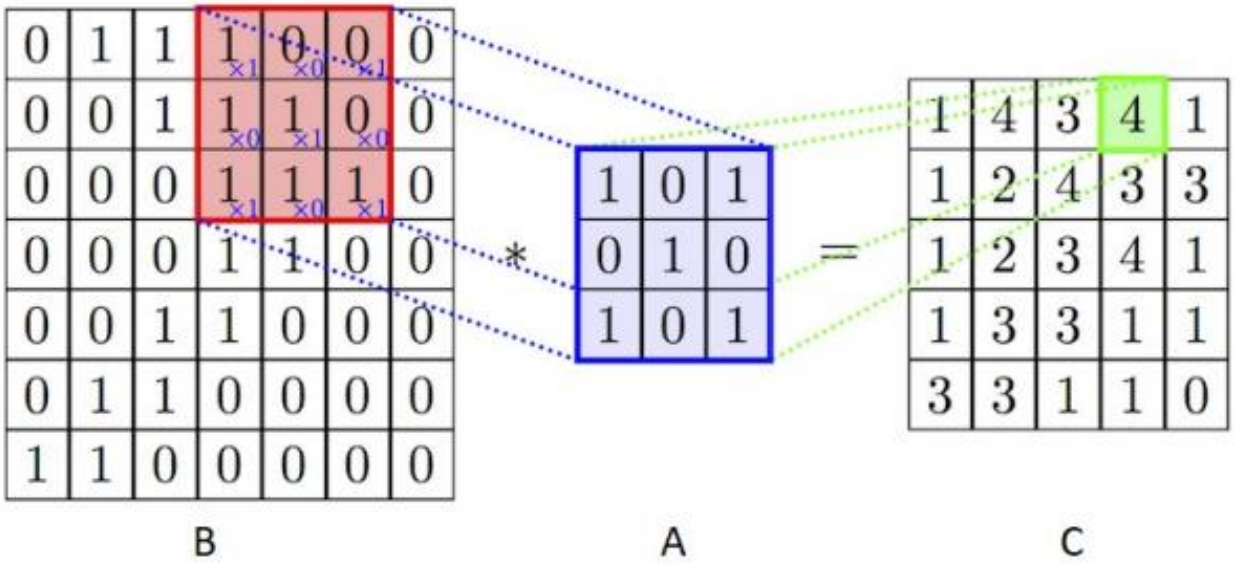


Рисунок 2.9 – Приклад згортки

Тут матриця B носить назву зображення, тоді як матриця A - фільтра. При проведенні операції згортки ядро "пересувається" за зображенням.

Свою популярність CNN набули за рахунок трьох важливих факторів:

– усунення ручного вилучення ознак;

– достатньо хороша точність результатів;

– можливість перенавчання CNN, завдяки чому можна використовувати нейронні мережі розпізнавання різноманітних графічних об'єктів.

У відповідності до структури використання можна сформувати свою

нейронну мережу або використовувати готову, завчасно навчивши її на певній вибірці. Зараз нейромережі можна використовувати в медицині для візуального детектування ракових клітин, в безпілотних авто для детектування об'єктів та розпізнавання дорожніх знаків, для ідентифікації та класифікації звуку, тексту, відео та зображень [12].

CNN включає вхідний і вихідний шари, а також множину прихованих шарів між ними. Такі приховані шари застосовуються для вивчення характеристик вихідного зображення, серед яких виділяють три найбільш поширені шари: згортковий чи згортка, шар активації чи ReLu та pooling (субдискретизуючий шар).

Вхідне зображення у згорткових шарах проходить через каскад фільтрів, кожний із них активує чи визначає якісь характеристики зображень (як приклад, вертикальні чи горизонтальні лінії). Для випадків із зображеннями в градаціях сірого кольору фільтр буде один, проте при багатоканальному зображенні фільтр вже міститиме набір ядер, кількість яких збігається із кількістю каналів зображення. Властиво скалярний результат згортки попадає на ReLu, іншими словами на якусь нелінійну функцію. Функція активації зі свого боку відсікає негативні значення та зберігає позитивні [13]. Субдискретизуючий шар «ущільнює» або зменшує зображення, здійснюючи деяке нелінійне перетворення та зберігаючи лише найціннішу інформацію. Як нелінійне перетворення зображення можна використовувати метод maxpooling, minpooling або averagepooling. Відобразити операцію maxpooling можна з прикладу зображення розміром 128 на 128 пікселів. Визначаються вікна, котрі не перетинаються, деякого розміру, наприклад, 2 на 2 пікселі, які будуть «ковзати» по карті ознак, котра була одержана використаннчям операції згортки до зображення, та у кожному такому вікні визначається своє максимальне значення. Результат використання maxpooling наведено на рис. 2.10.

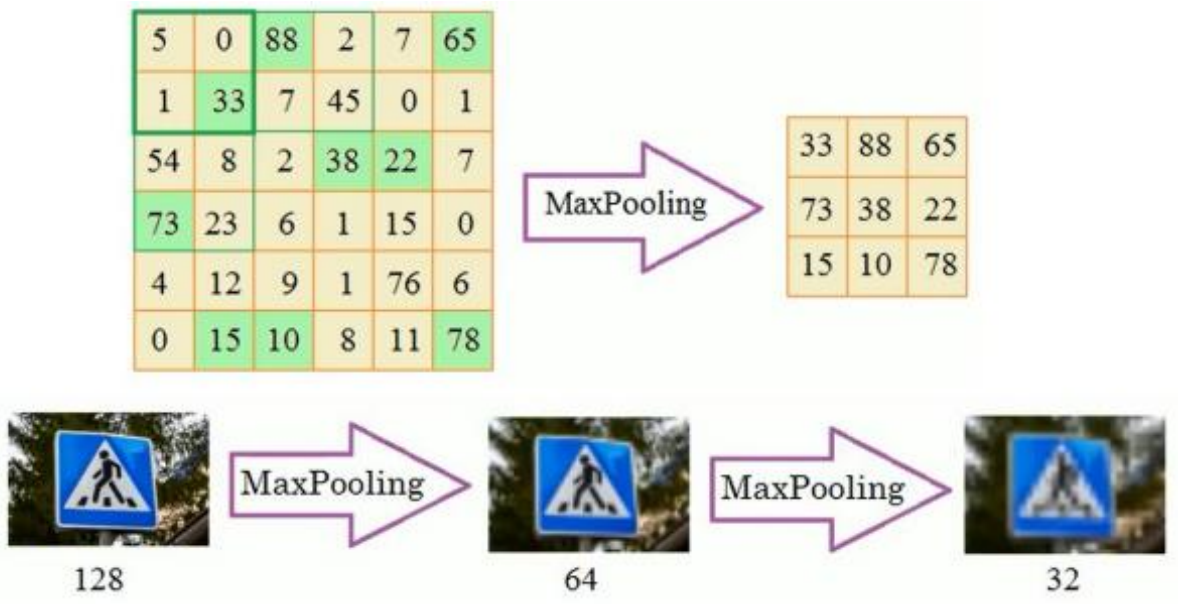


Рисунок 2.10 – Результат застосування операції maxpooling

Дані операції мають повторення на багатьох шарах для визначення різних параметрів зображення. Власне вхідне зображення для наступного шару є результатом використання фільтрів до властиво вхідного зображення на поточному шарі. На рис. 2.11 наведено приклад нейромережі із достатньо значним числом згорткових шарів [12].

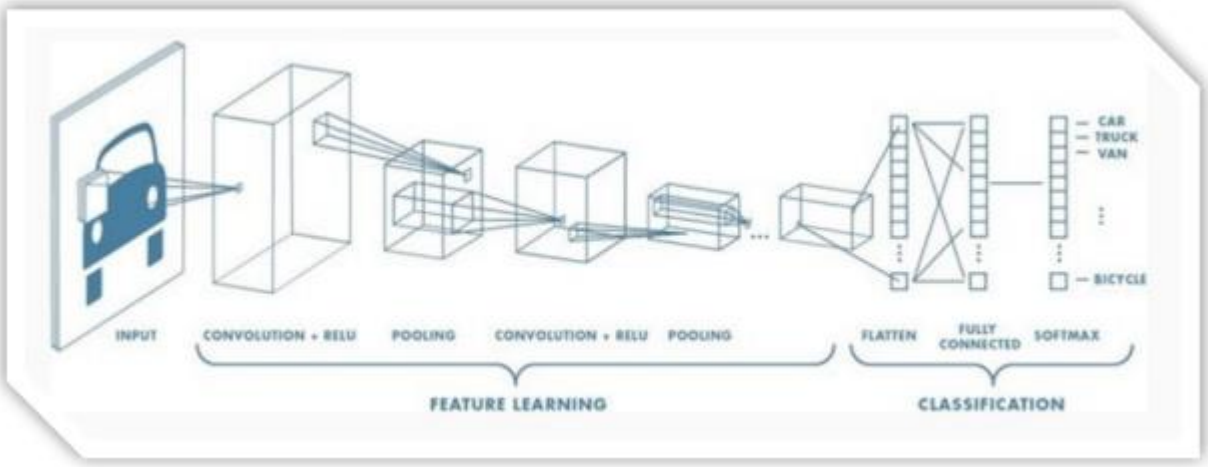


Рисунок 2.11 – Приклад нейронної мережі з великою кількістю згорткових шарів

Останні шари CNN призначені для класифікації. Результатом нейромережі є вектор, складовими котрого є можливість приналежності зображення тому чи іншому класу. Розмір такого вектора дорівнює числу класів, до котрих може

належати вхідне зображення.

