

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБЛЕННЯ ЗАСТОСУНКУ ДЛЯ ВИЯВЛЕННЯ ТА
КЛАСИФІКАЦІЇ ПОДІБНОСТІ РИС ЛЮДСЬКИХ ОБЛИЧ
(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-17-2

Корякін І.М.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Творошенко І.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Кобилін О.А.
(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Корякіну І.М.
(прізвище, ім'я, по батькові)1. Тема роботи Розроблення застосунку для виявлення та класифікації подібності рис людських облич

затверджена наказом університету від 20 травня 2021 року № 663Ст

2. Термін подання студентом роботи до екзаменаційної комісії 24 травня 2021 р.

3. Вихідні дані до роботи Науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, бібліотека комп'ютерного зору з відкритим кодом OpenCV. Перелік використовуваних програмних засобів: Python 3.9, PyCharm 2021.1.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз існуючих застосунків для виявлення та класифікації подібності ознак досліджуваних об'єктів в Україні та за кордоном.2. Розроблення підходу для виявлення та класифікації подібності рис людських облич.3. Розроблення застосунку для виявлення та класифікації подібності людських облич.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми розпізнавання об'єктів на зображеннях, постановка задачі, аналіз існуючих методів розпізнавання, обрання методу для розробки, еталонні зображення, тестові зображення.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	12.04.2021	
2	Аналіз завдання, підбір літератури	13.04.21-16.04.21	
3	Аналіз літератури з досліджуваної проблеми	17.04.21-20.04.21	
4	Аналіз технічних засобів	21.04.21-23.04.21	
5	Аналіз методів	24.04.21-05.05.21	
6	Програмна реалізація	06.05.21-15.05.21	
7	Оформлення пояснювальної записки	16.05.21-26.05.21	
8	Перевірка на плагіат	27.05.21	
9	Рецензування	28.05.21	
10	Підготовка презентації та доповіді	29.05.21-30.05.21	
11	Занесення роботи в електронний архів	31.05.21	
12	Попередній захист кваліфікаційної роботи	31.05.21	

Дата видачі завдання 12 квітня 2021 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Творошенко І.С.
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 78 с., 44 рис., 2 дод., 59 джерел.

КОМП'ЮТЕРНИЙ ЗІР, РОЗПІЗНАВАННЯ ОБРАЗІВ, РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, РОЗПІЗНАВАННЯ ОБЛИЧ, МЕТОД ВІОЛІ-ДЖОНСА, ПРИМІТИВИ ХААРА, АЛГОРИТМ LOCAL BINARY PATTERNS.

Об'єктом роботи є процес розробки застосунку для виявлення та класифікації подібності рис людських облич.

Метою роботи є розробка застосунку для виявлення та класифікації подібності рис людських облич.

Використано методи виявлення обличчя Віоли-Джонса за допомогою примітивів Хаара та розпізнавання обличчя за допомогою методу Local Binary Patterns. Проведено дослідження методів виявлення обличчя та методи їх розпізнавання. Також були розглянуті методи виявлення та розпізнавання об'єктів.

У результаті роботи здійснена програмна реалізація застосунку для виявлення та класифікації подібності рис людських облич.

Результат роботи апробовано у вигляді 2 тез доповідей під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ» та XXVI Міжнародної науково-практичної конференції «Topical issues of practice and science».

COMPUTER VISION, IMAGE RECOGNITION, OBJECT RECOGNITION, FACE RECOGNITION, VIOLA-JONES METHOD, HAAR-LIKE FEATURES, LOCAL BINARY PATTERNS ALGORITHM.

The object of the research is the process of developing an application for detecting and classifying the similarities of human faces.

The aim of the research is to develop an application for detecting and classifying the similarities of human faces.

Methods used Viola-Jones for face detection using Haar primitives and Local Binary Patterns for face recognition. The research of methods of face detection and methods of their recognition is carried out. Methods for detecting and recognizing objects were also considered.

As a result of implemented software implementation of the application for detection and classification of similarities of human faces is carried out.

The result of the research was tested in 2 forms of abstracts of the report during the International Youth Forum "RADIO ELECTRONICS AND YOUTH IN THE XXI CENTURY" and XXVI International Scientific and Practical Conference "Topical issues of practice and science".

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	9
1 Аналіз існуючих застосунків для виявлення та класифікації подібності ознак досліджуваних об'єктів в Україні та за кордоном.....	11
1.1 Сучасний стан розвитку застосунків для виявлення та класифікації подібності ознак досліджуваних об'єктів в Україні та за кордоном.....	11
1.2 Аналіз літературних джерел щодо існуючих підходів виявлення та класифікації подібності ознак досліджуваних об'єктів	18
1.3 Постановка задачі	21
2 Розроблення підходу для виявлення та класифікації подібності рис людських облич	23
2.1 Методи виявлення та класифікації подібності ознак на зображенні	23
2.1.1 Метод розпізнавання об'єктів Віоли-Джонса на основі порівняння зі зразком	23
2.1.2 Методи розпізнавання об'єктів кореляційного аналізу на основі статистики.....	25
2.1.3 Метод розпізнавання об'єктів головних компонент на основі структурного та синтаксичного підходів	26
2.1.4 Метод розпізнавання об'єктів лінійного дискримінантного аналізу на основі статистики та нейронних мереж	28
2.2 Методи виявлення та класифікації подібності рис людських облич.....	30
2.2.1 Метод Віоли-Джонса	30
2.2.2 Метод порівняння шаблонів	32
2.2.3 Метод власних облич.....	33
2.2.4 Методи на основі штучних нейронних мереж.....	34

2.3 Проблема вибору та класифікації інформативних ознак на зображенні	35
2.4 Розроблення методики виявлення та класифікації подібності рис людських облич.....	38
3 Розроблення застосунку для виявлення та класифікації подібності людських облич.....	41
3.1 Вибір інструментальних засобів для створення застосунку	41
3.2 Етапи розроблення застосунку для виявлення та класифікації подібності рис людських облич.....	42
3.2.1 Встановлення програмного забезпечення	42
3.2.2 Знаходження еталонних зображень та зображень людей, які будуть порівнюватися з еталонами.....	43
3.2.3 Написання детектору та ідентифікатора облич	43
3.2.4 Навчання програми розпізнавати обличчя.....	44
3.2.5 Підготовка зображень для розпізнавання.....	46
3.2.6 Вивід результатів щодо порівняння зображень людей із еталонами.....	47
3.3 Тестування розробленого застосунку та аналіз результатів	49
3.3.1 Тестування роботи програми при розпізнаванні людини, фотографія якої була використана як еталон для навчання	50
3.3.2 Тестування роботи програми з різною кількістю фотографій, використаних як зображення-еталони.....	52
3.3.3 Тестування роботи програми з різними типами вхідних даних	56
3.3.4 Тестування роботи програми у випадках, коли на фотографіях відсутні обличчя.....	56
3.3.5 Тестування роботи програми у випадках, коли на фотографіях зображені тварини.....	57
3.3.6 Тестування роботи програми при зміні параметрів для функцій детектування та розпізнавання облич.....	58

	7
3.4 Перспективи подальшої роботи	59
Висновки	61
Перелік джерел посилання	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

ШІ – штучний інтелект

ІТ – інформаційні технології

ОС – операційна система

ОЗ – обробка зображень

КЗ – комп'ютерний зір

РО – розпізнавання образів

НМ – нейронна мережа

БД – база даних

ПХ – примітиви Хаара

МГК – метод головних компонент

ЛДА – лінійний дискримінантний аналіз

ЛДФ – лінійна дискримінантна функція

КФ – критерій Фішера

ІО – інформативні ознаки

МП – мова програмування

ВД – вхідні дані

ВСТУП

Комп'ютерний зір – це область інформатики, яка спрямована на поліпшення здатності комп'ютерів виявляти та ідентифікувати предмети та людей на зображеннях та відео. Комп'ютерний зір, як і інші типи ШІ, зосереджений на автоматизації та виконанні завдань, що імітують людські можливості. У цьому випадку комп'ютерний зір намагається імітувати людський зір і сприйняття.

Комп'ютерний зір існує вже понад половини сторіччя, але останніми роками знову з'являється інтерес до того, як машини «відчувають» і як комп'ютерний зір використовується для створення продуктів для споживачів та бізнесу. Прикладами таких програм є Amazon Go, Google Lens, автономні транспортні засоби та розпізнавання облич.

Головним рушієм усього цього є комп'ютерний зір. Простіше кажучи, комп'ютерний зір – це дисципліна в широкому полі штучного інтелекту, яка готує машини для бачення. Його метою є вилучення значень із пікселів.

З біологічної точки зору метою є створення обчислювальної моделі зорової системи людини. З технічної точки зору комп'ютерний зір має на меті створити автономну систему, яка може виконувати деякі завдання, які може виконувати зорова система людини.

Прогрес в області комп'ютерного зору визначається двома факторами: розвиток теорії, методів, і розвиток апаратного забезпечення. Довгий час теорія і академічні дослідження випереджали можливості практичного використання систем комп'ютерного зору. Умовно можна виділити ряд етапів розвитку теорії:

– до 1970-х років сформувався основний понятійний апарат в області обробки зображень, що є основою для дослідження проблем зору. Також були виділені основні завдання, специфічні для машинного зору, пов'язані з оцінкою фізичних параметрів сцени по зображенням, хоча ряд цих завдань все ще розглядався в дуже спрощеній формі;

– до 1980-х сформувалася теорія рівнів представлення зображень в методах їх аналізу;

– до 1990-х виявляється сформованим систематичне уявлення про підходи до вирішення основних, що вже стали класичними, завдань машинного зору;

– з середини 1990-х відбувається перехід до створення і дослідження великомасштабних систем комп'ютерного зору, призначених для роботи в різних природних умовах;

– поточний етап найбільш цікавий розвитком методів автоматичної побудови уявлень зображень в системах розпізнавання зображень і комп'ютерного зору на основі принципів машинного навчання.

Як наглядно видно, зараз уже існує дуже розвинена теоретична частина, а також дуже потужна апаратна частину у порівнянні з початком 1980-х. Цим обумовлена велика актуальність і перспективність даного напрямку.

1 АНАЛІЗ ІСНУЮЧИХ ЗАСТОСУНКІВ ДЛЯ ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЇ ПОДІБНОСТІ ОЗНАК ДОСЛІДЖУВАНИХ ОБ'ЄКТІВ В УКРАЇНІ ТА ЗА КОРДОНОМ

1.1 Сучасний стан розвитку застосунків для виявлення та класифікації подібності ознак досліджуваних об'єктів в Україні та за кордоном

На сьогоднішній день тема комп'ютерного зору, а зокрема, тема обробки зображень, є досить актуальною. Це обумовлено тим, що застосунки, які використовують механізм обробки зображень для взаємодії з користувачами, є дуже зручними та незвичайними у використанні. Тому останнім часом все більше і більше з'являються як програми, які вміють обробляти зображення, так і користувачі таких застосунків. До того ж, з часом, для програмістів розробка таких застосунків стала сильно простішою, ніж умовно 10 років тому назад. За цей час розроблялися нові підходи та методи, додавалися нові бібліотеки, які в певній мірі спрощували та автоматизували процес розробки таких програм.

Інакше кажучи, саме сфера комп'ютерного зору зараз є однією з найбільш актуальних в ІТ-індустрії. Тому застосунки, пов'язані з цим, є і будуть користуватися великим попитом найближчим часом.

Для того, щоб переконатися в цьому твердженні, було проаналізовано існуючі програмні застосунки для виявлення та класифікації подібності ознак досліджуваних об'єктів із зображення за кордоном та навіть на території України.

Першим у пошуку був застосунок «StarByFace». Ця програма дозволяє дізнатися, які знаменитості схожі на користувача, просто завантаживши особисту фотографію (рис. 1.1) [1]. На фотографії повинна бути лише одна людина. Після завантаження фотографії сайт проаналізує основні форми обличчя. Потім він обробить результати та порівняє їх із зірками, які є в його базі даних. Тоді користувач отримає результат. Він може побачити відсоток того, наскільки актор схожий на нього.



Рисунок 1.1 – Скріншот мобільного застосунку «StarByFace»

Наступний програмний застосунок на пошуковій сторінці – «Celebs Like Me», який також пропонує можливість завантаження своєї фотографії. Але у випадку, коли такої можливості немає, користувач може сфотографуватися в Інтернеті за допомогою камери на своєму пристрої, включаючи вбудовану в його смартфон (рис. 1.2) [2]. Результат буде показаний майже миттєво. Також доступний розширений пошук, за допомогою якого він може обрізати з фотографії обличчя, щоб отримати більш точний результат.

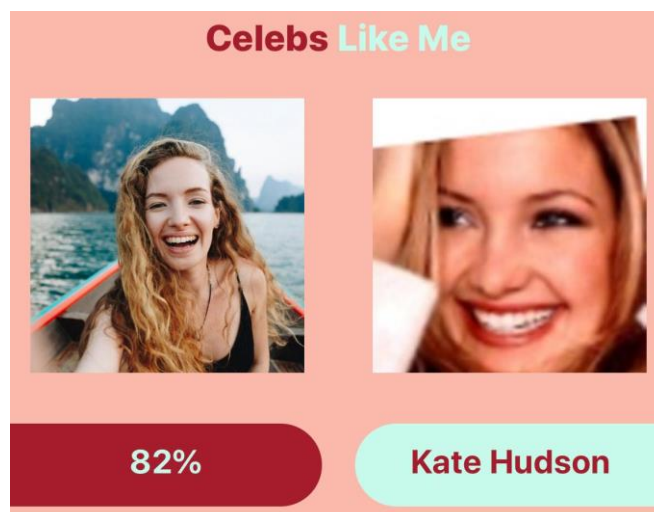


Рисунок 1.2 – Скріншот мобільного застосунку «Celebs Like Me»

Інша програма у цій тематиці – «Who Do I Look Like» (рис. 1.3) [3]. Вона використовує алгоритм машинного навчання Amazon для точнішого розпізнавання. В ній дається можливість порівняти свої селфі з портретами понад 1000 знаменитостей. Портрети сортуються за численними категоріями, що дозволяє краще визначити, на кого саме схожий користувач. Для пошуку алгоритм аналізує основні риси обличчя користувача, колір очей та волосся, форму вух та ряд інших параметрів. Недоліком цієї програми є те, що вона підтримується тільки на операційній системі iOS.



Рисунок 1.3 – Скріншот мобільного застосунку «Who Do I Look Like»

«Pictriv» – ще один портал, який дозволяє знайти свого близнюка серед зірок (рис. 1.4) [4]. Програма аналізує обличчя користувача та шукає схожу людину у своїй базі даних. Для роботи також потрібно завантажити особисту фотографію. Після завантаження фотографії сайт проаналізує обличчя та укаже стать та приблизний вік людини на фотографії. Усі схожі на користувача знаменитості будуть показані нижче.

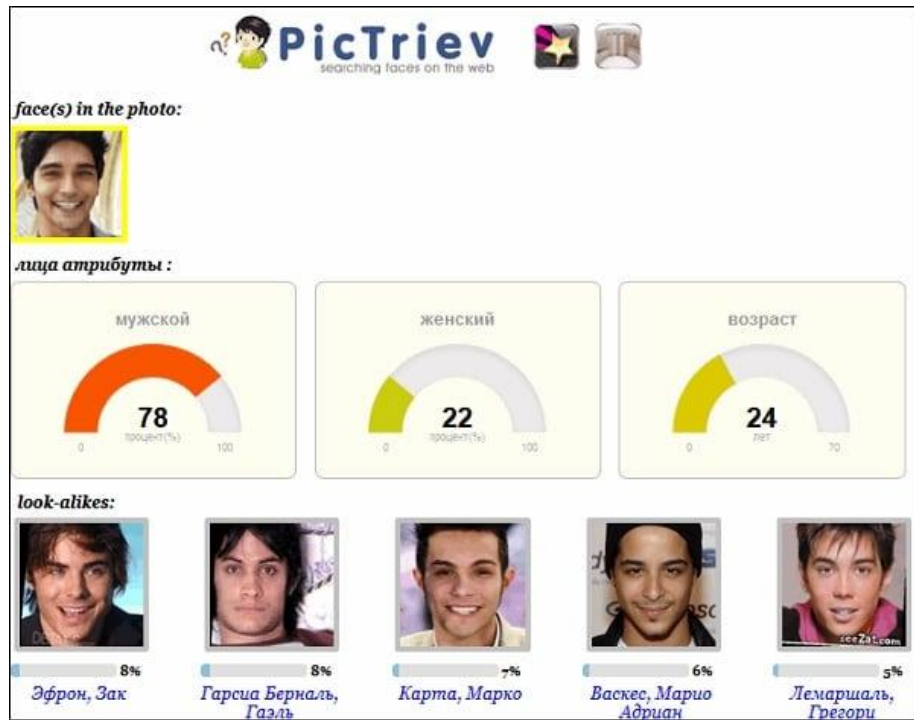


Рисунок 1.4 – Скріншот порталу «PicTrieв»

У програмі «LookALike» також присутня функція, де можна порівняти фото людини з певним персонажем фільму або серіалу (рис. 1.5) [5, 6].