CNN навчаються на великій навчальній вибірці. При роботі з громіздкими обсягами даних можна всі обчислення проводити на графічних процесорах, щоб прискорити процес як навчання нейронної мережі, так і обробки вхідних зображень.

2.3 Вибір інструментів реалізації системи

2.3.1 Мови програмування Java та Python

Java мова, що є інтерпретованою і строго типізованою.

Поміж багатьох можливостей Java варто виокремити такі:

– наявність збирача сміття, що дає змогу вивільнювати пам'ять, знищуючи при цьому об'єкти, котрі будуть не потрібні додатками;

– зручна обробка виняткових ситуацій;

– java collections;

– засоби фільтрації та форматування вводу/виводу;

– lambda - expressions, що дають змогу зменшити кількість коду та покращити його читабельність (з версії 1.8);

– підтримання узагальнень (з версії 1.5);

– підтримання багатопоточності;

– присутність засобів (класи), при допомозі яких можна проводити HTTP -запити та обробляти відповідні відповіді;

– присутність інструментів, котрі дають змогу просто створювати мережеві програми.

Python - інтерпретована мова, котра перебуває на стадії активного розвитку, напрямлена на вирішення різнопланових задач.

Python підтримує об'єктно-орієнтоване, функціональне та узагальнене програмування. Переваги Python – наявність механізму опрацювання винятків, генератори та ітератори, декоратори та підтримка лямбда – виразів, гнучкість та масштабованість. Можливості мови дозволяють поділ програми на окремі модулі, котрі ймовірно поєднати у пакети. Python прекрасно підходить для

створення мобільних застосунків різного спрямування, web -сайтів, а також нею користуються в МН.

2.3.2 Середовище розробки та редактор коду

Intellij IDEA від компанії JetBrains. Основне його призначення – створення різноманітного ПЗ. IDE підтримує системи контролю версій (як то Git чи SVN), доступ до різних БД (у версії Ultimate), інструменти збирання (наприклад, Maven), велике число різноманітних фреймворків, плагінів, при допомозі яких є можливість додавати до IDE нові функціональні можливості. Різні плагіни можливо або завантажувати через веб-сайт, чи завдяки вбудованій в IDE функції пошуку та інсталяції плагінів.

На підставі аналізу коду IDE пропонує користувачу потужні можливості для виконання рефакторингу коду:

- розумне автодоповнення;
- автодоповнення ланцюжків викликів та статичних членів;
- пошук повторів;
- швидкі виправлення та мовні вставки;
- рефакторинги для різних мов.

Необхідно також зазначити, що Intellij IDEA дає змогу програмувати різними мовами.

Visual Studio Code від Microsoft є безплатним редактором вихідного коду для ОС Windows, macOS та Linux. Цей продукт служить для створення різноманітних веб - та «хмарних» застосунків. Редактор має підсвітку синтаксису, можливість автозавершення коду (IntelliSense), налагодження та навігацію за кодом більшості мов програмування, Git та багато іншого. У редакторі є вмонтована консоль, в котрій користувач здатен побачити результат функціонування програми чи повідомлення щодо помилки.

2.3.3 Azure Functions Core Tools

Це програми командного рядка, котрі призначені для впроведення локальної розробки та виконання функцій Azure, а також, за необхідності,

публікування їх у Azure.

Основне призначення даного інструменту полягає в наступному:

– створення всіх необхідних файлів та папок для локальної розробки функцій Azure;

– локальне тестування та налагодження функцій;

– публікація функцій у хмарне сховище (Microsoft Azure).

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Компоненти Microsoft Azure

3.1.1 Хмарне сховище даних

Azure є комплексом служб хмарних обчислень від Microsoft, котрий дає змогу організаціям вирішувати різного роду бізнес-завдання. Хмарна платформа дозволяє малим та ще не досвідченим підприємствам, стартуючи із незначних витрат, швидко, якісно і зручно проводити масштабування інфраструктури відповідно до появи нових клієнтів та замовлень [3].

Основні переваги Microsoft Azure:

– проста та зручна співпраця із гібридною «хмарою»;
– створення різноманітних рішень;
– безпека;
– тестування нових версій ПЗ.

Хмарна платформа дає змогу вільного створення та розгортання різноманітних застосунків, керування ними у мережі Інтернет із застоуванням інструментів-фаворитів. Azure дозволяє використовувати всі локальні можливості, функціонувати у "хмарі" чи на кордонних пристроях.

За рахунок підтримки всіх застосунків та платформ і можливості використовувати відкритий код в Azure наявна можливість створення рішення у зручний для нас спосіб та проводити їх розгортання у яких-небудь середовищах.

Хмарна платформа володіє комбінованим захистом, який надає команда експертів, що гарантує безпеку та збереження даних клієнтів.

При допомозі Azure можна проводити тестування нових версій ПЗ, нам не потрібно проводити заміну локального обладнання. Наприклад, для тестування програми з Microsoft SQL Server 2014 необхідно створити екземпляр застосунку, виконати копію наших служб, попередньо підключену до нової БД, та провести аналізи результатів. В цьому випадку не треба доповнювати своє локальне обладнання.

Переваги Azure над AWS: конкурентні ціни; використання потенціалу

вихідного коду повністю; використання будь-яких ОС з відкритим кодом, мов та засобів.

За допомогою Azure було зроблено найбільший внесок у GitHub у 2017 році, на додачу вона є єдиним хмарним середовищем із вбудованою підтримкою Red Hat.

Класифікація даних. Кожен набір даних має свої вимоги, тому постає завдання пошуку оптимального рішення для забезпечення збереження даних. Основними факторами, котрі повинні постійно братися до уваги при відшуканні оптимального рішення, є: правильність класифікації даних, як дані застосовуватимуться вподальшому і яким чином забезпечити найкращу продуктивність програми.

Використовуються дані трьох типів: повністю структуровані; структуровані частково; неструктуровані.

Властиво розуміння різниці між ними та правильна класифікація – ключові фактори при пошуку найкращого рішення з метою збереження даних.

Перший тип даних має чіткий розподіл за рядками та стовпцями у таблицях, має хорошу організованість.

Дані другого типу добре впорядковані, мають певні властивості та значення, проте дозволяють деяку змінність (як приклад, Yaml, JSON, XML).

Для даних третього типу відсутня певна модель, не має змоги їх розподілити за таблицями (наприклад, BLOB -об'єкти, текстові файли).

Для виконуваної роботи основні дані - це зображення. Вони є неструктурованими.

Варто представити ключові дії над даними:

– потрібне лише отримання по ідентифікатору;

– клієнтам необхідно забезпечити значну кількість операцій читання із мінімальною затримкою;

– операція оновлення буде виконуватися вкрай рідко, на додачу для неї допустима вища затримка порівняно з операцією зчитування.

Сховище BLOB -об'єктів Azure - відмінний варіант для виконання цього завдання, воно забезпечує зберігання будь-яких файлів, поміж іншими відео та

фотографій. Також підтримується кешування вмісту, що часто використовується, зберігаючи його на кордонних серверах, а при допомозі Azure CDN є можливість скорочення затримки при подачі даних клієнтам. Використовуючи даний підхід, зображення можна рухати між різними рівнями (як то архівним, «гарячим» і «холодним») з метою збільшення власне пропускної спроможності та зниження витрат для найбільш широко використовуваних відео.

На додачу використовується ще два рішення –Служба додатків Azure або БД. Службу додатків Azure можна використовувати, якщо зображень буде небагато. Але при великій кількості даних краще використовувати сховище BLOB -об'єктів Azure разом із Azure CDN. Зберігання всієї інформації у БД - неоптимальне рішення через обсяг даних і продуктивності.

Azure Storage. У платформі Azure збереження даних можливе у різний спосіб. "Хмара" надає різні БД для збереження інформації, а саме: Azure SQL Server, Azure Cosmos DB та сховище таблиць Azure. Є кілька варіантів зберігання та надсилання повідомлень, наприклад, черги Azure та центри подій. За допомогою таких рішень, як файли Azure (Служба файлів Azure) та BLOB - об'єкти Azure, можна проводити збереження незв'язаних файлів. В Azure Storage (служба сховища Azure) містяться такі рішення [3]: BLOB -об'єкти Azure, служба файлів Azure, черги Azure та таблиці Azure (рис. 3.1).