Рисунок 1.5 – Скріншот програми «LookALike»

Ще один застосунок – «Looky», але який доступний лише для користувачів операційної системи iOS, де ви спочатку робите свою фотографію, чи завантажуєте її з галереї (рис. 1.6) [7]. Потім сервер виконує пошук і показує результати збігів серед зірок.

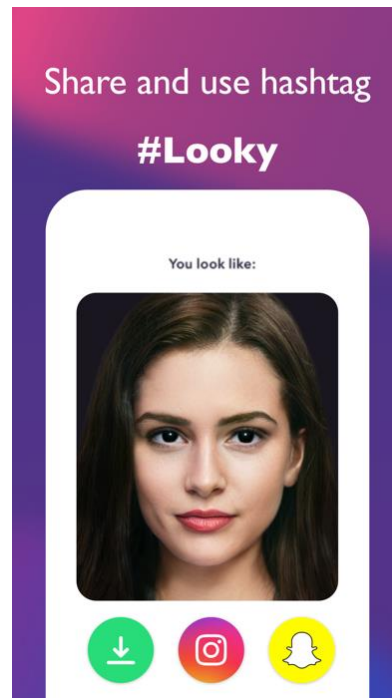


Рисунок 1.6 – Скріншот мобільного застосунку «Looky»

«Cinestaan» – це сервіс, який дозволяє знайти подібних зірок з індійського Боллівуду. Все, що користувачеві потрібно зробити, – це завантажити свою фотографію на портал. Технологія розпізнавання обличчя миттєво покаже відповідні результати.

Програма «FindFace» уже має більш широкий функціонал (рис. 1.7) [8]. Вона дозволяє знайти людину, дізнатися доступну інформацію про неї та навіть зв'язатися з цією людиною через акаунт у соціальній мережі, і для всього цього достатньо лише одного фото, зробленого мобільним телефоном. Безумовно великим плюсом цієї програми є те, що погане освітлення або невдалий кут не завадять алгоритму знайти потрібну людину. Сервіс ідентифікує людину на фотографії з точністю близько 70% за словами

розробників, і якщо бажану людину не знайшли, FindFace пропонує подивитися фотографію схожих людей.

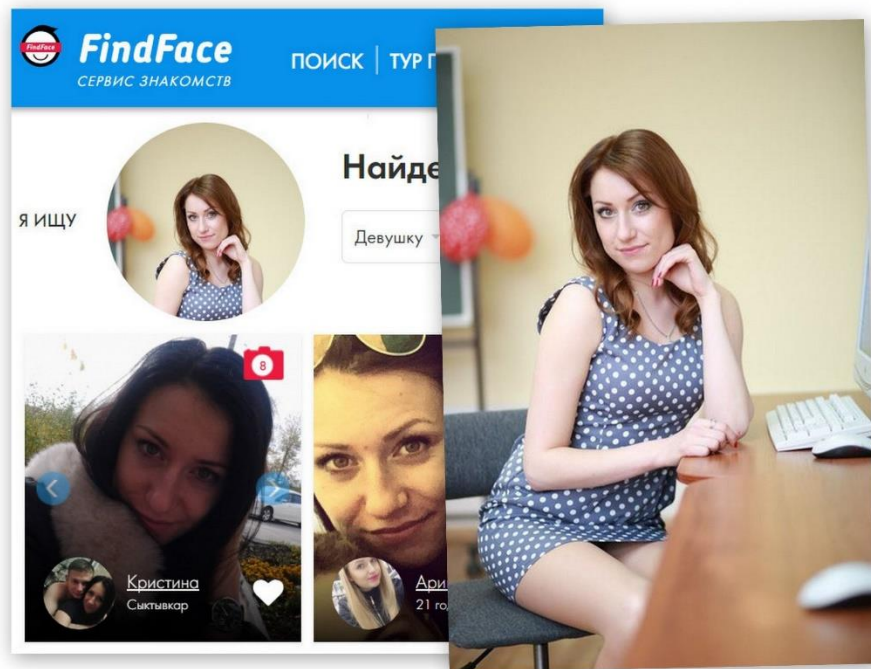


Рисунок 1.7 – Скріншот мобільного застосунку «FindFace»

Ще один застосунок, який також вміє ідентифікувати об'єкти на фото та до того ж за замовчуванням є майже на кожному телефоні за операційною системою Android та iOS – «Google Lens (Google Об'єktiv)». Він працює на базі системи комп'ютерного зору, розробленої для iPhone та смартфонів Android. Підтримується в додатках «Camera», «Google Photo», «Google Assistant». Володіє наступними функціями: виявлення дублікатів портретів; ідентифікація зображуваного; виявлення схожих предметів, виробів; розпізнавання штрих-коду; автоматичне підключення до бездротової мережі; копіювання тексту; визначення місця розташування.

Інша програма від цього ж розробника – «Google Goggles». Вона розпізнає копії портретів, читає штрих-коди, QR-коди, розпізнає товари. Додаткові функції: ідентифікація орієнтирів, предметів мистецтва, навколишнього світу; сканування, переклад текстів; відеозйомка, розпізнавання відеоданих.

Було знайдено ще 2 схожі програми – «CamFind» та «Зворотний пошук». Розглянуті програмні застосунки мають функцію зворотного пошуку. Сенс в тому, щоб виявити подібні з оригіналом портрети.

Не обійшлося і без програм від наших співвітчизників. Нещодавно українцями було розроблено програму «PolitScanner» (рис. 1.8) [9]. Біографія, доходи та діяльність народних депутатів України – про це все можна дізнатись тут, просто просканувавши обличчя парламентарів. Це стосується не тільки звичного зображення, але і плакатів, сітілайтів, банерів та листівок. Програма працює, використовуючи технології розпізнавання облич. Схема така: користувач за допомогою камери отримує фото обличчя певного депутата, і відразу він зможе переглянути біографію, декларацію та передвиборчі програми, які були знайдені в базі стосовно цього депутата.

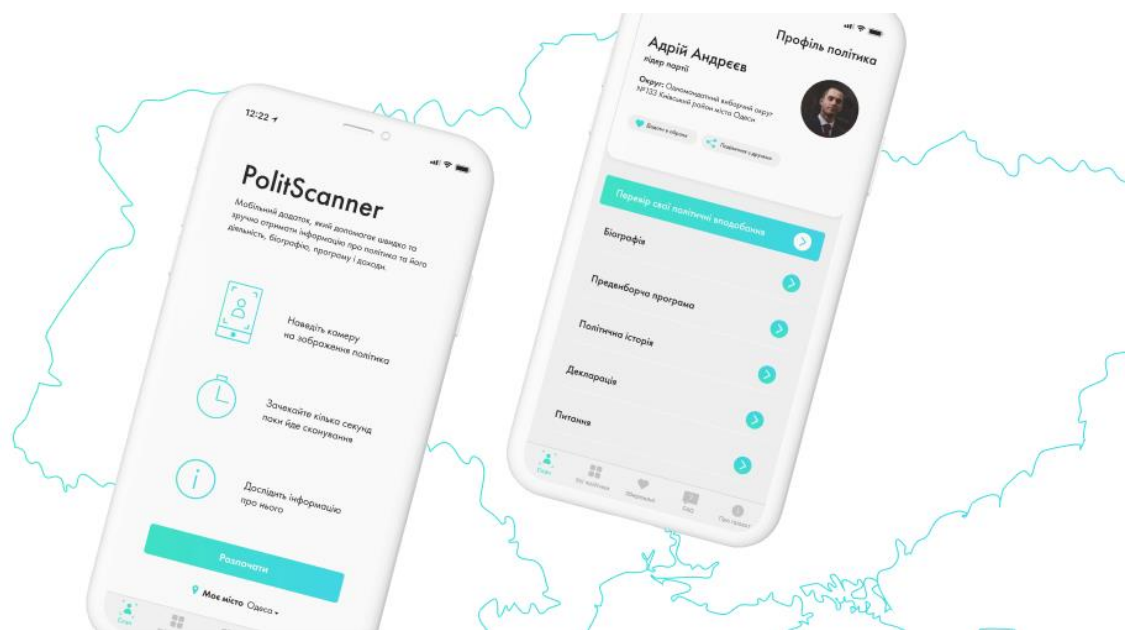


Рисунок 1.8 – Скріншот програми «PolitScanner»

Проаналізувавши існуючі програмні застосунки для виявлення та класифікації подібності ознак досліджуваних об'єктів із зображення, можна без сумнівів стверджувати, що ця тема є дуже актуальною на сьогоднішній день.

1.2 Аналіз літературних джерел щодо існуючих підходів виявлення та класифікації подібності ознак досліджуваних об'єктів

Для того, щоб більш детально вивчити тему роботи та вільно орієнтуватися в ній, було присвячено дуже багато часу для ознайомлення з великою кількістю наукового матеріалу. У даному підрозділі описаний результат аналізу літературних джерел стосовно даної теми.

У літературному джерелі [10] описуються методи побудови моделей форми та зовнішнього вигляду та способи їх створення, і як такі моделі можна використовувати для інтерпретації зображень. Дуже добре викладено про те, як саме пов'язана модель із ключовою метою комп'ютерного зору та якою повинна бути ця модель.

Наступним для ознайомлення була стаття [11]. У цій роботі було представлено структуру класифікації об'єктів, яка передбачає підкатегорії, щоб підвищити ефективність класифікації об'єктів на рівні категорії. Було чітко розділено дані на підкатегорії шляхом видобутку підкатегорій, керованих двозначністю. Даний матеріал є корисним у випадку, коли при порівнянні об'єктів характеристик унікальності недостатньо і виникає ситуація, коли система не може надати точний результат.

Ще одна стаття з того ж джерела [12]. Тут представлено новий адаптивний підхід активного навчання, який поєднує в собі міру щільності інформації та міру найбільшої невизначеності при виборі критичних випадків для позначення і класифікації зображень. Приведені результати роботи над двома важливими завданнями комп'ютерного зору, розпізнавання об'єктів та розпізнавання сцен, демонструють ефективність запропонованого підходу.

У статті [13] було досліджено глобальну самоподібність та її переваги перед локальною самоподібністю. Автори роблять три внески: вони пропонують обчислювально-ефективні алгоритми для вилучення дескрипторів глобальної самоподібності для класифікації; вони показують, як ефективно використовувати ці дескриптори для виявлення в структурі з

розсувним вікном та в структурі з гілками та прив'язками; експериментально демонструють на Pascal VOC 2007 та на ETHZ Shape Class, що глобальна самоподібність перевершує локальну як для класифікації, так і для виявлення, і що дескриптори глобальної самоподібності доповнюють звичайні дескриптори, такі як градієнти або колір.

Була досліджена ще одна стаття [14], у якій порівнюються особливості людського зору та комп'ютерного. Тут говориться про проблеми комп'ютерного зору взагалі. А саме, що при комп'ютерному розпізнаванні об'єктів (або класифікації) дані повинні трансформуватися у об'єктно-інваріантне внутрішнє подання. Для того, щоб вирішити цю проблему, потрібні дві ключові методики, тобто надійний детектор та інваріантний дескриптор. На жаль, люди поки що не знайшли жодного ефективного рішення. Зорові показники людини все ще перевершують ефективність комп'ютерного зору в багатьох аспектах. Отже, як майбутній напрям, комп'ютерний зір має набути деяких речей з неврології та науки про мозок. У роботі обговорюється, чому комп'ютерний зір може навчитись у людського зору і як на це вплинуть нові міждисциплінарні дослідження. Також розказується, із якими проблемами у майбутньому ми можемо зіткнутися, вивчаючи цей напрямок.

Наступні матеріали стосуються вже роботи комп'ютерного зору з обличчям, їх виявленням та ідентифікацією. І перша стаття на цю тему була [15]. В ній запропоновано метод автоматичного виявлення та локалізації граней на цифрових зображеннях. Метод заснований на засвоєнні прикладів та аналізі цифрових зображень із роздільною здатністю. Особливий акцент робиться на управлінні навчальними даними з метою покращення результатів. Надано різні експериментальні результати, отримані за допомогою багатошарового перцептронну (MLP) як класифікатора.

У статті [16] представлено компонентну основу для виявлення та ідентифікації обличчя. Модулі виявлення та ідентифікації обличчя мають однакову ієрархічну архітектуру. Вони обидва складаються з двох рівнів

класифікаторів, рівня з набором класифікаторів компонентів та рівня з єдиним комбінованим класифікатором. Класифікатори компонентів самостійно виявляють чи ідентифікують частини обличчя на зображенні. Їхні вихідні дані передаються комбінованому класифікатору, який виконує остаточне виявлення чи ідентифікацію обличчя. Тут авторами описано алгоритм, який автоматично вивчає два окремі набори компонентів обличчя для завдань виявлення та ідентифікації. В експериментах порівняно системи виявлення та ідентифікації зі стандартними глобальними підходами. Експериментальні результати чітко показують, що цей підхід на основі компонентів перевершує глобальний підхід.

У наступному джерелі [17] говориться про запропонований підхід де-ідентифікації облич у різних позах. Іноді обличчя людини слід де-ідентифікувати. Однією з проблем, яку повинен вирішити будь-який метод збереження природності обличчя, є зміна пози обличчя. У цій роботі побудовано кілька моделей облич на базі даних зображень обличчя, а потім їх використання для виявлення пози обличчя, а також галерею облич, яка потім використовується для де-ідентифікації. Дана стаття може бути дуже корисною, якщо також розглядати її як механізм для ідентифікації облич у різних позах.

Стаття [18] розповідає про ще один підхід розпізнавання облич. Система розпізнавання обличчя, представлена в цій роботі, є гібридом відомих алгоритмів. Першим етапом запропонованого методу є застосування алгоритму виявлення шкіри, щоб вказати всі місця розташування шкіри на зображенні. По-друге, визначити такі риси обличчя, як очі, рот і ніс. Нарешті, застосовується етап перевірки, щоб переконатися, що визначені риси є рисами обличчя. В експериментах на зображеннях, що мають вертикальні фронтальні грані з будь-яким фоном, ця система досягла високих показників виявлення та низьких помилкових спрацьовувань.

У статті [19] говориться про те, що метою даної роботи є класифікація та оцінка великої кількості методів, які займаються виявленням всіх областей

зображення, що містять обличчя, незалежно від його тривимірного положення, орієнтації та умов освітлення. У статті також обговорюються відповідні питання, такі як збір даних, показники оцінки та порівняльний аналіз. Проаналізувавши ці алгоритми та виявивши їх обмеження, автори роблять для себе акцент на кількох перспективних напрямках для подальших досліджень.

Після детального вивчення та аналізу різних літературних джерел, було зроблено висновок: тема є дуже актуальною, добре дослідженою, але, на жаль, практично далеко не всі розглянуті методи реалізовані на сьогоднішній день. Це означає, що в цій темі дуже багато простору для ентузіастів.

1.3 Постановка задачі

Після детального аналізу застосунків і літературних джерел в даному напрямку можна стверджувати, що тема кваліфікаційної роботи є дуже актуальною. Тому що на сьогодні продукти, що створюються в даній тематиці користуються великим попитом серед користувачів, а також дуже велика кількість теоретичного матеріалу ще не реалізована на практиці.

Об'єктом роботи є процес розробки застосунку для виявлення та класифікації подібності рис людських облич.