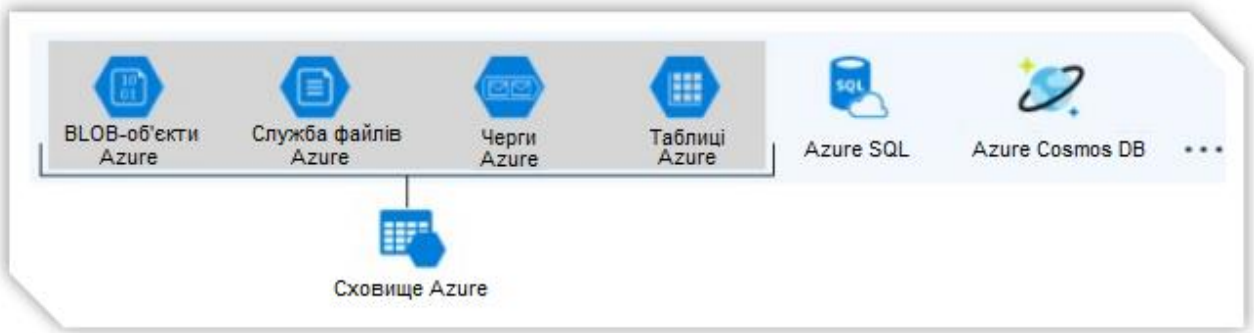


Рисунок 3.1 – Сховище Azure

ОЗЗ (Storage account) є контейнер, котрий проводить групування набору служб для служби Azure Storage (рис. 3.2). До його складу можуть входити лише служби даних із сховища Azure (BLOB -об'єкти Azure, файли Azure, черги Azure

та таблиці Azure). Властиво таке об'єднання служб даних в ОЗЗ дає змогу керувати ними як єдиним цілим (групою). Спеціалізовані атрибути, котрі задаються під час створення або редагування ОЗЗ, будуть застосовуватись до всього вмісту ОЗЗ. Власне всі служби, котрі містяться ОЗЗ, видаляться під час видалення відповідного ОЗЗ.

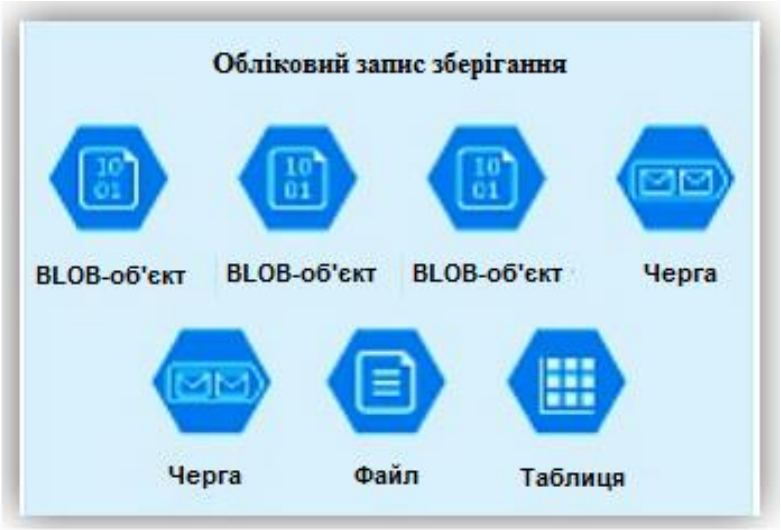


Рисунок 3.2 – ОЗЗ

ОЗЗ – ресурс Azure, який входить до групи ресурсів Azure (Azure Resource group). Служби даних Azure, такі як Azure Cosmos DB та Azure SQL, керують як незалежними ресурсами, такі служби не потрібно включати до ОЗЗ.

Azure Storage - сховище BLOB -об'єктів. Це найкраще оптимізоване рішення для збереження великих обсягів даних, котрі неструктоpовані, для прикладу текстові або двійкові дані.

Сховище BLOB -об'єктів ідеально підходить у таких випадках:

– потокова передача відео та звуку;
– зберігання інформації для обробки локальною чи хмарною службою;
– аварійне відновлення та архівація, зберігання резервних копій;
– збереження даних з метою забезпечення розподіленого доступу;
– миттєве обслуговування фото чи документів у браузері, включно із повними статичними веб-сайтами.

3.1.2 Безсерверні обчислення та функції Azure

Обчислення такого роду є функцією як послугою (Faas) чи хмарною мікрослужбою, котра доступна на запит. За рахунок того, що вся бізнес-логіка працює у форматі функцій, не потрібно готувати чи масштабувати інфраструктуру, при цьому її керування перебирає постачальник властиво хмарних служб.

У залежності від навантажень програми можуть масштабуватися вниз чи вгору. Існує кілька варіантів створення такої архітектури за допомогою компонентів Microsoft Azure. Варто виділити два найбільш популярні підходи - застосування Azure Logic Apps та Azure Functions (функцій Azure) (рис. 3.3):

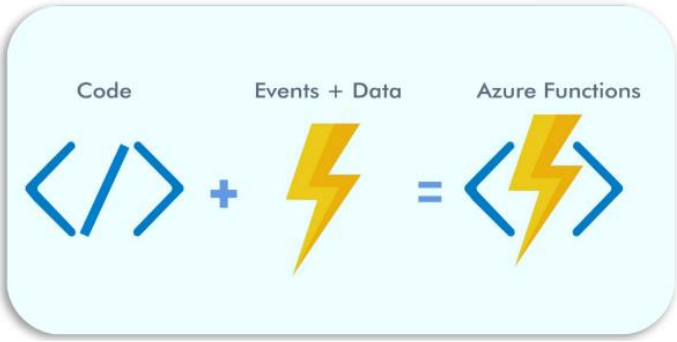


Рисунок 3.3 – Функції Azure

Azure Functions є безсерверною платформою обчислень, котра дає змогу втілювати бізнес-логіку, здатну виконуватися без попередньої підготовки самої інфраструктури. Код функції здатен бути написаним якою-небудь зручною мовою, в т.ч. F#, C#, JavaScript, Java, Python та PowerShell. Забезпечена підтримка диспетчерів пакетів NuGet та NPM, що дозволяє використовувати багато бібліотек у бізнес-логіці.

Плюси та мінуси рішення на базі безсерверних обчислень. Такі обчислення є чудовим варіантом для розміщення коду у хмарній інфраструктурі. Поміж основних плюсів даного підходу варто назвати:

– автоматичне масштабування та просте розгортання;
– відсутність надмірного виділення інфраструктури;
– логіка без відстеження стану;

– керування подіями;

– функції можна використовувати в традиційних обчислювальних середовищах.

При застосуванні Azure Functions, буде отримано автоматичне масштабування, при цьому не потрібно турбуватися про керування серверами. Потрібно проплатити кошти тільки за ті ресурси, що реально використовуються, а не властиво зарезервований час. Виконувати розгортання функції можна за допомогою інструментів (для прикладу, Visual Studio або Visual Studio Code) чи із застосуванням безперервного розгортання (за допомогою системи керування версіями).

Використання безсерверних обчислень допомагає розв’язати задачу надмірного виділення інфраструктури завдяки можливості автоматичного масштабування.

Примірники функцій можна формувати чи видаляти на запит. Функції без відстеження стану зручні для безсерверних обчислень, однак, якщо є потреба відстеження стану, його можна зберігати у службі сховища.

Azure Functions керуються подіями, тобто вони виконуються у відповідь на якусь подію (тригер), як варіант, відповідь на HTTP запит чи додавання повідомлення власне в чергу. Саме це вельми спрощує основу коду, даючи змогу оголошувати лише джерела даних (як то тригер чи вхідна прив'язка) та точку призначення (яка буде вихідною прив'язкою), у той же час не потрібно створювати код для опрацювання черг, BLOB -об'єктів та ін., потрібно тільки побудувати основну бізнес-логіку.

Власне функції і є ключовим компонентом безсерверних обчислень і, разом з тим, служать обчислювальною платформою з метою виконання коду різноманітного типу.

Але й такий підхід має свої недоліки: час та частоту виконання.

Час, за замовчуванням, становить 5 хв, але його можна і збільшити, проте не більше, ніж до 10 хв. За умови, що виконання функції може тривати більше 10 хв, вона може бути поміщеною на віртуальну машину. Час очікування властиво може обмежуватися і 2,5 хв, при умові, що хмарна служба

виконується за запитом HTTP. На додачу наявна можливість побудови стійких функцій для «оркестрування» виконання кількох функцій без обмеження часу очікування.

За масштабування через кожні 10 с може формуватися лише один екземпляр функції, проте ≤ 200 екземплярів загалом. Кожен екземпляр здатний обслуговувати кілька одночасних виконань, але варто зазначити, що різні тригери володіють різними вимогами до масштабування. Саме тому при застосуванні тригерів необхідно перш за все вивчити їх та ті обмеження, котрі пов'язані з ними.

Прив'язки Azure Functions. Є способом під’єднати до функції Azure дані та служби. Не потрібно писати код у тілі функцій для під’єднання до джерел даних та керування під’єднаннями тому, що прив'язки вміють перебувати у взаємодії із іншими службами. Власне код функції застосовує вхідні прив'язки як джерела даних та вихідні прив'язки властиво як сховище результатів. Для керування вхідними та вихідними даними кожної функції є можливість вказання нуля та більшої кількості прив'язок.

Тригер - різновид вхідної прив'язки, що дозволяє провести виконання функції.

Тригер BLOB -об'єктів. Є тригером (додаток-тригер), котрий дає змогу забезпечити виконання Azure Function при зміні вмісту сховища BLOB -об'єктів Azure (для прикладу, при поміщенні або зміні файлу до сховища). Для створення тригера BLOB -об'єктів потрібно задати ОЗЗ Azure та вказати розташування, яке тригер буде відстежувати (у цьому випадку контейнер із збереженими у ньому зображеннями).

БД Azure SQL. Є спеціальною інтелектуальною масштабованою службою реляційних БД, що входить до сімейства Azure SQL. Основні переваги Azure SQL:

- зручність;
- вартість;
- масштабованість;
- безпека;

– один сервер, багато БД.

При налаштуванні БД на локальному комп'ютері або віртуальній машині варто добре орієнтуватися у існуючих вимогах до наявного програмного та апаратного забезпечення. Нам також необхідно зрозуміти нові рекомендації щодо забезпечення безпеки та керування ОС, виконувати оновлення SQL Server, розв’язувати проблеми із збереженням даних та їх резервним копіюванням. За вибору Azure SQL всю цю роботу перебирає на себе власне хмарний сервіс, а нам залишається лише вибрати ім'я і вказати кілька параметрів для БД. Все це можна зробити із використанням веб-браузера чи відповідних скриптів, що дає змогу надзвичайно швидко та легко сформувати БД. Примірники БД можна видаляти чи додавати у скільки-небудь зручний час. Даний підхід дає змогу повністю сконцентрувати увагу властиво на розробці відмінної програми, не переймаючись налаштуванням ПЗ для БД.

Оскільки все управління БД «хмара» перебирає, то не потрібно купувати різне обладнання, підключати живлення та виконувати дії з обслуговування БД. Microsoft Azure пропонує кілька тарифних планів для БД Azure SQL, що дає змогу визначити якісний і грамотний баланс вартості та продуктивності. Варто зазначити, що можна вибрати і такий стартовий рівень витрат клієнтів, який не перевищує кількох доларів на місяць.

Хмарний сервіс власне і дозволяє коригувати розмір БД та продуктивність у будь-який зручний час для нас, фактично миттєво будувати резервні копії БД.

До складу Azure SQL входить і брандмауер, котрий дає змогу контролювати доступ до БД. Доступ матимуть лише спеціальні IP -адреси, яким довіряємо. Керувати БД (для прикладу, будувати сутності, виконвати запити) можна із застоуванням Visual Studio, SQL Server Management Studio, Visual Studio Code.

При створенні першої БД будується для неї логічний сервер, при допомозі якого є можливість виконувати адміністрування БД: керувати політиками безпеки та обліковими даними для входу, проводити редагування політик брандмауера. Такі правила можна редагувати і для будь-якої окремої БД,

розміщеної на логічному сервері.

3.2 Розробка системи

Перщ за все необхідно чітко поставати поетапні завдання:

– розробити систему збору, зберігання та зчитування відеоданих;

– розробити систему обробки відео за допомогою алгоритмів чи готових моделей МН.

Для втілення стартового етапу перш за все потрібно провести підготовку хмарного сховища для збереження зображень. З цією метою була розроблена ресурсна група, а вже у ній - ОЗЗ (побудувати все це можна напряму при допомозі порталу чи наперед написаних скриптів) (Додаток А). Як шаблон проектування було обрано шаблон Producer-Consumer (рис. 3.4).

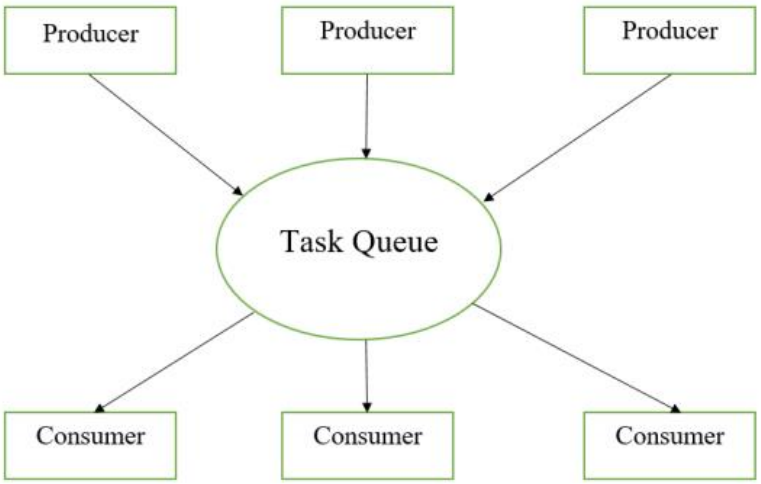


Рисунок 3.4 – Загальний вигляд шаблону проектування Producer-Consumer

Для його втілення застосовувалися два потоки: Producer (виробник) та Consumer (виконавець). Перший з них є якимося потоком, котрий генерить «завдання» і поміщає їх у чергу Queue (у роботі застосовувалася ArrayBlockingQueue – черга, котра реалізує класичний кільцевий буфер). Другий ж потік «добуває» завдання із черги, виконує та відправляє результати до визначеного приймача (кожен потік володіє своїм окремим класом). При допомозі потоку-виробника відбувається збір даних та будується черга запитів.

У наслідок можливостей бібліотеки OpenCV проводиться захват поточного зображення із камери, і, за умови його коректного захвату і відсутності жодних проблем іншого роду, в чергу надходить нове зображення для наступного опрацювання. Щойно керування передається потоку-виробнику, починається сканування та опрацювання черги повідомлень. За умови, що вона містить є нові завдання, вони опрацьовуються потоком-споживачем.

Сама ж обробка завдань проходить так: формується об'єкт класу CascadeClassifier, на вхід конструктора котрого на вхід поступає лише один параметр – стрічка. В цій ситуації стрічка - назва каскаду, котрий надалі застосовується для детектування осіб на зображенні (стрічка може містити назву одного з каскадів Хаара, так і назва lbp каскаду). У даного класу є вже готовий метод для аналізування зображення - detectMultiScale(photo, faceRects), на вхід котрого приймається два параметри: перший є вихідним зображенням, котре є об'єктом класу Mat, другий - об'єкт класу MatofRect. Отриманим результатом виконання цього методу буде матриця прямокутників, в якій кожен прямокутник є областю, в границях якої було розпізнано обличчя (у facerects буде збережено результат опрацювання). При умові, що така матриця непорожня, тоді на фото наявні особи, є можливість їх виділення та надсилання результату для збереження у попередньо підготовлене сховище, у іншому випадку треба перейти до опрацювання наступного повідомлення. Варто зауважити, що зв'язок із «хмарою» забезпечується із використанням класу AzureManager (Додаток Б). Для запису даних у хмару попередньо потрібно створити в ОЗЗ контейнер для збереження BLOB- об'єктів, в котрий кожне зображення буде записуватися як окремий файл (тут необхідно враховувати той факт, що кожен новий контейнер повинен мати унікальне ім'я, у іншому випадку з’явиться повідомлення щодо помилки), для вирішення такого роду проблем варто використати метод random. За такого підходу до збереження зовнішній доступ до даних отримати неможливо, що і дає змогу забезпечити захист та конфіденційність інформації.

В наступній частині роботи спроектовано та розроблено застосунок з ідентифікації обличь. Ця програма являє собою програму-тригер (BLOB -

тригер), котра прив'язана до того контейнера, в котрому знаходяться дані (зображення) і функціонує, за вміст контейнера піддається оновленню (для прикладу, додаються оновлені зображення) (Додаток В). На рис. 3.5 BLOB - тригер відстежує стан контейнера firstjavaapp, щойно до нього відбудеться звернення, тригер виконає свою роботу та результат буде внесено у контейнер output.

```
{
  "scriptFile": "__init__.py",
  "bindings": [
    {
      "name": "blobin",
      "type": "blobTrigger",
      "direction": "in",
      "path": "firstjavaapp/{blobname}.{blobextension}",
      "connection": "teststor2021_STORAGE"
    },
    {
      "name": "blobout",
      "type": "blob",
      "direction": "out",
      "path": "output/{blobname}_result.jpg",
      "connection": "teststor2021_STORAGE"
    }
  ]
}
```

Рисунок 3.5 – Приклад functionjson файлу для BLOB -тригера

На вхід застосунку подається зображення, котре уподальшому обробляється для ідентифікації на ньому осіб.