Метою роботи є розробка застосунку для виявлення та класифікації подібності рис людських облич.

Виходячи із мети роботи, необхідно вирішити такі завдання:

- дослідити методи виявлення та класифікації подібності ознак на зображенні;
- дослідити методи виявлення та класифікації подібності рис людських облич;
- дослідити проблему вибору та класифікації інформативних ознак на зображенні;

- розробити методику виявлення та класифікації подібності рис людських облич;
- визначитися з інструментальними засобами для створення застосунку;
- виконати всі етапи розроблення застосунку для виявлення та класифікації подібності рис людських облич;
- протестувати розроблений застосунок та зробити аналіз результатів;
- визначити перспективи подальшої роботи.

2 РОЗРОБЛЕННЯ ПІДХОДУ ДЛЯ ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЇ ПОДІБНОСТІ РИС ЛЮДСЬКИХ ОБЛИЧ

2.1 Методи виявлення та класифікації подібності ознак на зображенні

Розпізнавання образів – це процес віднесення вихідних даних до певного класу за допомогою виділення істотних ознак, що характеризують ці дані, із загальної маси несуттєвих даних [20–40].

На сьогоднішній день виділяють 4 групи методів розпізнавання об'єктів:

– порівняння зі зразком (ці методи реалізуються за допомогою геометричної нормалізації і також враховується фактична відстань до прототипу);

– нейронні мережі (тут для обраної мережі підбираються відповідні коефіцієнти. Далі ця нейронна мережа обробляє вхідний об'єкт. Група рецепторів мережі відповідає за прийом своєї характеристичної властивості);

– статистичні методи (такі методи включають в себе елементи теорії статистичних рішень та Байєсовський підхід);

– структурні та синтаксичні методи (ці методи розбивають вхідний об'єкт на його компоненти. Далі будуються правила в залежності від входження цих компонентів до певного класу чи ні).

2.1.1 Метод розпізнавання об'єктів Віоли-Джонса на основі порівняння зі зразком

Цей метод був запропонований у 2001 році для пошуку і виявлення об'єктів на зображенні. І на сьогоднішній день він є найпоширенішим методом для цих цілей. Метод використовує принцип сканувального вікна.

Об'єкти шукаються з використанням функцій Хаара, які засновані на вейвлетах Хаара. Найбільш поширена реалізація цього методу належить

бібліотеці комп'ютерного зору OpenCV і зараз використовується для розпізнавання осіб (рис. 2.1) [41].



Рисунок 2.1 – Загальна схема роботи методу Віюлі-Джонса

Після навчання на тестовій вибірці є навчена база даних з T слабких класифікаторів. Для кожного класифікатора відомі: ознака Хаара, що використовується в цьому класифікаторі, його положення всередині вікна розміром 24×24 пікселя і значення порога E .

На вхід методу надходить зображення $I(r, c)$ розміром $W \times H$, де $I(r, c)$ – яскравісна складова зображення. Результатом роботи методу є безліч прямокутників $R(x, y, w, h)$, що визначають положення об'єктів в оригінальному зображенні I .

Метод сканує зображення I на декількох масштабах, починаючи з базової шкали: розмір вікна 24×24 пікселя і 11 масштабів, при цьому кожен наступний рівень в 1,25 рази більше попереднього, за рекомендацією авторів.

Для того, щоб класифікатор прийняв рішення щодо того, чи є в розглянутому щойно вікні шуканий шаблон, потрібно, щоб виконувалася умова сильного класифікатора:

$$H(x) = \begin{cases} 1, & \text{якщо } \sum_{t=1}^T a^{(t)} h_j^{(t)} \geq \frac{1}{2} \sum_{t=1}^T a^{(t)} \\ 0, & \text{в іншому випадку} \end{cases}. \quad (2.1)$$

В даний час існує декілька способів оптимізації методу Віоли-Джонса. Так, завдяки використанню адаптованого сканувального вікна, крок зсуву якого можна змінювати і робити різним для осей X і Y , можна трохи підвищити надійність і швидкість виявлення об'єктів та розпізнавання осіб [42].

Метод має наступні переваги:

- можливе виявлення більше ніж 1 об'єкту на зображенні;
- використання простих класифікаторів демонструє добру швидкість розпізнавання і дозволяє використовувати цей метод у відеопотоці.

Із негативних факторів можна відзначити те, що цей метод є складним для навчання, так як для навчання потрібна велика кількість початкових даних і велика кількість часу для самого навчання.

2.1.2 Методи розпізнавання об'єктів кореляційного аналізу на основі статистики

Виявлення об'єктів, визначених еталонами, здійснюється методами кореляційного аналізу. Кореляційний аналіз об'єднує групу методів, в яких передбачається, що для кожного класу об'єктів відомо одне або декілька контрольних зображень. Виходячи з цієї інформації, класифікується кожне зображення з тих, що надаються на вхід. Традиційна техніка порівняння зображення з еталоном заснована на розгляді зображень як двовимірних функцій яскравості (дискретні двовимірні матриці інтенсивності). У цьому випадку вимірюється або відстань між зображеннями, або ступінь їхньої близькості [43].

При розпізнаванні об'єктів на зображенні шляхом порівняння з еталонами для посилення ймовірності правильного розпізнавання база даних

еталонів збільшується, що призводить до зростання часу розпізнавання. Одним із шляхів вирішення цієї проблеми є використання інформаційно-теоретичного підходу та загальносистемного принципу мінімуму інформаційної неузгодженості Куллбека-Лейблера [44]. Використовуючи алгоритм самонавчання, заснований на метриці Куллбека-Лейблера з бази даних, що містить велику кількість зображень певної множини об'єктів, відбираються зображення-еталони. Для кожного зображення-еталону будується гістограма якоїсь ознаки, наприклад, яскравість, колір, текстура, форма. Розпізнавання здійснюється шляхом порівняння гістограм зображення об'єкта та еталонних зображень за принципом мінімуму інформаційної неузгодженості. Порівняння гістограм проводиться із застосуванням класичного байєсівського підходу для розпізнавання без вчителя – на основі критерію максимуму апостеріорної імовірності того, що зображення X належить класу, заданому еталоном X_i . Це дозволяє зменшити обсяг обчислень за рахунок виключення операції динамічного вирівнювання гістограм. Для підвищення обчислювальної ефективності відбір зображень-еталонів та сам процес розпізнавання виконують із використанням методу направленої перебору альтернатив [44].

Недоліком цих методів є велика кількість часу та ресурсів, необхідних для отримання результату. Але в якості переваги є те, що результат буде точним.

2.1.3 Метод розпізнавання об'єктів головних компонент на основі структурного та синтаксичного підходів

Багато систем розпізнавання об'єктів на зображеннях використовують проєкційні методи побудови простору ознак меншої розмірності. Одним з поширених методів зменшення розмірності зображень є метод головних компонентів [45]. В даний час пропонується багато алгоритмів, що

використовують цей метод для вирішення проблеми пошуку та розпізнавання об'єктів на зображеннях.

Отже, завдання методу головних компонент полягає в тому, щоб побудувати новий простір ознак меншої розмірності, дисперсія між осями якої буде перерозподілена так, щоб максимізувати дисперсію по кожній з них. Для цього виконується послідовність наступних дій:

- обчислюється загальна дисперсія вихідного простору ознак. Це не можна зробити простим підсумовуванням дисперсій по кожній змінній, оскільки вони, в більшості випадків, не є незалежними. Тому підсумовувати потрібно взаємні дисперсії змінних, які визначаються з коваріаційної матриці;

- обчислюються власні вектори і власні значення коваріаційної матриці, що визначають напрямки головних компонент і величину пов'язаної з ними дисперсії;

- проводиться зниження розмірності. Діагональні елементи коваріаційної матриці показують дисперсію за вихідною системою координат, а її власні значення – за новою. Тоді розділивши дисперсію, пов'язану з кожною головною компонентою на суму дисперсій за всіма компонентами, отримуємо частку дисперсії, пов'язану з кожною компонентою. Після цього відкидається стільки головних компонент, щоб частка решти становила 80%–90%.

Для випадкових величин X та Y коваріація обчислюється за формулою:

$$\text{cov}(X, Y) = E[(X - E[X])(Y - E[Y])], \quad (2.2)$$

де E – математичне очікування.

Перевагами методу головних компонент є те, що він є дуже гнучким і на практиці дуже часто застосовують різні спеціальні критерії для визначення числа компонент, щоб збільшити точність вимірювань.

Основними недоліками цього методу є:

- неможливість смислової інтерпретації компонент, оскільки вони «вбирають» в себе дисперсію від декількох вихідних змінних;
- метод може працювати тільки з безперервними даними.

2.1.4 Метод розпізнавання об'єктів лінійного дискримінантного аналізу на основі статистики та нейронних мереж

Технологія розпізнавання зображень із використанням методів проєкції включає два етапи. На першому етапі класифікатор будується за допомогою навчального набору зображень. На другому етапі невідомі зображення розпізнаються за допомогою побудованого класифікатора.

Для побудови класифікатора використовуються різні методи, з яких можна відзначити лінійний дискримінантний аналіз. Цей метод дає змогу перетворити вихідний простір зображень в малорозмірний простір ознак, в якому зображення класів групуються навколо їх центрів, а центри класів віддаляються один від одного настільки, наскільки це можливо [45].

Тобто за допомогою цього методу максимізується відношення міжкласових відмінностей до внутрішньокласових.

Для двокласової задачі лінійна дискримінантна функція, що є розділяючою гіперплощиною, має вигляд:

$$D(X) = W^T X = a, \quad (2.3)$$

де $W = (w_1, w_2, \dots, w_l)$ – ваговий вектор одиничної довжини;

a – скалярна порогова величина.

Послідовність дій для розпізнавання, що дозволяє віднести невідомий об'єкт X до однієї з груп, має такий вигляд (рис. 2.2):

- якщо $W^T X < a$, то w_1 (перший клас);
- якщо $W^T X \geq a$, то w_2 (другий клас).

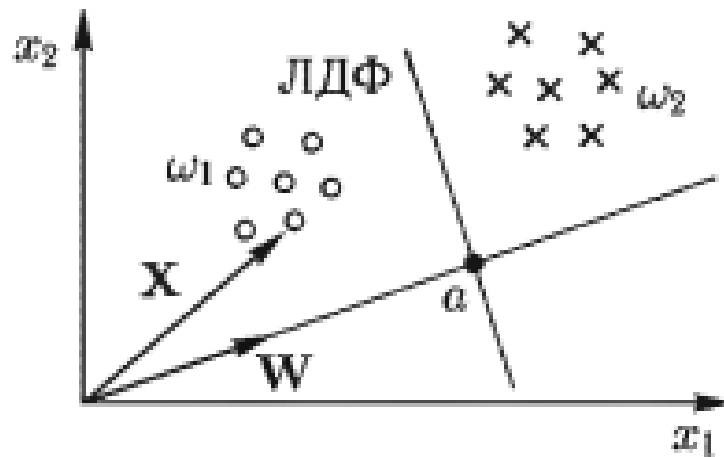


Рисунок 2.2 – Графік лінійної дискримінантної функції та варіанти віднесення об'єкту до певного класу

Для того щоб визначити складові вектора W і поріг a , можна скористатися критерієм мінімізації помилки класифікації. Однак для цього потрібно досить складна ітеративна процедура пошуку параметрів вирішального правила. Це завдання можна вирішити, застосовуючи інший критерій оптимізації – критерій Фішера. Він дозволяє знайти такий вектор W , при якому проекції точок класів на обраний напрям поділяються найкращим чином. Порог a визначається виходячи з критерію оптимальності розбиття об'єктів на два класи. У цьому випадку метод пошуку невідомих параметрів W і a можна представити таким чином:

- знайти W , використовуючи критерій Фішера;
- спроектувати точки обох класів на пряму, яка визначається положенням W ;
- вирішити одновимірну задачу по пошуку найкращої величини a .

Перевагою методу є велика точність розпізнавання для широкого діапазону умов освітленості.

Недоліками є залежність від змін ракурсу та велика потреба в ресурсах при роботі із масштабними базами даних.

2.2 Методи виявлення та класифікації подібності рис людських облич

Завдання розпізнавання обличчя складається з двох підзадач: автоматичного розпізнавання обличчя на зображенні та, при необхідності, ідентифікації особи за обличчям.

Розпізнавання обличчя належить до завдань розпізнавання зображень. Такі проблеми не мають точного аналітичного рішення.

Основною ідеєю розпізнавання людського обличчя є виділення інформативних особливостей зображення, кодування та порівняння закодованого обличчя з базою даних. У найбільш загальному випадку алгоритм вирішення проблеми розпізнавання обличчя людини на зображенні складається з наступних етапів:

- виявлення факту присутності обличчя людини на зображенні та виділення цього обличчя;
- розпізнавання та опис ключових рис обличчя (наприклад, таких як очі, ніс, брови, рот, вуха тощо);
- подання обличчя в певному просторі (моделювання);
- порівняння зі стандартами та ідентифікація (класифікація).

2.2.1 Метод Віоли-Джонса

На сьогодні метод Віоли-Джонса [46, 47] є найпопулярнішим методом пошуку області обличчя на зображеннях завдяки своїй високій швидкості та ефективності.

Детектор обличчя Віоли-Джонс заснований на трьох основних принципах:

- інтегральне подання зображень за характеристиками Хаара, що дозволяє дуже швидко розрахувати необхідні ознаки;

– метод побудови класифікатора на основі адаптивного алгоритму посилення (AdaBoost);

– метод поєднання класифікаторів у каскадну структуру.

Ці технології дозволяють швидко реалізувати знаходження обличчя на зображенні в режимі реального часу.

Для того, щоб почати виявлення обличчя на зображенні за допомогою методу Віоли-Джонса, необхідна наявність самого зображення, обраного вікна сканування та обрані використовувані ознаки. Далі вікно сканування починає послідовно рухатися по зображенню з кроком в 1 ділянку вікна (припустимо, розмір самого вікна є 24×24 ділянки). При скануванні зображення в кожному вікні обчислюється приблизно 200 000 варіантів розташування ознак, за рахунок зміни масштабу ознак і їх положення в вікні сканування. Сканування проводиться послідовно для різних масштабів. Масштабується не саме зображення, а скануюче вікно (змінюється розмір скануємої ділянки). Усі знайдені ознаки потрапляють до класифікатора, який і приймає рішення щодо того, чи є на зображенні обличчя, чи ні (рис. 2.3) [48].

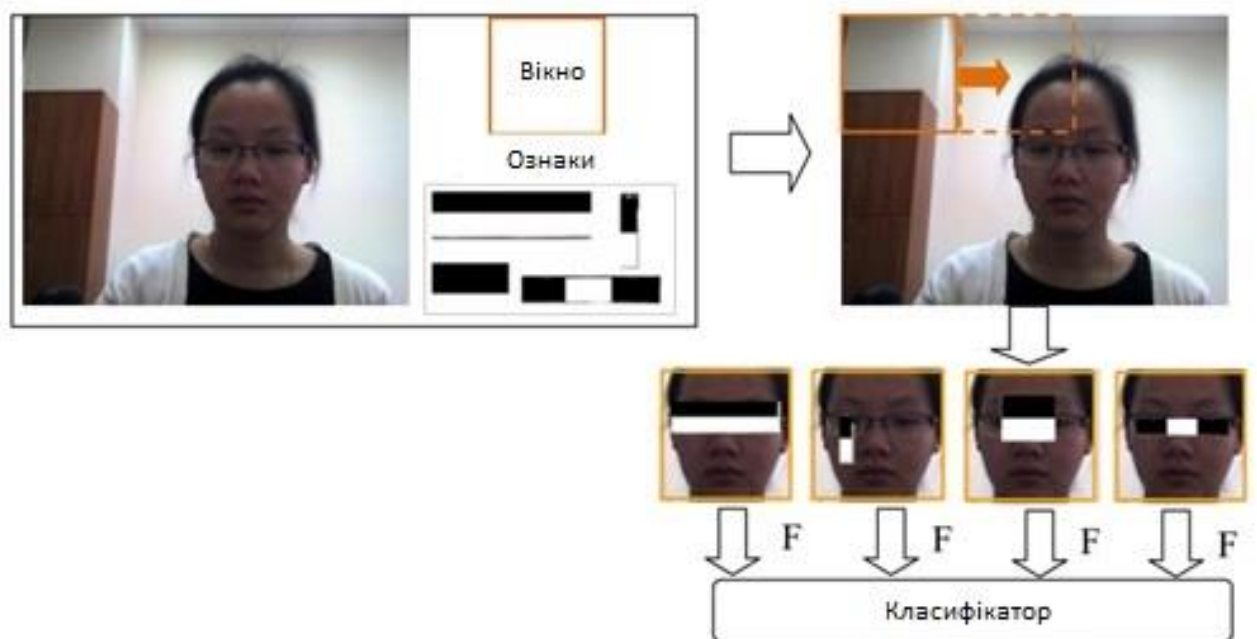


Рисунок 2.3 – Схема розпізнавання облич за методом Віоли-Джонса

Переваги:

- висока швидкість виявлення об'єктів;
- висока ймовірність точного виявлення обличчя під різними невеликими кутами;
- низька ймовірність хибного виявлення обличчя.