Хід подій обробки фото складається з таких елементів:

– виконати зменшення розміру зображення до розміру 700 х 700 пікселів, за умови, що воно велике;

– виділити всі обличчя на зображенні;

– порівняти виділені обличчя з обличчями з БД із відомими обличчями;

– при виявленні знайомого обличчя внести новий запис до таблиць «PHOTOS PERSONS INFO» та «PHOTOS» (рис. 3.6).

Рисунок 3.6 – Повідомлення BLOB -тригера про успішне прийняття та опрацювання зображення

Виділення обличь на фото можливо проводити при допомозі існуючих алгоритмів, зокрема HOG, метод Віоли-Джонса чи із застосуванням CNN. Найточніше розпізнавання обличь на зображенні може бути досягнуто при використанні CNN, проте даний підхід потребує значних ресурсів для обчислення, ось чому в даному додатку використовувався алгоритм HOG через його швидкодію та досить хорошу точність. Кожне розпізнане обличчя на фото обводиться синім прямокутником і підписується визначеним ім'ям, якщо обличчя заздалегідь невідоме, буде вказаний підпис «Stranger». Оброблене фото буде збережено у хмарному контейнері (Додаток Г).

При проектуванні БД враховувався те, що на одному фото може бути кілька людей, та одна і та ж сама людина може бути на кількох фотографіях. Зважаючи на це було обрано зв'язок багато-багатьом між таблицями. За такого підходу формуються три таблиці, зокрема дві таблиці - «джерела» («PERSONS» і «PHOTOS») та допоміжна таблиця («PHOTOS_PERSONS_INFO»). ER -діаграма БД показано на рис. 3.7.

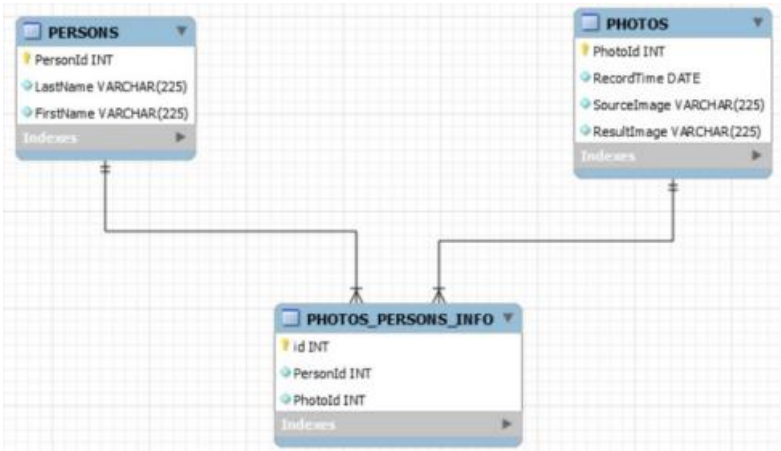


Рисунок 3.7 – ER -діаграма БД

Фрагмент коду додавання нового запису до таблиць «PHOTOS» та «PHOTOS_PERSONS_INFO» показано на рис. 3.8.

```
def write_db_record(faces_id, curtime, image_name):
    with pyodbc.connect('Driver=' + DRIVER +';Server=tcp:' + SERVER + ',1433;Database=db1;Uid=' + USERNAME
+';Pwd=' + PASSWORD + ';Encrypt=yes;TrustServerCertificate=no;Connection Timeout=30;') as conn:
        with conn.cursor() as cursor:
            pic_id = str(cursor.execute(SELECT_PHOTO_ID).fetchone()[0])
            cursor.execute("INSERT INTO " + PHOTOS_DB + " VALUES(CAST('" + curtime + "' AS DATETIME), '" +
            image_name + "', '" + image_name.replace(".jpg", "_result.jpg") + "');")
            for person_id in faces_id:
                cursor.execute("INSERT INTO " + PHOTOS_PERSONS_DB + " VALUES(" + str(person_id) + ", " + pic_id + ");")
```

Рисунок 3.8 – Приклад додавання нового запису до БД

Програма-тригер здатна опрацьовувати одночасно кілька фото. Щоб це забезпечити потрібно у файлі local.settings.json у полі «PYTHON_THREADPOOL_THREAD_COUNT» вказати число у діапазоні від 1 до 32, котре задає кількість потоків, які задіяні в обробці нових зображень (рис. 3.9).

local.settings.json > ...

```
{
  "IsEncrypted": false,
  "Values": {
    "AzureWebJobsStorage": "DefaultEndpointsProtocol=https;
    "FUNCTIONS_WORKER_RUNTIME": "python",
    "PYTHON_THREADPOOL_THREAD_COUNT": "8",
    "teststor2021_STORAGE": "DefaultEndpointsProtocol=https
  }
}
```

Рисунок 3.9 – Приклад вказання значення 8

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Класифікація шкідливих та небезпечних виробничих факторів

Шкідливий виробничий фактор – небажане явище, яке супроводжує виробничий процес і вплив якого на працюючого може призвести до погіршення самопочуття, зниження працездатності, захворювання, виробничо зумовленого чи професійного, і навіть смерті, як результату захворювання. Небезпечний виробничий фактор – небажане явище, яке супроводжує виробничий процес і дія якого за певних умов може призвести до травми або іншого раптового погіршення здоров'я працівника (гострого отруєння, гострого захворювання) і навіть до раптової смерті [16].

Поділ несприятливих чинників виробничого середовища на шкідливі та небезпечні зумовлене різним характером їх дії на людський організм, тим, що вони потребують різних заходів та засобів для боротьби з ними та профілактики викликаних ними ушкоджень, а також рядом причин організаційного характеру. В той же час між шкідливими та небезпечними виробничими факторами інколи важко провести чітку межу. Один і той же чинник може викликати травму і профзахворювання (наприклад, високий рівень іонізуючого або теплового випромінювання може викликати опік або навіть призвести до миттєвої смерті, а довготривала дія порівняно невисокого рівня цих же факторів – до хвороби; пилинка, що потрапила в око, спричиняє травму, а пил, що осідає в легенях, – захворювання, що зветься пневмоконіоз). Через це всі несприятливі виробничі чинники часто розглядаються як єдине поняття – небезпечний та шкідливий виробничий фактор (НШВФ) [16]. За своїм походженням та природою дії всі НШВФ можна поділити на 5 груп: фізичні, хімічні, біологічні, психофізіологічні та соціальні До фізичних НШВФ відносяться машини та механізми або їх елементи, а також вироби, матеріали, заготовки тощо, які рухаються або обертаються; конструкції, які руйнуються; системи, устаткування або елементи обладнання, які знаходяться під підвищеним тиском; підвищена запиленість та

загазованість повітря; підвищена або понижена температура повітря, поверхонь приміщення, обладнання, матеріалів; підвищені рівні шуму, вібрації, ультразвуку, інфразвуку; підвищений або понижений барометричний тиск та його різкі коливання; підвищена та понижена вологість; підвищена швидкість руху та підвищена іонізація повітря; підвищений рівень іонізуючих випромінювань; підвищене значення напруги в електричній мережі; підвищені рівні статичної електрики, електромагнітних випромінювань; підвищена напруженість електричного, магнітного полів; відсутність або нестача світла; недостатня освітленість робочої зони; підвищена яскравість світла; понижена контрастність; прямий та віддзеркалений блиск; підвищена пульсація світлового потоку; підвищені рівні ультрафіолетової та інфрачервоної радіації; гострі крайки, зачипки, шершавість на поверхні заготовок, інструментів та обладнання; розташування робочого місця на значній висоті відносно землі (підлоги); слизька підлога; невагомість.

Хімічні НШВФ:

- за характером дії на організм людини поділяються на токсичні, задушливі, наркотичні, подразнюючі, сенсибілізуючі, канцерогенні, мутагенні та такі, що впливають на репродуктивну функцію;

- за шляхами проникнення в організм людини поділяються на такі, що потрапляють через: 1) органи дихання; 2) шлунково-кишковий тракт; 3) шкіряні покриви та слизова оболонка;

- які перебувають у різному агрегатному стані: 1)твердому 2)газоподібному 3)рідкому.