Недоліки:

- необхідна велика кількість часу для навчання;
- при великому куті точність виявлення значно падає;
- велика чутливість до освітлення.

2.2.2 Метод порівняння шаблонів

Основою цього методу [49] є вибір ділянок обличчя на зображенні, а потім порівняння цих областей для двох різних зображень. Кожна область, що зійшлася, збільшує ступінь схожості зображень. Для порівняння областей зображень використовуються прості алгоритми, такі як піксельне порівняння (рис. 2.4).



Рисунок 2.4 – Виділення областей обличчя на зображенні для подальшого порівняння із шаблоном

Недоліком цього методу є те, що він вимагає великих ресурсів як для зберігання областей, так і для їх порівняння. Оскільки використовується найпростіший алгоритм порівняння, знімки повинні бути зроблені в суворо встановлених умовах: не допускаються помітні зміни кута, освітлення, емоційних проявів тощо.

Точність розпізнавання за допомогою цього методу становить близько 80%, що є хорошим результатом.

2.2.3 Метод власних облич

Метод власних облич [50] використовує аналіз головних компонентів для зменшення розмірності даних без значної втрати інформації.

Простір власних облич формується за допомогою методу головних компонентів до навчальної множини зображень. Потім навчальні зображення проєктуються на простір власних облич. Далі тестове зображення проєктується на новий простір і обчислюється відстань між проєктованим тестовим зображенням та зображеннями з навчальної множини зображень. Розпізнаним вважається те зображення, яке є найближчим до навчального.

Крім того, метод головних компонентів використовується лише для розпізнавання контурів обличчя на зображенні. Для облич значення компонентів у власному просторі мають велике значення, а в доповненні власного простору вони близькі до нуля. Виходячи з цього факту, можна з'ясувати, чи є вхідне зображення обличчям (рис. 2.5) [51].



Рисунок 2.5 – Приклад зображень власних облич (власних векторів)

Перевагою цього методу є те, що якщо дотримані ідеалізованих умов, точність розпізнавання за допомогою нього може досягати значень більше 90%. Також можливе зберігання та пошук зображень у великих базах даних, реконструкція зображень. Але сам процес обчислення є дуже трудомістким. Зображення слід отримувати в умовах доброго освітлення, повинна проводитися якісна попередня обробка, приводячи зображення до стандартних умов.

2.2.4 Методи на основі штучних нейронних мереж

Перевагою використання штучних нейронних мереж для виявлення облич є можливість навчання систем для вибору ключових характеристик облич на навчальних вибірках.

Нейронні мережі забезпечують можливість отримання класифікатора добре моделюючого складну функцію розподілу зображень облич.

У задачах класифікації при цьому відбувається неявне присвоєння ключових ознак у межах мережі, визначення значущості цих ознак та систем взаємних залежностей між ними.

Серед нейронних мереж для вирішення завдань розпізнавання образів найчастіше застосовуються багатосарові персептрони зі зворотним поширенням помилки, мережі з радіальнобазисною функцією й згорткові нейронні мережі (Convolutional Neural Networks) [52, 53]. Успіх обумовлений можливістю враховування двовимірної топології зображення, на відміну від багатосарового персептрона.

Відмінними рисами згорткової нейронної мережі є локальні рецепторні поля (забезпечують локальну двовимірну зв'язність нейронів), загальні ваги (забезпечують детектування деяких рис в будь-якому місці зображення) і ієрархічна організація з просторовими семплінгом (spatial subsampling). Завдяки цим нововведенням згорткова нейронна мережа забезпечує часткову

стійкість до змін масштабу, зсувів, поворотів, змін ракурсу і іншим перетворенням (рис. 2.6).

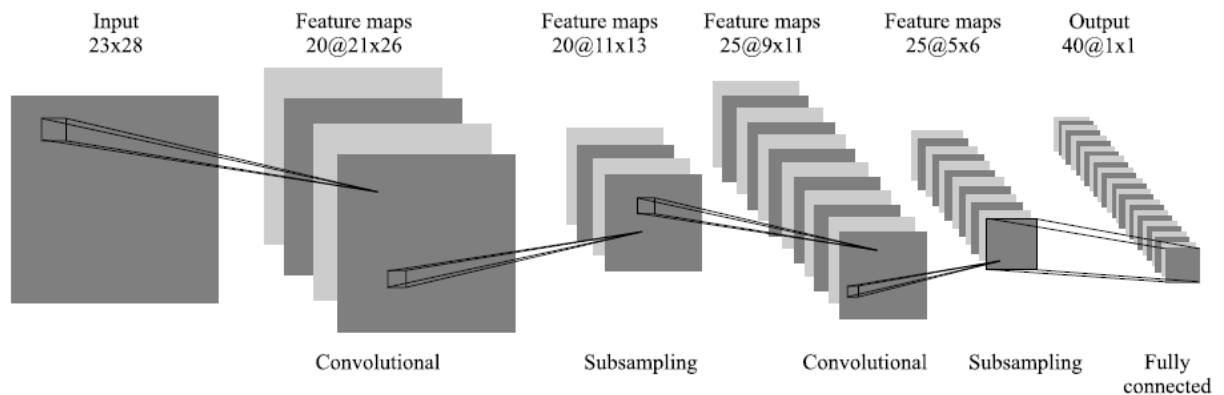


Рисунок 2.6 – Схематичне зображення архітектури згорткової нейронної мережі

Нейронні мережі можуть бути з успіхом застосовані для виявлення облич на зображеннях та верифікації персони за обличчям. При цьому якість розпізнавання знижується при збільшенні кількості класів, які необхідно передбачити.

Недоліки:

- необхідність повного перенавчання мережі на всьому наявному наборі у випадку додавання до бази даних нового еталонного обличчя;
- необхідна велика кількість часу для навчання.

2.3 Проблема вибору та класифікації інформативних ознак на зображенні

Вибір ознак є необхідним етапом при побудові систем класифікації. Вдале рішення даного завдання забезпечує як зниження розмірності вектора вимірювань і опису об'єктів, так і підвищення ефективності системи класифікації в цілому.

Нехай вся множина зображень Ω в базі даних (вибірці зображень, використовуваної для відбору ознак) підрозділяється на групи (класи):

$$\Omega = \bigcup_{l=0}^{L-1} \Omega_l, \quad (2.4)$$

кожний клас Ω_l складається з N зображень $X^{(n,l)}$:

$$\Omega_l = \bigcup_{n=0}^{N_l-1} X^{(n,l)}. \quad (2.5)$$

Кожне цифрове зображення можна розглядати як безліч числових ознак, заданих у просторі, розмірність якого визначається числом пікселів зображення. Проте класифікація об'єктів (зображень) в такому просторі практично неможлива внаслідок ряду причин. По-перше, уявлення зображення є надмірним, його ознаки є сильно залежними. По-друге, навчання багатопараметричних класифікаторів для великої розмірності ознак вимагає значних ресурсів (часових, обсягу навчальної вибірки і так далі), причому якість таких класифікаторів при реальному застосуванні може не задовольняти заданим вимогам. По-третє, пошук зображення в великих базах даних з використанням великої кількості ознак потребує невиправдано великих обчислювальних ресурсів. Під час перетворення в інший простір без зниження його розмірності (наприклад, шляхом спектрального перетворення) ці проблеми залишаються незмінними.

Вибір ознак може здійснюватися без урахування зв'язку з якістю класифікації, коли ознаки вибираються, виходячи з мінімізації (максимізації) деякого критерію [54]. Більш кращою є процедура відбору ознак з використанням кордону Чернова або відстані Бхатачарія, однак застосовність цих критеріїв обмежується колом завдань з відомими щільностями ймовірностей [55]. У разі відомих розподілів ознак можна також використовувати поняття дивергенції і ентропії [56, 57]. У тому випадку, коли розподіли ознак для класів є невідомими, найкращим вибором є очевидний

критерій вірогідності помилки класифікації, оцінюваний безпосереднім чином.

Для відбору інформативних ознак, які забезпечують вирішення задачі розпізнавання, існує ще один метод, що складається з трьох етапів:

- попередній відбір інформативних ознак;
- формування множин інформативних ознак;
- остаточний відбір інформативних ознак.

На першому етапі проводиться попередній відбір ознак. Попередній відбір інформативних ознак заснований на дисперсійному аналізі і дозволяє знизити розмірність простору ознак з декількох сотень тисяч ознак (значень пікселів) до декількох десятків або сотень ознак.

На другому етапі з використанням методу приєднання-відкидання формуються найбільш інформативні множини ознак. Сформований на першому етапі набір ознак (визначається набором індексів пікселів на полях ознак, які розраховуються за заданим алгоритмом) є сильно надмірним, однак він може служити основою для подальшого відбору ознак. Мета другого етапу відбору ознак – знайти різні варіанти комбінацій ознак, що забезпечують найкращі результати класифікації для різних типів класифікаторів та критеріїв класифікації. Повний перебір всіх можливих комбінацій ознак є практично нереалізованим завданням. Тому зазвичай замість повного перебору використовуються комбінації алгоритмів послідовного приєднання і послідовного відкидання ознак.

На третьому етапі з використанням критерію класифікації вибирається одне або кілька множин ознак, що забезпечують найбільш ефективне рішення задачі класифікації. Спочатку для заданих еталонної та контрольної вибірок зображень розраховуються масиви векторів ознак. Потім розглядаються всі множини ознак, отримані на другому етапі відбору ознак. При цьому для кожного множини відомий вид класифікатора і критерію класифікації, при яких ця множина була відібрана. Для кожної такої множини проводиться навчання класифікатора і перевірка результату з використанням еталонної та

контрольної вибірок. Наприкінці обирається одна (або декілька) множина ознак, виходячи з результатів тестування і застосовуваного критерію класифікації (рис. 2.7).

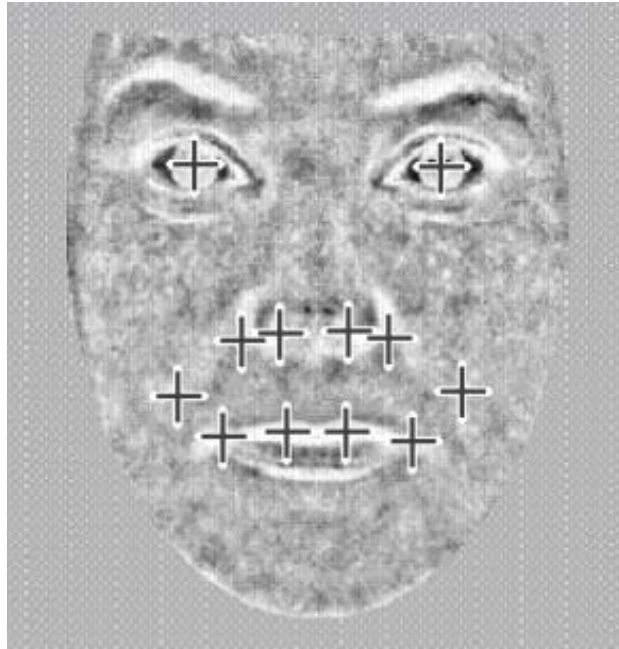


Рисунок 2.7 – Результат розрахованих інформативних ознак на тестовому зображенні

Описаний метод відбору інформативних ознак заснований на розрахунку як локальних, так і глобальних ознак, а також на використанні алгоритму приєднання-відкидання ознак і критеріїв класифікації є досить ефективним для задач розпізнавання на зображеннях.

2.4 Розроблення методики виявлення та класифікації подібності рис людських облич

Проаналізувавши велику кількість літературних джерел та, дослідивши розглянуті методи, можна зробити певні висновки щодо реалізації розробки застосунку для виявлення та класифікації подібності рис людських облич.

Для того, щоб розпізнати обличчя людини на зображенні, по-перше, потрібно виявити, де саме на фото розташоване обличчя людини. Для цього потрібно виділити його основні ключові компоненти, такі як очі, ніс, губи, лоб і так далі. Це можна здійснити за допомогою шаблони – примітиви Хаара (рис. 2.8).

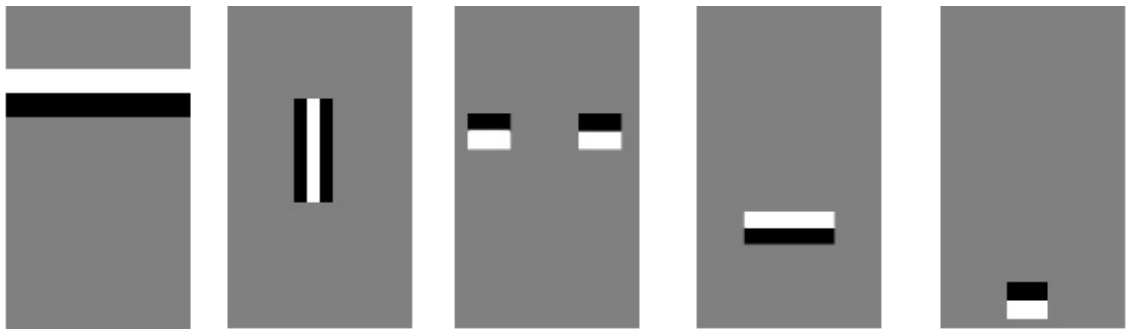


Рисунок 2.8 – Приклади примітивів Хаара

Якщо шаблони відповідають певним областям на зображенні, припускається, що зображення містить людське обличчя. Для кожного з шаблонів розраховується різниця між яскравістю білої та чорної областей. Це значення порівнюється з еталоном, після чого приймається рішення про те, чи є частина людського обличчя на зображенні, чи ні.

Обраний метод для реалізації – метод Віоли-Джонса (каскади Хаара). Для того, щоб обличчя різних розмірів можна було виявляти на зображенні, застосовується метод сканувального вікна. Саме у цьому вікні розраховуються усі примітиви. Воно за певну кількість ітерацій проходить по зображенню та обробляє його. У випадку, коли обличчя не було виявлено, сканувальне вікно збільшується та проходиться по зображенню ще раз. Робота методу не закінчується, поки обличчя не виявляється, чи розмір сканувального вікна не стає рівним розміру самого зображення.

Після того, як обличчя на зображенні було виявлено, його необхідно ідентифікувати. Для вирішення цього завдання буде використовуватися метод Local Binary Patterns. Суть його полягає в тому, що зображення розбивається на частини. І в кожній такій частині кожен піксель порівнюється з сусідніми 8

пікселями. Якщо значення центрального пікселя більше сусіднього, то пишеться 0, в іншому випадку – 1. І так для кожного пікселя на зображенні розраховується певне значення (рис. 2.9).