Біологічні НШВФ – це: - патогенні мікроорганізми (бактерії, віруси, рикетсії, спірохети, грибки, найпростіші) та продукти їхньої життєдіяльності; - макроорганізми (тварини та рослини) та продукти їхньої життєдіяльності. До психофізіологічних НШВФ відносяться фізичні (статичні та динамічні) перевантаження і нервово-психічні перевантаження (розумове перенапруження, перенапруження аналізаторів, монотонність праці, емоційні перевантаження). 6 Соціальні НШВФ – це неякісна організація роботи, понаднормова робота, змушеність праці в колективі з поганими відносинами між його членами,

соціальна ізольованість з відривом від сім_ї, зміна біоритмів, незадоволеність роботою, фізична та/або словесна образа та її ризик, насильство та його ризик. Один і той же НШВФ за природою своєї дії може належати водночас до різних груп.

4.2 Вплив вібрації на людину

Вібрація - це механічні коливання пружних тіл або коливальні рухи механічних систем. Для людини вібрація є видом механічного впливу, який має негативні наслідки для організму [17].

Причиною появи вібрації є неврівноважені сили та ударні процеси в діючих механізмах. Створення високопродуктивних потужних машин і швидкісних транспортних засобів при одночасному зниженні їх матеріалоємності неминуче призводить до збільшення інтенсивності і розширення спектру вібраційних та віброакустичних полів. Цьому сприяє також широке використання в промисловості і будівництві високоефективних механізмів вібраційної та віброударної дії.

Дія вібрації може приводити до трансформування внутрішньої структури і поверхневих шарів матеріалів, зміни умов тертя і зносу на контактних поверхнях деталей машин, нагрівання конструкцій. Через вібрацію збільшуються динамічні навантаження в елементах конструкцій, стиках і сполученнях, знижується несуча здатність деталей, ініціюються тріщини, виникає руйнування обладнання. Усе це приводить до зниження строку служби устаткування, зростання імовірності аварійних ситуацій і зростання економічних витрат. Вважають, що 80% аварій в машинах і механізмах здійснюється внаслідок вібрації. Крім того, коливання конструкцій часто є джерелом небажаного шуму. Захист від вібрації є складною і багатоплановою в науково-технічному та важливою у соціальноекономічному відношеннях проблемою нашого суспільства [17].

Вплив вібрації на людину залежить від її спектрального складу, напрямку дії, прикладення, тривалості впливу, а також від індивідуальних особливостей людини. При оцінці вібраційного впливу потрібно враховувати, що коливальні

процеси притаманні живому організму. В основі серцевої діяльності і кровообігу та біострумів мозку лежать ритмічні коливання. Внутрішні органи людини можна розглядати як коливальні системи з пружними зв'язками. Частоти їх власних коливань лежать у діапазоні 3..6 Гц. Частоти власних коливань плечового пояса, стегон і голови щодо опорної поверхні (положення стоячи) складають 4...6 Гц, голови щодо плечей (положення сидячи) 25...30 Гц.

При впливі на людину зовнішніх коливань (хитавиці, струсів, вібрації) відбувається їхня взаємодія з внутрішніми хвильовими процесами, виникнення резонансних явищ. Так, зовнішні коливання частотою менш 0,7 Гц утворюють хитавицю і порушують у людини нормальну діяльність вестибулярного апарата. Інфразвукові коливання (менш 16 Гц), впливаючи на людину, пригнічують центральну нервову систему, викликаючи почуття тривоги, страху. При певній інтенсивності на частоті 6..7 Гц інфразвукові коливання, втягуючи у резонанс внутрішні органи і систему кровообігу, здатні викликати травми, розриви артерій, тощо [17].

Вібрація, що діє на людину, має широкий діапазон – від десятих часток одного до декількох тисяч Гц. Характерними ознаками шкідливого впливу вібрації на людину є можливі зміни у функціональному стані: підвищена втома, збільшення часу моторної реакції, порушення вестибулярної реакції. Медичними дослідженнями встановлено, що вібрація є подразником периферичних нервових закінчень, розташованих на ділянках тіла людини, що сприймають зовнішні коливання. Адекватним фізичним критерієм оцінки її впливу на організм людини є коливальна енергія, що виникає на поверхні контакту, а також енергія, поглинена тканинами і передана опорно-руховому апарату та іншим органам. У результаті впливу вібрації виникають нервовосудинні розлади, ураження кістково-суглобної та інших систем організму. Відзначаються, наприклад, зміни функції щитовидної залози, сечостатевої системи, шлунково-кишкового тракту. Так, медичні дослідження показали , що у працюючих в умовах вібрації відбуваються значні зміни кістковосуглобної системи, які виражаються у функціональній перебудові кісткової тканини, регіональному остеопорозі, кістовидних утвореннях у кістках, асептичному некрозі кісток, хронічних

переломах. Відзначається, що терміни виникнення змін у кістках у працівників вібраційних професій коливається в межах від 6-8 місяців до 2-5 років.

Шкідливість вібрації збільшується при одночасному впливі на людину таких факторів, як знижена температура, підвищений шум, запиленість повітря, тривала статична напруга тощо. Сучасна медицина розглядає виробничу вібрацію як могутній стрес-фактор, що має негативний вплив на психомоторну працездатність, емоційну сферу і розумову діяльність, підвищує ймовірність виникнення різних захворювань і нещасних випадків. Особливо небезпечний тривалий вплив вібрації для жіночого організму. Широкий комплекс патологічних відхилень, викликаний впливом вібрації на організм людини, кваліфікується як віброзахворювання [17].

Вібрація як фізичний чинник виробничого середовища спостерігається в металообробній, гірничодобувній, металобудівній, машинобудівній, авіаційній та інших галузях народного господарства. Джерелом вібрації можуть бути різні механізми, вібраційне устаткування, віброінструменти, акустичні системи, транспортні та сільськогосподарські машини.

Загальна вібрація поділяється на транспортну вібрацію, яка діє на людину на робочих місцях в транспортних засобах (трактори сільськогосподарські та промислові, самохідні сільськогосподарські машини (комбайни), тягачі, грейдери ті інші); транспортно-технологічну вібрацію, яка діє на людину на робочих місцях машин з обмеженою рухливістю (екскаватори, крани промислові та будівельні, гірничі комбайни, транспорт виробничих приміщень та інші) та технологічну вібрацію, яка діє на людину на робочих місцях стаціонарних машин чи передається на робочі, де немає джерел вібрації (верстати та метало-деревообробне, пресувально-ковальське обладнання, ливарні машини, електричні машини, насосні агрегати та вентилятори, обладнання для буріння свердловин, бурові верстати, машини для тваринництва, очищення та сортування зерна (у тому числі сушарні), обладнання промисловості будматеріалів (крім бетоноукладачів), установки хімічної та нафтохімічної промисловості та інші.

Оператори машин, які зазнають у процесі трудової діяльності впливу вібрації, підлягають попереднім та періодичним медичним оглядам відповідно

до Порядку проведення медичних оглядів працівників певних категорій, затвердженого Наказом МОЗ України від 21.05.2007 р. №246. Обов'язкові попередні (під час прийняття на роботу) та періодичні (протягом трудової діяльності) медичні огляди дозволять визначити стан здоров'я працівника та можливість виконання без погіршення стану здоров'я професійних обов'язків, своєчасно виявити ранні ознаки хронічного професійного захворювання, забезпечує динамічне спостереження за станом здоров`я в умовах дії шкідливих та небезпечних факторів і трудового процесу, вирішує питання щодо можливості продовжувати роботу в умовах дії шкідливих та небезпечних факторів і трудового процесу [17].

За результатами періодичних медичних оглядів роботодавець забезпечує проведення відповідних оздоровчих заходів Заключного акта у повному обсязі та усуває причини, що призводять до професійних захворювань. Організовує проведення лабораторних досліджень умов праці на робочих місцях та вживає заходів до усунення небезпечних і шкідливих для здоров'я виробничих факторів.

До роботи операторами машин допускаються особи не молодші 18 років, які пройшли попередній медичний огляд, мають відповідну кваліфікацію та ознайомлені з характером впливу вібрації на організм.

ВИСНОВКИ

При виконанні роботи вирішено такі завдання:

– розглянуто існуючі системи та комплекси відеоаналітики, проведено їх порівняльний аналіз;

– розглянуто та вивчено технології кордонних та хмарних обчислень;

– вивчено бібліотеки, алгоритми та засоби з розпізнавання та ідентифікації осіб на зображеннях, зокрема OpenCV, метод Віоли-Джонса, CNN, HOG і Microsoft Cognitive Services;

– розглянуто та вивчено компоненти Microsoft Azure: БД Azure, хмарне сховище, безсерверні обчислення та функції Azure;

– спроектовано та розроблено програму з обробки відеоданих та додаток-тригер з ідентифікації осіб.