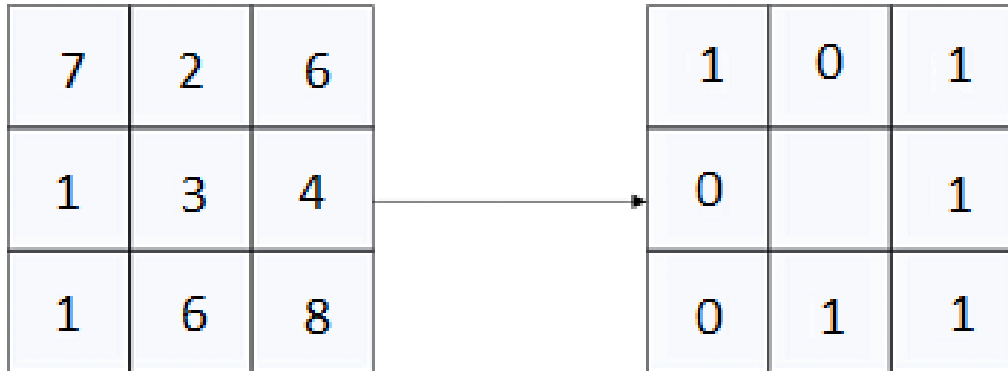


Рисунок 2.9 – Приклад результату методу Local Binary Patterns

Далі на основі цих значень для всіх частин, на які було розбите зображення, будується гистограма. Усі гистограми з усіх частин, на які розбивалося зображення, об'єднуються в один вектор, який потім характеризує зображення в цілому. Якщо необхідно дізнатися, наскільки схожі дві особи, потрібно обчислити для кожного з них такий вектор і порівняти їх.

Вибір даного методу для реалізації розробки застосунку аргументується тим, що він є дуже ефективним, його результати розпізнавання отримуються дуже швидко, також є можливість виявлення більше одного обличчя. Ще одним позитивним фактором є те, що в бібліотеці OpenCV метод частково реалізований. Але із недоліків можна зазначити те, що він складним для навчання, так як для навчання потрібна велика кількість тестових даних і передбачає великий час навчання.

3 РОЗРОБЛЕННЯ ЗАСТОСУНКУ ДЛЯ ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЇ ПОДІБНОСТІ ЛЮДСЬКИХ ОБЛИЧ

3.1 Вибір інструментальних засобів для створення застосунку

У рамках кваліфікаційної роботи був розроблений застосунок для виявлення схожості обраних фото людей із еталонами-знаменитостями. Для реалізації було обране середовище JetBrains PyCharm. Це обумовлено тим, що дане середовище є найпотужнішим і актуальним інструментом для розроблення застосунків на мові програмування Python.

Для реалізації розробки застосунку найчастіше використовувалася бібліотека OpenCV. OpenCV – це open source бібліотека комп'ютерного зору, яка призначена для аналізу, класифікації та обробки зображень. Широко використовується в таких мовах програмування як C, C++, Python і Java. Вона має безліч інструментарію для роботи з зображеннями, представляє собою прості способи маніпулювання колірними просторами.

OpenCV має такий доступний функціонал:

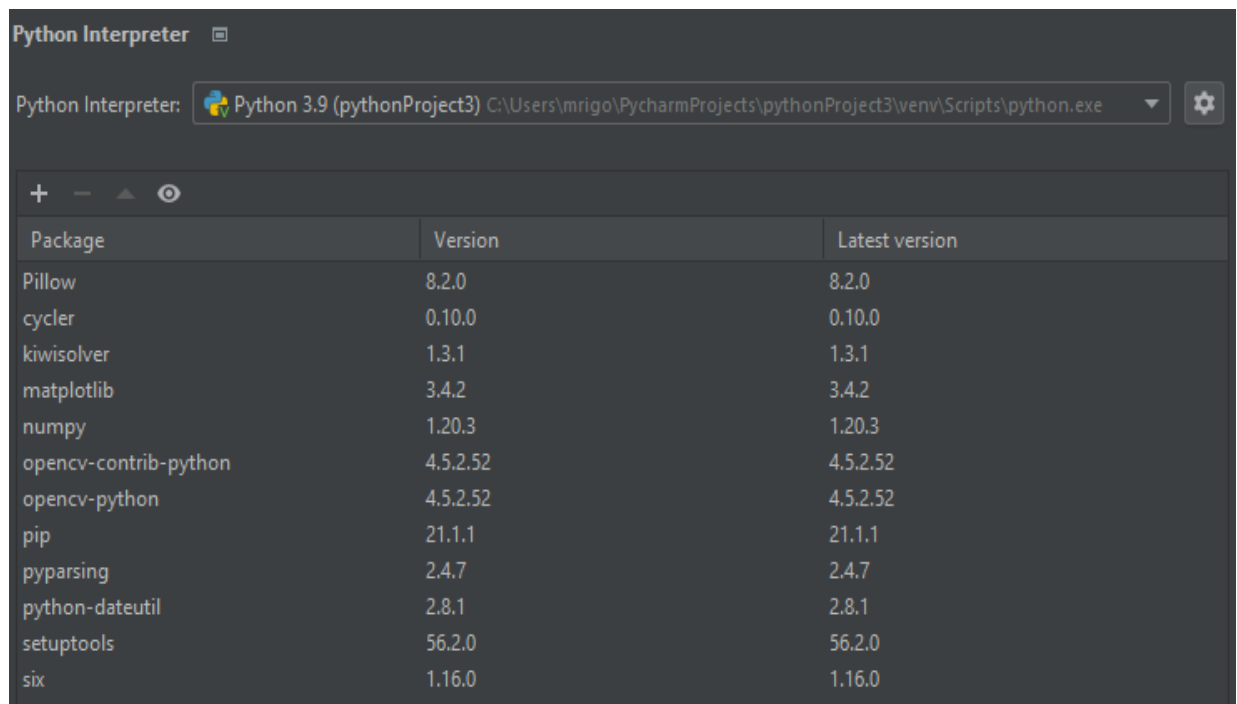
- імпорт та перегляд зображень;
- обрізання зображень;
- зміна розміру;
- повертання зображень;
- градіації сірого та поріг;
- розмиття та згладжування;
- малювання прямокутників;
- малювання ліній;
- текст на зображеннях;
- виявлення об'єктів та облич;
- розпізнавання об'єктів та облич;
- збереження зображень.

Також були використані бібліотеки для Python – NumPy та PIL. Перша з яких додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих і дуже швидких математичних функцій для операцій з цими масивами. А друга – бібліотека, призначена для роботи з растровою графікою.

3.2 Етапи розроблення застосунку для виявлення та класифікації подібності рис людських облич

3.2.1 Встановлення програмного забезпечення

Першим етапом розроблення застосунку є встановлення усього необхідного програмного забезпечення, переліченого в попередньому підрозділі. Було встановлено актуальну версію мови програмування Python та IDE PyCharm. Також були встановлені усі необхідні бібліотеки для роботи над розробкою, серед яких NumPy та PIL (рис. 3.1).



The screenshot shows the 'Python Interpreter' window in PyCharm. At the top, it displays 'Python Interpreter: Python 3.9 (pythonProject3) C:\Users\mriga\PycharmProjects\pythonProject3\venv\Scripts\python.exe'. Below this is a table of installed packages:

Package	Version	Latest version
Pillow	8.2.0	8.2.0
cycler	0.10.0	0.10.0
kiwisolver	1.3.1	1.3.1
matplotlib	3.4.2	3.4.2
numpy	1.20.3	1.20.3
opencv-contrib-python	4.5.2.52	4.5.2.52
opencv-python	4.5.2.52	4.5.2.52
pip	21.1.1	21.1.1
pyarsing	2.4.7	2.4.7
python-dateutil	2.8.1	2.8.1
setuptools	56.2.0	56.2.0
six	1.16.0	1.16.0

Рисунок 3.1 – Скріншот застосованих для розробки бібліотек

3.2.2 Знаходження еталонних зображень та зображень людей, які будуть порівнюватися з еталонами

Для даної кваліфікаційної роботи було обрано 3 знаменитості, фото яких будуть використовуватися як зображення-еталони для навчання. Цими знаменитостями є:

– Ілон Маск (видатний інженер, винахідник і підприємець, співзасновник компанії PayPal, засновник компаній SpaceX, Tesla, SolarCity і The Boring Company);

– Марія Кюрі (французький вчений польського походження, була першою жінкою-вченим – нобелівським лауреатом в історії і першим двічі нобелівським лауреатом в історії);

– Кріштіану Роналду (один із найвідоміших гравців сучасного футболу, виступав за велику кількість клубів, за збірну команду Португалії, він встановив не один десяток рекордів).

Для більшої точності розпізнавання було знайдено по 3-5 фотографії-еталона кожної знаменитості та поміщено у папку зображень для навчання програми.

Було прийняте рішення, що із еталонами будуть порівнюватися фото колег з потоку та друзів. Для цього було зібрано 15 фотографій різних людей, які потім помістилися в іншу відповідну папку.

3.2.3 Написання детектору та ідентифікатора облич

Для виявлення облич створюється об'єкт `faceCascade`, який використовує змінну `cascadePath`. Змінна `cascadePath` містить ім'я файлу з уже готовими значеннями для розпізнавання облич. Цей файл також можна взяти з директорії з `OpenCV`.

Далі для розпізнавання облич створюється об'єкт recognizer за допомогою вбудованої функції OpenCV – LBPHFaceRecognizer_create, у якій першим та другим параметром задається область аналізу зображення по пікселям, наступними двома параметрами задаються розміри цих областей, а останнім – порогове значення для розпізнавання обличчя (рис. 3.2).

```
# Для детектирования лиц используем каскады Хаара
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath)

# Для распознавания используем локальные бинарные шаблоны
recognizer = cv2.face.LBPHFaceRecognizer_create(2, 16, 4, 4, 23)
```

Рисунок 3.2 – Скріншот коду детектору та ідентифікатора облич

3.2.4 Навчання програми розпізнавати обличчя

Для того, щоб програма могла розпізнавати обличчя, їй необхідно проаналізувати всі зображення-еталони. Для цього була написана функція, яка повертає усі зображення облич, які були знайдені за вказаною директорією. Ця функція зчитує кожену фотографію і повертає саме ту область, де знаходиться обличчя.

Функція faceCascade.detectMultiScale визначає області на фотографії, де є людські обличчя. Вона повертає список з параметрами [x, y, w, h] для кожної знайденої особи (рис. 3.3). Ці параметри описують прямокутну область в тому місці, де знайшлося обличчя. Вона приймає такі параметри:

- image (саме зображення);
- scaleFactor (визначає те, на скільки буде збільшуватися скануюче вікно пошуку на кожній ітерації);
- minNeighbors (ступінь фільтрування пошуку обличчя);
- minSize (мінімальний розмір обличчя на фото).

```

def get_images(path):
    # Ищем все фотографии и записываем их в image_paths
    image_paths = [os.path.join(path, f) for f in os.listdir(path)]
    images = []
    labels = []
    names = []
    for image_path in image_paths:
        # Переводим изображение в черно-белый формат и приводим его к формату массива
        gray = Image.open(image_path).convert('L')
        image = np.array(gray, 'uint8')
        # Из каждого имени файла извлекаем номер человека, изображенного на фото
        subject_number = int(os.path.split(image_path)[1].split("-")[0])
        subject_name = os.path.split(image_path)[1].split("-")[1].split("(")[0]
        print("Face saved: name -", subject_name, "; number -", subject_number)

        # Определяем области где есть лица
        faces = faceCascade.detectMultiScale(image, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
        # Если лицо нашлось добавляем его в список images, а соответствующий ему номер в список labels
        for (x, y, w, h) in faces:
            images.append(image[y: y + h, x: x + w])
            labels.append(subject_number)
            names.append(subject_name)
            # В окне показываем изображение
            cv2.imshow('Training', image[y: y + h, x: x + w])
            cv2.waitKey(100)
    return images, labels, names

```

Рисунок 3.3 – Скріншот коду функції, яка знаходить обличчя на фото за вказаною директорією

Далі слідує виклик написаної функції для виявлення та зберігання зображень-еталонів у пам'яті комп'ютера (рис. 3.4).

```

# Путь к фотографиям
path = './etalons'
# Получаем лица и соответствующие им номера
cv2.namedWindow('Training', cv2.WINDOW_NORMAL)
cv2.waitKey(1000)
images, labels, names = get_images(path)
cv2.destroyAllWindows()

# Обучаем программу распознавать лица
recognizer.train(images, np.array(labels))
print("the system is trained")

```

Рисунок 3.4 – Скріншот коду навчання програми

На цьому етапі вважається, що програма успішно навчилася розпізнавати обличчя, так як в неї вже є дані, з якими вона потім буде порівнювати інші зображення та виконувати прийняття рішення.

3.2.5 Підготовка зображень для розпізнавання

Далі виконуються схожі операції, що і при навчанні програми. Але тепер обличчя виявляються серед зображень, які будуть розпізнаватися пізніше за допомогою тієї ж функції `faceCascade.detectMultiScale` (рис. 3.5), але з іншими параметрами, так як ці зображення відрізняються від еталонів форматом та розміром. Параметри підібрані для кращого виявлення (рис. 3.6).

```
def detectMultiScale(self, image, scaleFactor=None, minNeighbors=None, flags=None, minSize=None, maxSize=None):
    """
    detectMultiScale(image[, scaleFactor[, minNeighbors[, flags[, minSize[, maxSize]]]]) -> objects
    . @brief Detects objects of different sizes in the input image. The detected objects are returned as a list
    . of rectangles.
    .
    . @param image Matrix of the type CV_8U containing an image where objects are detected.
    . @param objects Vector of rectangles where each rectangle contains the detected object, the
    . rectangles may be partially outside the original image.
    . @param scaleFactor Parameter specifying how much the image size is reduced at each image scale.
    . @param minNeighbors Parameter specifying how many neighbors each candidate rectangle should have
    . to retain it.
    . @param flags Parameter with the same meaning for an old cascade as in the function
    . cvHaarDetectObjects. It is not used for a new cascade.
    . @param minSize Minimum possible object size. Objects smaller than that are ignored.
    . @param maxSize Maximum possible object size. Objects larger than that are ignored. If 'maxSize == min
    .
    . The function is parallelized with the TBB library.
    .
    . @note
    . - (Python) A face detection example using cascade classifiers can be found at
    .   opencv\_source\_code/samples/python/faceDetect.py
    """
    pass
```

Рисунок 3.5 – Скріншот визначення функції
`cv2.CascadeClassifier.detectMultiScale`

```
# Создаем список фотографий для распознавания
path = './test_samples'
image_paths = [os.path.join(path, f) for f in os.listdir(path)]

cv2.namedWindow('Recognizing', cv2.WINDOW_NORMAL)
cv2.waitKey(1000)

for image_path in image_paths:
    # Ищем лица на фотографиях
    gray = Image.open(image_path).convert('L')
    image = np.array(gray, 'uint8')
    faces = faceCascade.detectMultiScale(image, scaleFactor=1.08, minNeighbors=10, minSize=(100, 100))
```

Рисунок 3.6 – Скріншот підготовки зображень для розпізнавання

3.2.6 Вивід результатів щодо порівняння зображень людей із еталонами

Програма кожний раз при роботі з зображенням виводить його на екран. Реалізоване це за допомогою функцій бібліотеки OpenCV, а саме – `cv2.namedWindow` (рис. 3.7), `cv2.waitKey` (рис. 3.8) та `cv2.imshow` (рис. 3.9). Також на кожному етапі роботи програми виконується вивід до консолі з використанням базової функції `print`, який повідомляє додаткову важливу інформацію. Сам результат наприкінці виводиться до консолі для зображень, для яких програма змогла знайти схожий еталон (рис. 3.10).

```
def namedWindow(winname, flags=None): # real signature unknown; restored from __doc__
    """
    namedWindow(winname[, flags]) -> None
    . @brief Creates a window.
    .
    . The function namedWindow creates a window that can be used as a placeholder for images and
    . trackbars. Created windows are referred to by their names.
    . If a window with the same name already exists, the function does nothing.
    . You can call cv::destroyWindow or cv::destroyAllWindows to close the window and de-allocate any associated
    . memory usage. For a simple program, you do not really have to call these functions because all the
    . resources and windows of the application are closed automatically by the operating system upon exit.
    . Qt backend supports additional flags:
    . - **WINDOW_NORMAL or WINDOW_AUTOSIZE:** WINDOW_NORMAL enables you to resize the
    . window, whereas WINDOW_AUTOSIZE adjusts automatically the window size to fit the
    . displayed image (see imshow ), and you cannot change the window size manually.
    . - **WINDOW_FREERATIO or WINDOW_KEEPRATIO:** WINDOW_FREERATIO adjusts the image
    . with no respect to its ratio, whereas WINDOW_KEEPRATIO keeps the image ratio.
    . - **WINDOW_GUI_NORMAL or WINDOW_GUI_EXPANDED:** WINDOW_GUI_NORMAL is the old way to draw the window
    . without statusbar and toolbar, whereas WINDOW_GUI_EXPANDED is a new enhanced GUI.
    . By default, flags == WINDOW_AUTOSIZE | WINDOW_KEEPRATIO | WINDOW_GUI_EXPANDED
    .
    . @param winname Name of the window in the window caption that may be used as a window identifier.
    . @param flags Flags of the window. The supported flags are: (cv::WindowFlags)
    """
    pass
```