Вибираючи алгоритм для ідентифікації осіб, ймовірно орперувати такими метриками, як швидкодія і точність. Щодо швидкості робот HOG є найшвидшим, а вже потім йдуть метод Віоли-Джонса та CNN. Щодо точності – тут найкращим є CNN, проте він потребує потужних обчислювальних ресурсів, власне тому при застосуванні CNN всі обчислення найкраще проводити на графічному процесорі. HOG працює досить добре, проте явно видно проблеми із розпізнаванням невеликих обличь (Додаток Д). Класифікатори HaarCascade у методі Віоли-Джонса функціонують аналогічно якісно, як і HOG, на жаль їх навчання триває досить довго. Швидкість роботи кожного алгоритму перебуває у прямій залежності від якості зображення: чим якісніше фото, тим швидше та точніше воно буде опрацьоване. Значні видозміни обличчя роблять розпізнавання малоймовірним, проте для найбільш типових ситуацій, для прикладу, людина вдягнула окуляри, кліпнула, засміялася, алгоритми здатні її правильно ідентифікувати. Кольоровість зображень не допомагає, але й не заважає розпізнаванню та ідентифікації обличь: більшість алгоритмів та методів працює з чорно-білими зображеннями.

Необхідно вказати шляхи щодо покращення поточної версії системи:

– дослідити інші засоби та алгоритми з розпізнавання та ідентифікації

осіб у відеопотоці;

– використовувати платформу EdgeX для під’єднання інших камер, щоби при додаванні нової не здійснювати повторно монтування програми камери кожного разу;

– на вхід до програми-тригера передавати не фото, а відео, котре надалі опрацьовувати, з метою оптимізації збереження інформації в хмарному сховищі.

ПЕРЕЛІК ДЖЕРЕЛ

1. Object selection in the image according to the Viola – Jones method [Електронний ресурс] – Режим доступа: https://api-2d3d-cad.com/viola-jones-method/#1. (Дата звернення: 07.03.2025).

2. What is edge computing? [Електронний ресурс] – Режим доступа: https://www.ibm.com/think/topics/edge-computing (дата звернення: 09.03.2025).

3. Azure IoT Edge | Microsoft Azure. [Електронний ресурс] – Режим доступа:: https://azure.microsoft.com/ services/ iot-edge/ (дата звернення: 09.03.2025).

4. Viola P., Jones M. J. Robust real-time face detection. International Journal of Computer Vision. Vol. 57, no. 2, 2004. P. 137–154

5. Метод компресії DVPack 2 [Електронний ресурс] – Режим доступа: https://algoritm.org/arch/07_4/07_4_14.pdf. (Дата звернення: 12.03.2025).

6. Метод розпізнавання обличь Віоли – Джонса (Viola – Jones) [Електронний ресурс] – Режим доступа: https://oxozle.com/2015/04/11/metodraspoznavaniya-lic-violy-dzhonsa-viola-jones/. (Дата звернення: 12.03.2025).

7. Виявлення автомобільних номерів на відео за допомогою класифікатора SVM та дескриптора HOG [Електронний ресурс] – Режим доступа: https://delirium-00.livejournal.com/1872.html. (Дата звернення: 16.03.2025).

8. RESEARCH ON THE STATE-OF-THE-ART DEEP LEARNING BASED MODELS FOR FACE DETECTION AND RECOGNITION [Електронний ресурс] – Режим доступа: https://science.lpnu.ua/ictee/all-volumes-and-issues/volume-4-number-2-2024/research-state-art-deep-learning-based-models (Дата звернення: 18.03.2025).

9. Офіційна документація бібліотеки OpenCV [Електронний ресурс] – Режим доступа: https://docs.opencv.org/4.x/d9/d52/ tutorial_java_dev_intro.html/ (Дата звернення: 19.03.2025).

10. Introduction to Java Development. [Електронний ресурс] – Режим доступа: https://docs.opencv.org/. (Дата звернення: 19.03.2025).

11. Face Detection with Python Using OpenCV [Електронний ресурс] – Режим доступа https://www.datacamp.com/tutorial/face-detection-python-opencv. (Дата звернення: 20.03.2025).

12. Згорткова нейронна мережа – просте пояснення CNN та її застосування [Електронний ресурс] – Режим доступа: https://evergreens.com.ua/ua/articles/cnn.html. (Дата звернення: 22.03.2025).

13. Що таке згорткові нейронні мережі (CNN, ConvNet)? [Електронний ресурс] – Режим доступа: https://thetransmitted.com/adlucem/shho-take-zgortkovi-nejronni-merezhi-cnn-convnet/ (Дата звернення: 22.03.2025).

14. QuickStart: Manage blobs with Java v12 SDK [Електронний ресурс] – Режим доступа: https://docs.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobsjava. (Дата звернення: 28.03.2025).

15. What is the Azure Face service? [Електронний ресурс] – Режим доступа: https://docs.microsoft.com/en-us/azure/cognitiveservices/face/overview. (Дата звернення: 29.03.2025).

16. Lupenko, S. A., Lytvynenko, I. V., Sverstiuk, A., Shelestovskyi, B., & Horkunenko, A. (2021). Software for Statistical Processing and Modeling of a Set of Synchronously Registered Cardio Signals of Different Physical Nature. CMIS, 194-205.

17. Lytvynenko, I., Lupenko, S., Nazarevych, O., Shymchuk, G., & Hotovych, V. (2021). Mathematical model of gas consumption process in the form of cyclic random process. 2021 IEEE 16th International Conference on Computer Sciences and Information.

18. Lytvynenko I. V. Method of segmentation of determined cyclic signals for the problems related to their processing and modeling. Scientific journal of the Ternopil National Technical University. No. 4 (88). 2017. ISSN: 2522-4433. P. 153–169. https://doi.org/10.33108/visnyk_tntu2017.04.153

19. Заікіна Д., Глива В. Основи охорони праці та безпека життєдіяльності. 2019. URL: https://doi.org/10.31435/rsglobal/001 (дата звернення: 14.04.2025).

20. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання / укл.: Стручок В. С. Тернопіль: ФОП Паляниця В. А., 2022. 156 с.

ДОДАТКИ

Додаток А

Створення ресурсної групи та облікового запису зберігання даних за допомогою PowerShell

```
PS /home/bronetsky-and> $resgroup = "picgroup"
PS /home/bronetsky-and> $location = "westeurope"
PS /home/bronetsky-and> New-AzResourceGroup -Name $resgroup -Location $location

ResourceGroupName : picgroup
Location          : westeurope
ProvisioningState : Succeeded
Tags              :
```

```
                  New-AzStorageAccount -ResourceGroupName $resgroup `
>>   -Name "picdata" `
>>   -SkuName Standard_LRS `
>>   -Location $location `
>>

StorageAccountName ResourceGroupName PrimaryLocation SkuName      Kind      AccessTier CreationTime         ProvisioningState EnableHttpsTrafficOnly LargeFileShares
------------------ ----------------- --------------- -------      ----      ---------- ------------         ----------------- ---------------------- ---------------
picdata            picgroup          westeurope      Standard_LRS StorageV2 Hot        5/2/2019,1:49:46 PM Succeeded         True
```

Додаток Б

Клас Azure Manager для взаємодії з хмарним сховищем та аналізу зображень

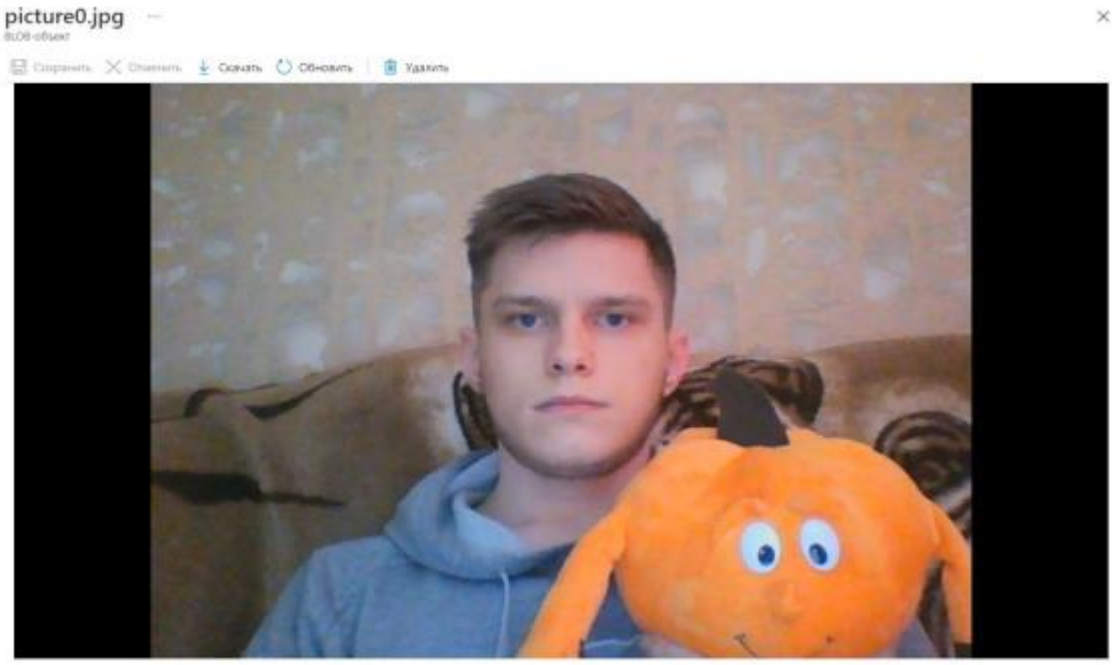
```
public class AzureManager {
    private final String CONNECTSTR = "insert connection string";
    private BlobServiceClient blobServiceClient;
    private BlobContainerClient containerClient;
    public AzureManager(String containerName)
    {
        this.blobServiceClient = new BlobServiceClientBuilder().connectionString(CONNECTSTR).buildClient();
        String offContainerName = containerName.toLowerCase() + java.util.UUID.randomUUID();
        this.containerClient = blobServiceClient.createBlobContainer(offContainerName);
    }

    public void UploadData(String filePath, String filename)
    {
        BlobClient blobClient = this.containerClient.getBlobClient(filename.toLowerCase());
        System.out.println("\nUploading to Blob storage as blob:\n\t" + blobClient.getBlobUrl());
        blobClient.uploadFromFile(filePath);
        System.out.println("Uploading file finished!");
    }
    public void                    ()
    {
      for(BlobItem item : containerClient.listBlobs())
          System.out.println(item.getName());
    }
    public void             (String filePath, String filename)
    {
        File                = new File( pathname filePath + "DOWNLOAD" + filename);
        System.out.println("\nDownloading blob to\n\t " + filePath + filename);
        BlobClient blobClient = this.containerClient.getBlobClient(filename.toLowerCase());
        blobClient.downloadToFile( filePath filePath + "DOWNLOAD" +filename);
        System.out.println("Done");
    }
```