Рисунок 3.7 – Скріншот визначення функції `cv2.namedWindow`

```

def waitKey(delay=None): # real signature unknown; restored from __doc__
    """
    waitKey([, delay]) -> retval
    . @brief Waits for a pressed key.
    .
    . The function waitKey waits for a key event infinitely (when  $\text{delay} \leq 0$ ) or for delay
    . milliseconds, when it is positive. Since the OS has a minimum time between switching threads, the
    . function will not wait exactly delay ms, it will wait at least delay ms, depending on what else is
    . running on your computer at that time. It returns the code of the pressed key or -1 if no key was
    . pressed before the specified time had elapsed. To check for a key press but not wait for it, use
    . #pollKey.
    .
    . @note The functions #waitKey and #pollKey are the only methods in HighGUI that can fetch and handle
    . GUI events, so one of them needs to be called periodically for normal event processing unless
    . HighGUI is used within an environment that takes care of event processing.
    .
    . @note The function only works if there is at least one HighGUI window created and the window is
    . active. If there are several HighGUI windows, any of them can be active.
    .
    . @param delay Delay in milliseconds. 0 is the special value that means "forever".
    """
    pass

```

Рисунок 3.8 – Скріншот визначення функції cv2.waitKey

```

def imshow(winname, mat): # real signature unknown; restored from __doc__
    """
    imshow(winname, mat) -> None
    . @brief Displays an image in the specified window.
    . The function imshow displays an image in the specified window. If the window was created with the
    . cv::WINDOW_AUTOSIZE flag, the image is shown with its original size, however it is still limited by the screen.
    . Otherwise, the image is scaled to fit the window. The function may scale the image, depending on its depth:
    . - If the image is 8-bit unsigned, it is displayed as is.
    . - If the image is 16-bit unsigned or 32-bit integer, the pixels are divided by 256. That is, the
    . value range [0,255*256] is mapped to [0,255].
    . - If the image is 32-bit or 64-bit floating-point, the pixel values are multiplied by 255. That is, the
    . value range [0,1] is mapped to [0,255].
    . If window was created with OpenGL support, cv::imshow also support ogl::Buffer , ogl::Texture2D and
    . cuda::GpuMat as input.
    . If the window was not created before this function, it is assumed creating a window with cv::WINDOW_AUTOSIZE
    . If you need to show an image that is bigger than the screen resolution, you will need to call namedWindow("
    . @note This function should be followed by a call to cv::waitKey or cv::pollKey to perform GUI
    . housekeeping tasks that are necessary to actually show the given image and make the window respond
    . to mouse and keyboard events. Otherwise, it won't display the image and the window might lock up.
    . For example, **waitKey(0)** will display the window infinitely until any keypress (it is suitable
    . for image display). **waitKey(25)** will display a frame and wait approximately 25 ms for a key
    . press (suitable for displaying a video frame-by-frame). To remove the window, use cv::destroyWindow.
    . @param winname Name of the window.
    . @param mat Image to be shown.
    """
    pass

```

Рисунок 3.9 – Скріншот визначення функції cv2.imshow

```

for (x, y, w, h) in faces:
    # Если лица найдены, пытаемся распознать их
    # Функция recognizer.predict в случае успешного распознавания возвращает номер и параметр confidence,
    # этот параметр указывает на уверенность алгоритма, что это именно тот человек, чем он меньше, тем больше
    number_predicted, conf = recognizer.predict(image[y: y + h, x: x + w])

    # Извлекаем настоящий номер человека на фото и сравниваем с тем, что выдал алгоритм
    name_actual = os.path.split(image_path)[1].split(".")[0]

    if number_predicted > 0:
        cur_name = ""
        i = 0
        for el in labels:
            if labels[i] == number_predicted:
                cur_name = names[i]
                break
            i += 1
        print(name_actual, "is", conf, "% similar to", cur_name)

cv2.imshow("Recognizing", image[y: y + h, x: x + w])
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Рисунок 3.10 – Скріншот коду для виводу результатів щодо порівняння людей із еталонами

3.3 Тестування розробленого застосунку та аналіз результатів

Тестування в області розробки програмного забезпечення – це процес оцінки того, що усі частини програми ведуть себе так, як і планувалося.

Тести виконують перевірки програмного забезпечення, перевіряючи, що перевірений результат відповідає специфікаціям, враховуючи різні вхідні дані.

Автоматизовані тести діляться на різні типи. Кожен тип тесту має свою мету (призначення) і область дії:

- end-to-end-тести або наскрізні тести, як їх називають при перекладі (тестують систему в цілому, емулюючи реальний призначений для користувача простір);

- інтеграційні тести (мета даних тестів полягає в тому, щоб переконатися, що кілька залежних один від одного модулів коду працюють разом, і їх взаємодія працює належним чином);

- модульні тести: вони тестують певну функціональність ізольовано.

В рамках кваліфікаційної автоматизовані тести не використовувалися.

Фінальна версія програми вже сконфігурована для стабільної роботи та використання. Усі інші маніпуляції проводяться для визначення, як саме нестандартні зміни та дії користувача віддзеркаляться на результаті, що зробить програма. Також тестування проводиться для того, щоб визначити переваги та недоліки обраної методики для реалізації у нестандартних умовах.

3.3.1 Тестування роботи програми при розпізнаванні людини, фотографія якої була використана як еталон для навчання

Для кожного з трьох обраних знаменитостей була завантажена додаткова фотографія, але вже поміщена не в папку для еталонів, а в папку, яка призначена для зберігання фотографій людей, яких програма буде порівнювати із еталонами. Результати роботи програми приведені на скріншотах (рис. 3.11–3.13).

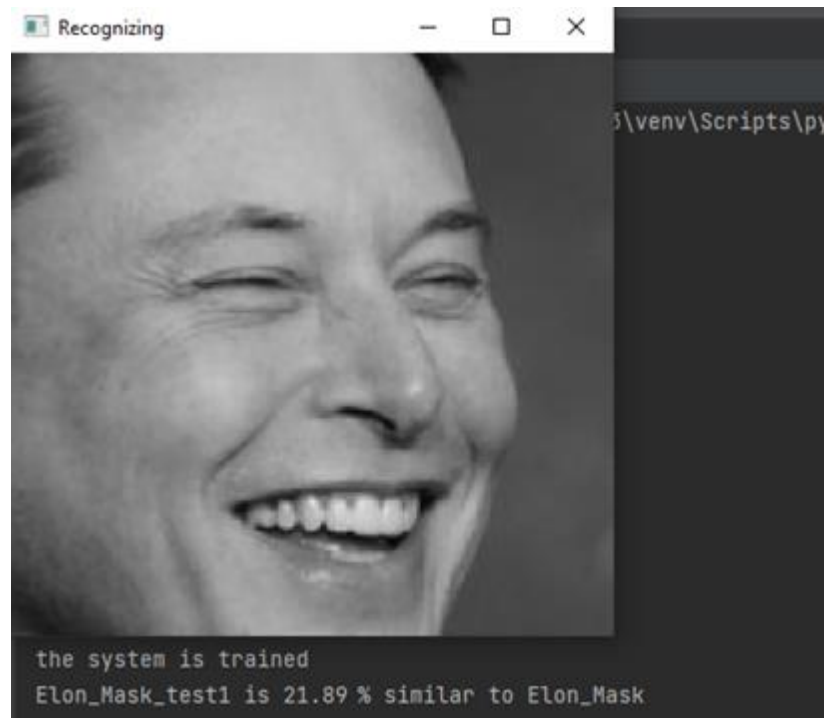


Рисунок 3.11 – Скріншот результату роботи програми при порівнянні фотографії Ілона Маска з його зображеннями-еталонами

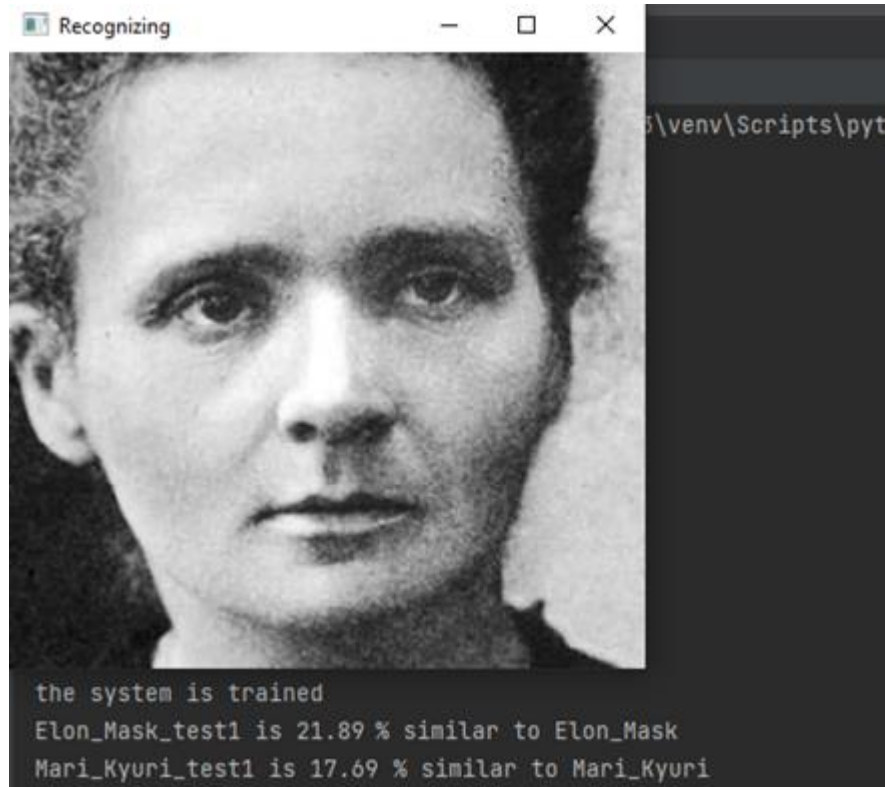


Рисунок 3.12 – Скріншот результату роботи програми при порівнянні фотографії Марії Кюрі з її зображеннями-еталонами

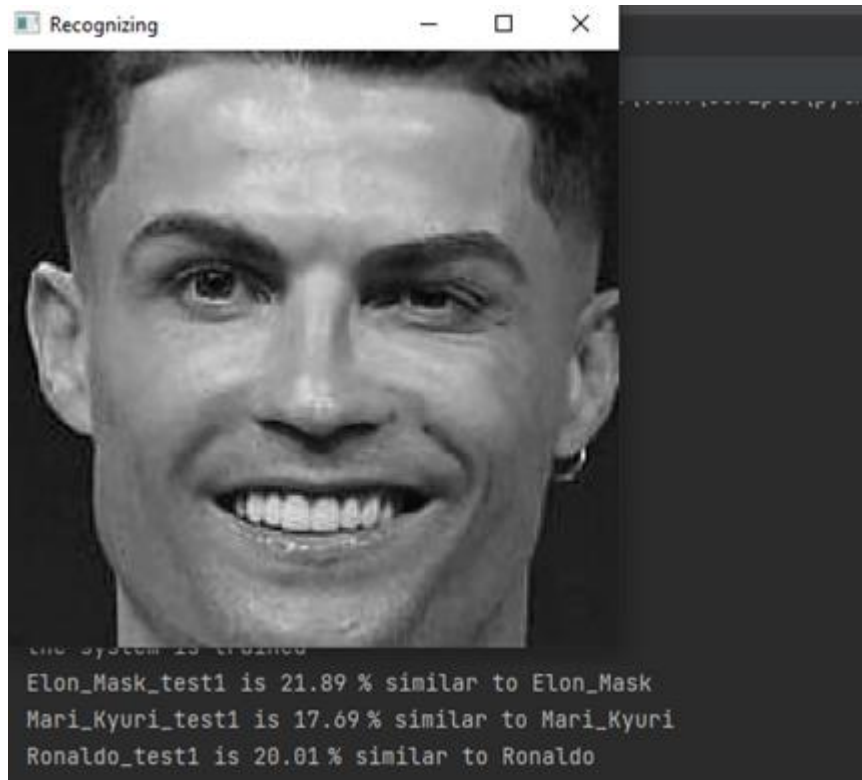


Рисунок 3.13 – Скріншот результату роботи програми при порівнянні фотографії Кріштіану Роналду з його зображеннями-еталонами

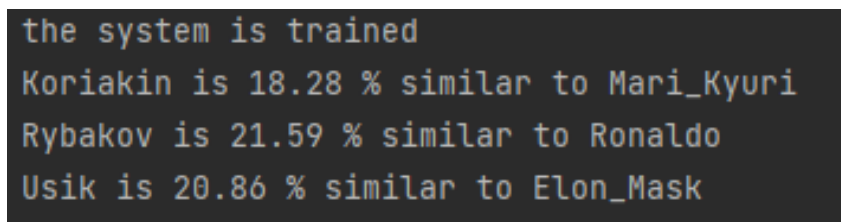
Як видно, програма коректно розпізнає обличчя знаменитостей, дане тестування на результат негативно не вплинуло. Сумнів програми аргументований тим, що вираз обличчя знаменитостей сильно відрізняється від їх зображень-еталонів. У випадку із Марією Кюрі на точність ще вплинула якість самої фотографії. Однак програма все одно правильно ідентифікувала особистості.

3.3.2 Тестування роботи програми з різною кількістю фотографій, використаних як зображення-еталони

Для цього тесту було запущено розроблену програму 2 рази. Першого разу всі тестові фотографії (додаток А) порівнювалися з єдиними фотографіями-еталонами знаменитостей (рис. 3.14). На другий раз ці ж тестові фотографії порівнювалися з тими ж знаменитостями, але фотографій-еталонів кожної було по 3-5 штук (рис. 3.15). Результат вийшов такий (рис 3.16–3.21):

- у випадку, коли фотографій-еталонів було по 1 для кожної знаменитості, менша кількість тестових фотографій були розпізнані програмою як схожі на знаменитість;

- при більшій кількості фотографій-еталонів знаменитостей кількість розпізнаних облич на тестових фотографіях ставала більшою та упевненість програми ставала більшою, але час, необхідний для навчання програми, також зростає.



```
the system is trained
Koriakin is 18.28 % similar to Mari_Kyuri
Rybakov is 21.59 % similar to Ronaldo
Usik is 20.86 % similar to Elon_Mask
```

Рисунок 3.14 – Скріншот результату роботи програми при порівнянні фотографій із 3 фотографіями-еталонами

```

the system is trained
Koriakin is 18.28 % similar to Mari_Kyuri
Rybakov is 21.21 % similar to Ronaldo
Smirnov is 22.84 % similar to Ronaldo
Usik is 20.86 % similar to Elon_Mask
Uvarova is 22.32 % similar to Ronaldo
Zhadan is 21.24 % similar to Ronaldo

```

Рисунок 3.15 – Скріншот результату роботи програми при порівнянні фотографій із 15 фотографіями-еталонами

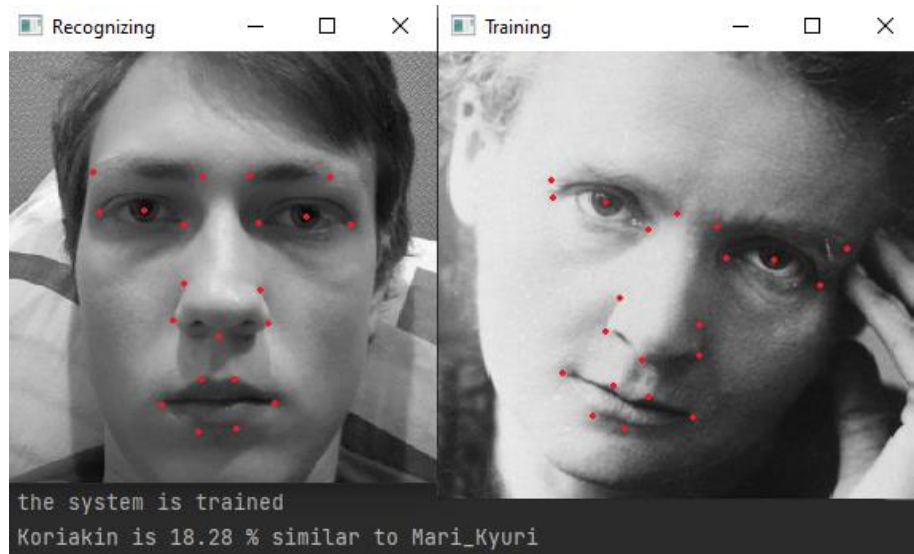


Рисунок 3.16 – Перший результат порівняння тестової фотографії з еталонами

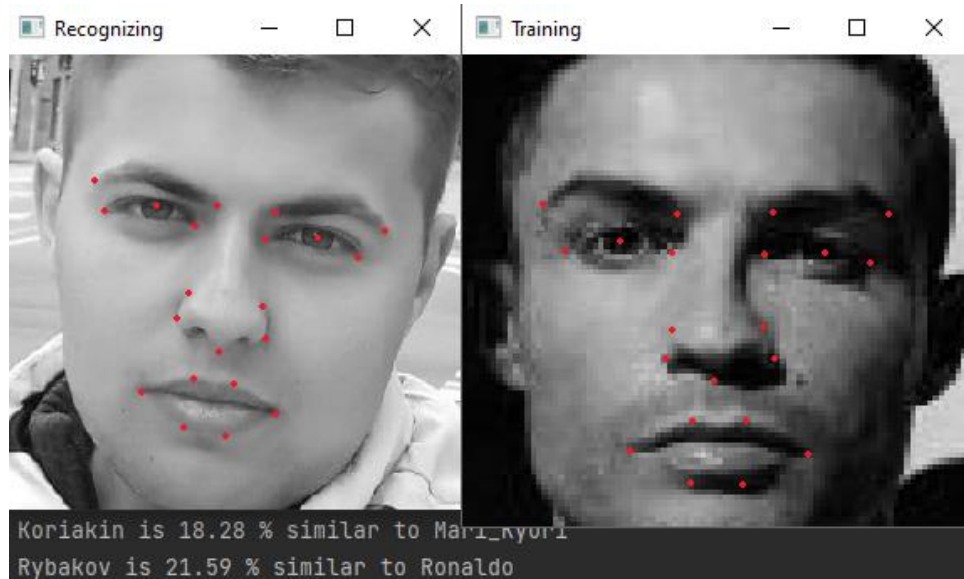


Рисунок 3.17 – Другий результат порівняння тестової фотографії з еталонами

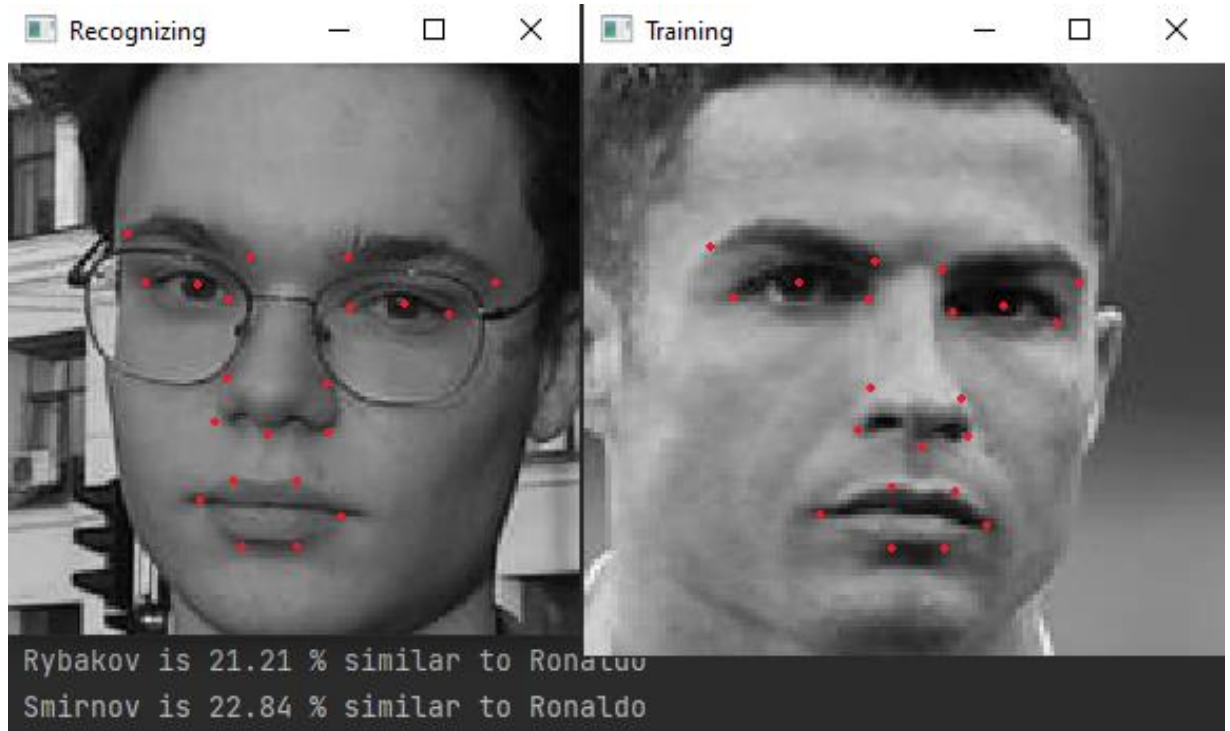


Рисунок 3.18 – Третій результат порівняння
тестової фотографії з еталонами

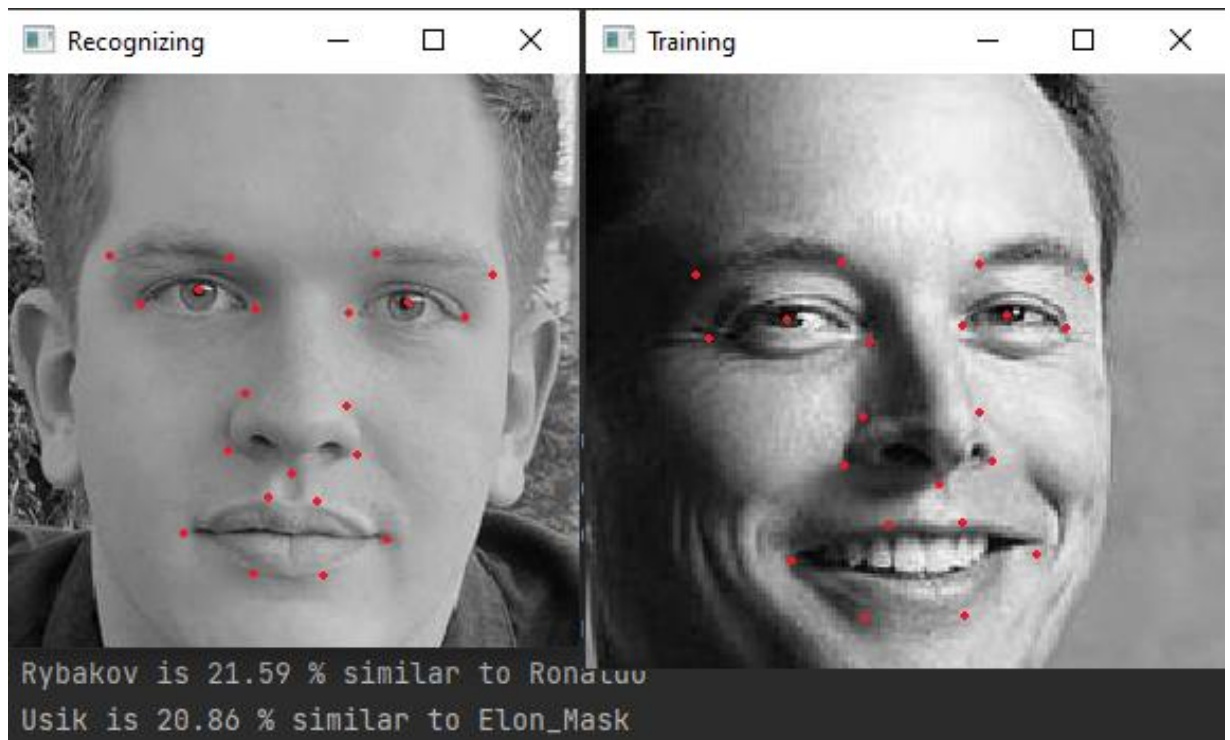


Рисунок 3.19 – Четвертий результат порівняння
тестової фотографії з еталонами

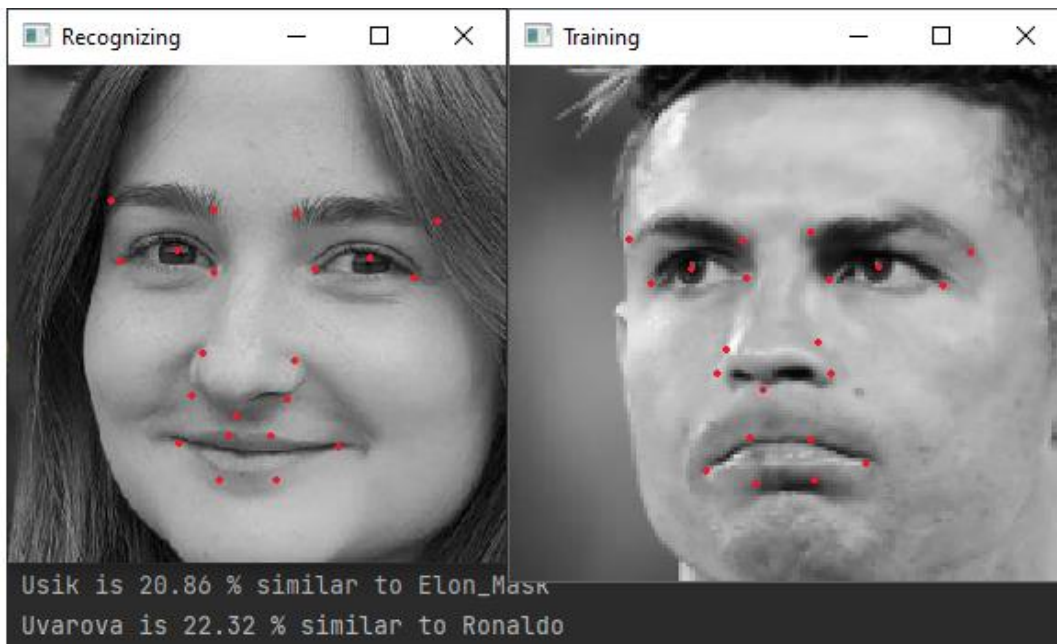


Рисунок 3.20 – П'ятий результат порівняння тестової фотографії з еталонами

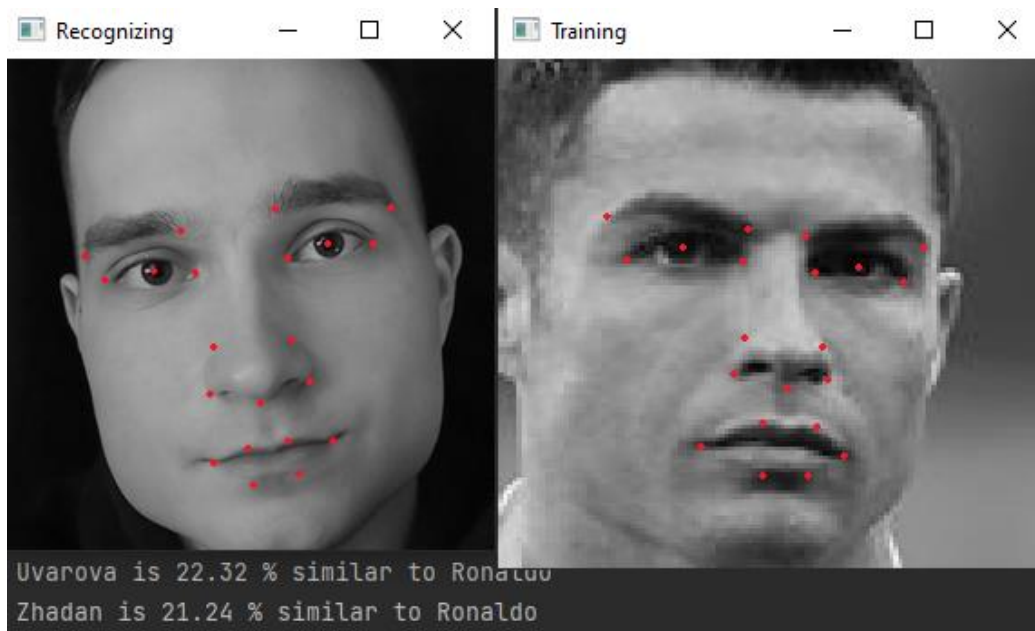


Рисунок 3.21 – Шостий результат порівняння тестової фотографії з еталонами

Дане тестування показало, що додавання зображень-еталонів позитивно віддзеркалюється на точності розпізнавання, але це в той же час збільшує кількість часу, необхідну для навчання програми, що і було підтверджено на практиці (додаток Б). Цей тест було пройдено успішно.

3.3.3 Тестування роботи програми з різними типами вхідних даних

Вхідні дані для даного програмного застосунку – будь-які зображення (фото). Ці зображення можуть бути різних форматів. Були успішно протестовані для відкривання програмою такі формати зображень:

- JPEG (JPG);
- JPEG 2000;
- PNG;
- GIF;
- RAW;

Реалізовано це завдяки актуальним та гнучким бібліотекам мови програмування Python. Даний тест пройдено успішно.

3.3.4 Тестування роботи програми у випадках, коли на фотографіях відсутні обличчя

Для тестування було додано до відповідної папки зображення, на якому відсутнє людське обличчя (рис. 3.22). У результаті програма просто проігнорувала виклик функції ідентифікації, так як функція детекції обличчя нічого не повернула. Даний тест пройдено успішно.



Рисунок 3.22 – Тестове зображення для розробленої програми, на якому відсутнє людське обличчя

3.3.5 Тестування роботи програми у випадках, коли на фотографіях зображені тварини

Під час проведення тесту у відповідну папку були поміщені зображення 3 тварин – собаки, kota та мавпи (рис. 3.23–3.25). Ні на жодній фотографії програма не змогла виявити людське обличчя.

Як і в попередньому тестуванні, тут програма також проігнорувала виклик функції ідентифікації, так як функція детекції облич нічого не повернула. Дане тестування також пройдено успішно.



Рисунок 3.23 – Тестове зображення із собакою для розробленої програми



Рисунок 3.24 – Тестове зображення із котом для розробленої програми



Рисунок 3.25 – Тестове зображення із мавпою для розробленої програми

3.3.6 Тестування роботи програми при зміні параметрів для функцій детектування та розпізнавання облич

Хоча дані параметри на даному етапі розробки програмного застосунку недоступні користувачу, однак їх все одно слід протестувати, так як в майбутньому у випадку продовження розробки можна буде надати користувачеві доступ до додаткових параметрів.

В програмі є 2 функції, які приймають певні параметри – це функція детектування (рис. 3.26) облич та їх ідентифікації (рис. 3.27). Детальніше вони описані у попередньому підрозділі.