Додаток В

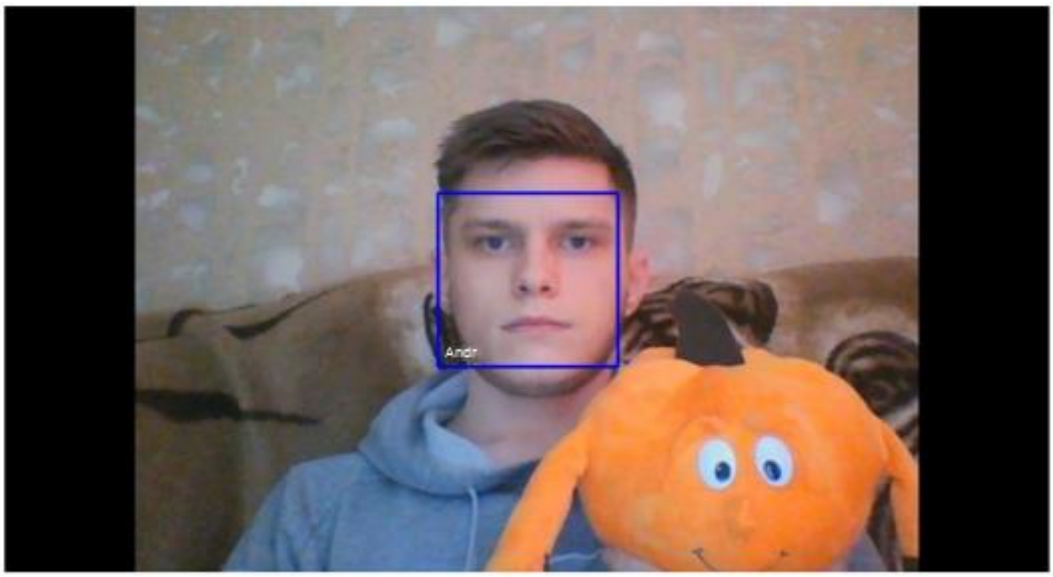
Приклад програми – тригера (фрагмент)

```
import logging
import io
import sys
from PIL import Image
import azure.functions as func
import azure.storage.blob as storblob

import threading
import numpy as np
import face_recognition
import cv2
import glob
import datetime
import pyodbc

    # final image composite size
FINAL_COMPOSITE_MAX_HEIGHT = 700
FINAL_COMPOSITE_MAX_WIDTH = 700
lock = threading.Lock()

    # date, path to local db
UNKNOWN = "Stranger"
PICPATH = "KnownFaces\\"

    # constants for azure db
SERVER = 'bdlserver.database.windows.net'
PHOTOS_DB = 'PHOTOS(RecordTime, SourceImage, ResultImage)'
PHOTOS_PERSONS_DB = 'PHOTOS_PERSONS_INFO(PersonId,PhotoId)'
USERNAME = 'badadmin'
PASSWORD = 'Mdevil2008'
CONTAINER_IN = 'firstjavaapp/'
DRIVER = '{ODBC Driver 17 for SQL Server}'
SELECT_PHOTO_ID = "SELECT CASE WHEN (SELECT COUNT(1) FROM PHOTOS) = 0 THEN 1
ELSE IDENT_CURRENT('PHOTOS') + 1 END;"


class Person:
    firstName = ""
    lastName = ""
    id_ = 0

    def __init__(self, firstName_, lastName_, id_):
        self.firstName = firstName_
        self.lastName = lastName_
```

Додаток Г



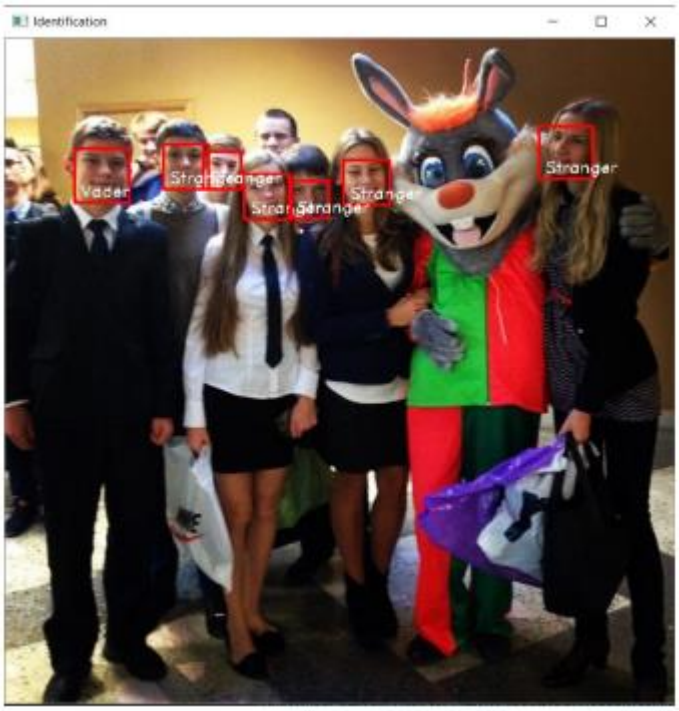
Нове зображення із зафіксованим обличчям



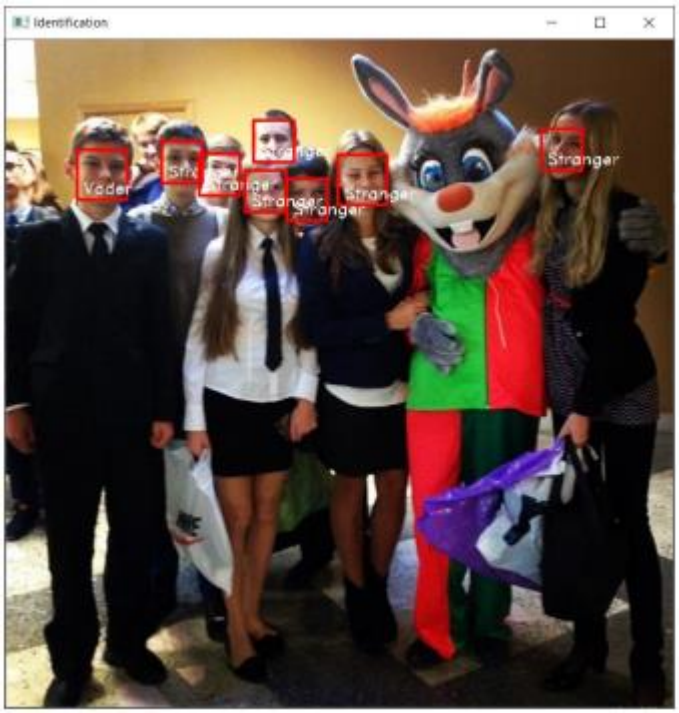
Оброблене зображення з ідентифікованим обличчям, що зберігається у хмарному контейнері

Додаток Д

Порівняння HOG і CNN щодо швидкодії та точності



HOG впорався за 9 секунд



CNN впоралася за 48 секунд

Усі обчислення проводились на процесорі Intel Core i7 – 9750H 2.6 GHz