```
# Определяем области где есть лица
faces = faceCascade.detectMultiScale(image, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
```

Рисунок 3.26 – Скріншот коду, де функція `faceCascade.detectMultiScale` приймає параметри

```
# Для распознавания используем локальные бинарные шаблоны
recognizer = cv2.face.LBPHFaceRecognizer_create(2, 16, 4, 4, 23)
```

Рисунок 3.27 – Скріншот коду, де функція `LBPHFaceRecognizer_create` приймає параметри

У функції `faceCascade.detectMultiScale` перший параметр завжди задається програмно на рівні коду. Другий параметр – `scaleFactor`: при його зменшенні після кожної ітерації сканувальне вікно буде збільшуватися з меншою швидкістю, тим самим час виявлення збільшиться, а точність – також збільшиться. У випадку збільшення параметру вірогідність того, що обличчя на фото буде виявлено, стає меншою. Наступний параметр – `minNeighbors`. Від збільшення цього параметру функція стає менш впевненою у тому, що на даній ітерації положення сканувального вікна присутнє саме людське обличчя. Останній параметр – `minSize`. Від нього залежить розмір сканувального вікна з першої ітерації. Збільшення параметру призводить до більшої продуктивності, але до меншої точності виявлення.

Функція `LBPFaceRecognizer_create` приймає 5 параметрів. Перші 2 – значення кількості пікселів, які будуть оброблятися для ідентифікації особистості. При збільшенні значень точність розпізнавання значно зростає, але падає продуктивність системи. Наступні 2 – задання областей, на які програма розбиває обличчя. При зменшенні параметру точність зростає, час на обробку зростає. І останній параметр – поріг впевненості методу. Чим вище даний параметр, тим менше метод впевнений, що обличчя розпізнане.

Дане тестування пройдене успішно, хоча на даному етапі не є сильно актуальним.

3.4 Перспективи подальшої роботи

Розроблена програма на даному етапі має такі функції, як навчання на наборі зображень-еталонів, на основі яких є можливість знаходити схожі обличчя. Можна додавати зображення-еталони самотійно, також можна додавати фотографії людей, які потім будуть порівнюватися із еталонними. Результати роботи програми виводяться на екран та консоль.

Програмне забезпечення було успішно протестовано, що дає змогу зробити відповідне заключення щодо стабільності продукту при його використанні.

Але, незважаючи на всі указані переваги програмного забезпечення, присутні і недоліки.

Даний напрям є дуже перспективним. Тому даний проект потрібно розвивати в майбутньому. Перші плани для змін – побудувати кращий інтерфейс та розширити функціонал.

ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований застосунок для виявлення та класифікації подібності рис людських облич.

Було виконано все із постановки задачі, а саме:

– досліджено методи виявлення та класифікації подібності ознак на зображенні, даний аналіз показав, що на сьогоднішній день виділять 4 групи методів розпізнавання об'єктів, для прикладу було приведено по одному методу для кожної групи, цей аналіз дав можливість виявити переваги та недоліки кожного з них;

– досліджено методи виявлення та класифікації подібності рис людських облич, даний аналіз показав, що у найбільш загальному випадку алгоритм вирішення проблеми розпізнавання обличчя людини на зображенні складається з 4 етапів, для прикладу було приведено 4 методи розпізнавання облич, цей аналіз дав можливість виявити переваги та недоліки кожного з них;

– досліджено проблему вибору та класифікації інформативних ознак на зображенні, даний аналіз показав, що при розробці подібних продуктів існує ряд проблем, які необхідно враховувати;

– розроблено методику виявлення та класифікації подібності рис людських облич, дана розробка дала змогу виявити переваги та недоліки обраного методу;

– визначено з інструментальними засобами для створення застосунку;

– виконані всі етапи розроблення застосунку для виявлення та класифікації подібності рис людських облич;

– протестовано розроблений застосунок та зроблено аналіз результатів, після цього етапу було визначено слабкі та сильні сторони розробленої програми;

– визначено перспективи подальшої роботи, на даному етапі актуальність роботи підтвердилася, було визначено переваги та недоліки розробленого застосунку.

Мету роботи досягнуто.

Результат роботи апробовано у вигляді 2 тез доповідей під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ» та XXVI Міжнародної науково-практичної конференції «Topical issues of practice and science» [58, 59].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Star by Face: похожие знаменитости. URL: <https://play.google.com/store/apps/details?id=com.starbyface&hl=ru&gl=US> (дата звернення: 21.04.2021).
2. Celebs Like Me. URL: <https://play.google.com/store/apps/details?id=com.taoufiqlofti.celebslikeme&hl=ru&gl=US> (дата звернення: 21.04.2021).
3. Приложение Градиент (Who Do I Look Like). URL: <https://guide-apple.ru/wp-content/uploads/2019/09/Gradient-Photo-Editor.jpg> (дата звернення: 21.04.2021).
4. Как по фото найти двойника в интернете. URL: <https://sdelaicomp.ru/wp-content/uploads/2017/10/7-6.jpg> (дата звернення: 21.04.2021).
5. Celebrity Look-alike. URL: <https://image.winudf.com/v2/image1/YWkubWFyYy5sb29rYWxpa2Vfc2NyZWVuXzJfMTU1NDA3NzM4OV8wNzQ/screen-2.jpg?fakeurl=1&type=.jpg> (дата звернення: 21.04.2021).
6. Celebrity Look-alike. URL: <https://image.winudf.com/v2/image1/YWkubWFyYy5sb29rYWxpa2Vfc2NyZWVuXzJfMTU1NDA3NzM4OF8wODE/screen-1.jpg?fakeurl=1&type=.jpg> (дата звернення: 21.04.2021).
7. Looky - Celebrity Lookalike. URL: https://is4-ssl.mzstatic.com/image/thumb/PurpleSource114/v4/80/76/58/8076588d-eff4-eb26-e28b-2daf7175f41b/f74d9f89-6dfd-4c77-a750-8be528ef1efd_1-iPhone_8_-_Screen_3.png/392x696bb.png (дата звернення: 21.04.2021).
8. FindFace: приложение для поиска людей по фотографии. URL: <https://i.pinimg.com/originals/e3/ae/07/e3ae07020561f273dda1fe0f0ca91c41.jpg> (дата звернення: 21.04.2021).

9. PolitScanner. URL: https://static.tildacdn.com/tild6335-6133-4232-a332-376633353239/fb_2.png (дата звернення: 21.04.2021).
10. Cootes, T. F., & Taylor, C. J. (2004). Statistical models of appearance for computer vision.
11. Dong, J., Xia, W., Chen, Q., Feng, J., Huang, Z., & Yan, S. (2013). Subcategory-aware object classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 827-834).
12. Li, X., & Guo, Y. (2013). Adaptive active learning for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 859-866).
13. Deselaers, T., & Ferrari, V. (2010, June). Global and efficient self-similarity for object classification and detection. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 1633-1640). IEEE.
14. Zhang, B. (2010, July). Computer vision vs. human vision. In 9th IEEE International Conference on Cognitive Informatics (ICCI'10) (pp. 3-3). IEEE.
15. Burel, G., & Carel, D. (1994). Detection and localization of faces on digital images. *Pattern Recognition Letters*, 15(10), 963-967.
16. Heisele, B., Serre, T., & Poggio, T. (2007). A component-based framework for face detection and identification. *International Journal of Computer Vision*, 74(2), 167-181.
17. Samarzija, B., & Ribaric, S. (2014, May). An approach to the de-identification of faces in different poses. In 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 1246-1251). IEEE.
18. Berbar, M. A., Kelash, H. M., & Kandeel, A. A. (2006, July). Faces and facial features detection in color images. In Geometric Modeling and Imaging--New Trends (GMAI'06) (pp. 209-214). IEEE.
19. Yang, M. H., Kriegman, D. J., & Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 24(1), 34-58.

20. Кобилін О.А., Творошенко І.С. Методи цифрової обробки зображень: навч. посібник. Харків: ХНУРЕ, 2021. 124 с.

21. Daradkeh, Y.I., Tvoroshenko, I., Gorokhovatskyi, V., Latiff, L.A., and Ahmad, N. (2021) Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic, *IEEE Access*, 9, pp. 13417-13428.

22. Gorokhovatskyi, V., Rusakova, N., and Tvoroshenko, I. (2020) The application of image analysis methods and predicate logic in applied problems of magnetic monitoring, *Telecommunications and Radio Engineering*, 79(20), pp. 1801-1811.

23. Творошенко І.С. (2018) Особливості застосування сучасних принципів штучного інтелекту до розробки ефективних механізмів моделювання складних систем. *Science and Technology of the Present Time: Priority Development Directions of Ukraine and Poland: International Multidisciplinary Conference* (Wolomin, Republic of Poland, 19–20 October 2018). Wolomin: Izdevnieciba «Baltija Publishing». Volume 4. pp. 118-121.

24. Tvoroshenko I., and Dziubenko M. (2020) Modern methods of analysis of the movement scheme using video detection of vehicles, *Abstracts of V International Scientific and Practical Conference «Study of modern problems of civilization»* (October 19-23, 2020). Oslo, Norway, pp. 422-428.

25. Tvoroshenko I., and Tkachenko D. (2020) Mechanisms of image classification based on descriptors of local features, *Abstracts of IV International Scientific and Practical Conference «Integration of scientific bases into practice»* (October 12-16, 2020). Stockholm, Sweden, pp. 443-448.

26. Tvoroshenko I., and Zarivchatskyi R. (2020) Analysis of existing methods for searching object in the video stream, *Abstracts of VI International Scientific and Practical Conference «About the problems of science and practice, tasks and ways to solve them»* (October 26-30, 2020). Milan, Italy, pp. 500-505.

27. Kobylin O., Gorokhovatskyi V., Tvoroshenko I., and Peredrii O. (2020) The application of non-parametric statistics methods in image classifiers based on structural description components, *Telecommunications and Radio Engineering*, 79(10), pp. 855-863.

28. Gorokhovatskyi V.O., Tvoroshenko I.S., and Vlasenko N.V. (2020) Using fuzzy clustering in structural methods of image classification, *Telecommunications and Radio Engineering*, 79(9), pp. 781-791.

29. Gorokhovatskyi V., and Tvoroshenko I. (2020) Image Classification Based on the Kohonen Network and the Data Space Modification, *In CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2020)*, 2608, pp. 1013-1026.

30. Gorokhovatskyi V.O., Tvoroshenko I.S., and Peredrii O.O. (2020) Image classification method modification based on model of logic processing of bit description weights vector, *Telecommunications and Radio Engineering*, 79(1), pp. 59-69.

31. Ahmad M. Ayaz, Tvoroshenko Irina, Baker Jalal Hasan, and Lyashenko Vyacheslav (2019) Computational Complexity of the Accessory Function Setting Mechanism in Fuzzy Intellectual Systems, *International Journal of Advanced Trends in Computer Science and Engineering*, 8(5), pp. 2370-2377.

32. Yousef Ibrahim Daradkeh, and Iryna Tvoroshenko (2020) Application of an Improved Formal Model of the Hybrid Development of Ontologies in Complex Information Systems, *Applied Sciences*, 10(19). p. 6777.

33. Tvoroshenko I.S., and Gorokhovatsky V.O. (2020) Effective tuning of membership function parameters in fuzzy systems based on multi-valued interval logic, *Telecommunications and Radio Engineering*, 79(2), pp. 149-163.

34. Daradkeh Y.I., and Tvoroshenko I. (2020) Technologies for Making Reliable Decisions on a Variety of Effective Factors using Fuzzy Logic, *International Journal of Advanced Computer Science and Applications*, 11(5), pp. 43-50.

35. Lyashenko V., Mustafa S.K., Tvoroshenko I., and Ahmad M.A. (2020) Methods of Using Fuzzy Interval Logic During Processing of Space States of Complex Biophysical Objects, *International Journal of Emerging Trends in Engineering Research*, 8(2), pp. 372-377.

36. M. Ayaz Ahmad, Irina Tvoroshenko, Jalal Hasan Baker, and Vyacheslav Lyashenko (2019) Modeling the Structure of Intellectual Means of Decision-Making Using a System-Oriented NFO Approach, *International Journal of Emerging Trends in Engineering Research*, 7(11), pp. 460-465.

37. Tvoroshenko Irina, Ahmad M. Ayaz, Mustafa Syed Khalid, Lyashenko Vyacheslav, and Alharbi Adel R. (2020) Modification of Models Intensive Development Ontologies by Fuzzy Logic, *International Journal of Emerging Trends in Engineering Research*, 8(3), pp. 939-944.

38. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Intelligent classification of biophysical system states using fuzzy interval logic, *Telecommunications and Radio Engineering*, 78(14), pp. 1303-1315.

39. Творошенко І.С., Теслева К.О. Про розроблення програмного застосування для задачі логістики засобами геоінформаційних технологій. *Сучасний рух науки: тези доповідей VII міжнародної науково-практичної інтернет-конференції (Дніпро, 6–7 червня 2019 р.)*. Дніпро, 2019. С. 1659-1664.

40. Творошенко І.С., Зеленський М.О. Дослідження гібридних методів для класифікації складноструктурованих зображень. *Сучасний рух науки: тези доповідей VIII міжнародної науково-практичної інтернет-конференції (Дніпро, 3–4 жовтня 2019 р.)*. Дніпро, 2019. Т. 3. С. 382-387.

41. Лисенко, Г. Л., Тарновський, М. Г., & Кузьменко, Л. В. (2017). Сучасні тенденції у вирішенні задач виявлення та розпізнавання об'єктів на зображеннях. *Оптико-електронні інформаційно-енергетичні технології*, 33(1), 18-23.

42. Нургатин, А. Р. (2014). Метод улучшения алгоритма Виолы-Джонса. *Математические структуры и моделирование*, (4 (32)).

43. Vizil'ter, Y. V., Zheltov, S. Y., Knyaz, V. A., Khodarev, A. N., & Morzhin, A. V. (2007). Obrabotka i analiz tsifrovyykh izobrazhenii s primerami na LabVIEW IMAQ Vision (Processing and Analysis of Digital Images with Examples on LabVIEW IMAQ Vision).

44. Савченко, А. В. (2011). Теоретико-вероятностная модель полутонового изображения для задачи распознавания образов без учителя на основе метода направленного перебора. Компьютерная оптика, 35(3).

45. Мокеев, В. В., & Томилов, С. В. (2014). О решении задачи распознавания изображений методом главных компонент и линейным дискриминантным анализом. Компьютерная оптика, 38(4).

46. Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 (Vol. 1, pp. I-I). IEEE.

47. Viola, P., & Jones, M. J. (2004). Robust real-time face detection. International journal of computer vision, 57(2), 137-154.

48. Выделение объектов на изображении по методу Виолы-Джонса (Object selection in the image according to the Viola-Jones method). URL: <https://api-2d3d-cad.com/viola-jones-method/> (дата звернения: 29.04.2021).

49. Perveen, N., Kumar, D., & Bhardwaj, I. (2013). An overview on template matching methodologies and its applications. International Journal of Research in Computer and Communication Technology, 2(10), 988-995.

50. Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1), 71-86.

51. Выделение и распознавание лиц. URL: http://wiki.technicalvision.ru/index.php/%D0%92%D1%8B%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B8_%D1%80%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BB%D0%B8%D1%86 (дата звернения – 30.04.2021).

52. Al-Allaf, O. N. (2014). Review of face detection systems based artificial neural networks algorithms. arXiv preprint arXiv:1404.1292.

53. Анализ существующих подходов к распознаванию лиц. URL: <https://habr.com/ru/company/synesis/blog/238129/> (дата звернення 30.04.2021).

54. Ту, Д., & Гонсалес, Р. (1978). Принципы распознавания образов: Пер. с англ. Мир.

55. Фукунага, К. (1979). Введение в статистическую теорию распознавания образов: Пер. с англ. Наука.

56. Анисимов, Б. В., Курганов, В. Д., & Злобин, В. К. (1983). Распознавание и цифровая обработка изображений: Учеб. пособие для вузов. Высш. шк..

57. Ковалевский, В. А. (1976). Методы оптимальных решений в распознавании изображений. Наука. Гл. ред. физ.-мат. лит..

58. Tvoroshenko I., and Koriakin I. (2021) Analysis of methods for detecting and classifying the likeness of human features, Abstracts of XXVI International Scientific and Practical Conference «Topical issues of practice and science» (May 18-21, 021). London, Great Britain, pp. 674-679.

59. Корякін І.М. Про особливості деяких методів розпізнавання облич людей на зображенні. Радіоелектроніка та молодь у ХХІ столітті: тези доповідей 25-го Міжнародного молодіжного форуму (Харків, 20–21 квітня 2021 р.). Харків: ХНУРЕ, 2021. Т. 7, 10. С. 30-31.