

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СТИРЕНКО

(підпис)

“ ” _____ 2021 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою “Комп’ютерні системи та мережі”

спеціальності 123 “Комп’ютерна інженерія”

на тему: _____ «Система розпізнавання облич»

Виконала: студентка 4 курсу, групи ІО-71
(шифр групи)

Приймич Анна-Вікторія Олегівна

(прізвище, ім’я, по батькові)

(підпис)

Керівник ас. Каплунов А. В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль) д.т.н. проф. Сімоненко В. П.

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що в цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ – 2021 р.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальність 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

Сергій СТИРЕНКО

(підпис)

“ ” _____ 2021 р.

ЗАВДАННЯ

на бакалаврський дипломний проєкт студента

Приймич Анни-Вікторії Олегівни

1. Тема проєкту Система розпізнавання облич

керівник проєкту Каплунов Артем Володимирович, асистент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 11 травня 2021 року № 1139-с

2. Термін здачі студентом закінченого проєкту 4 червня 2021 року

3. Вихідні дані по проєкту технічна документація, теоретичні та статистичні дані.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються) Опис та аналіз предметної області, аналіз інструментів для розробки програмного забезпечення, розробка програмного забезпечення.

5. Перелік графічного матеріалу (з точним позначенням обов’язкових креслень) структурна схема системи, узагальнена схема роботи системи, схема взаємодії програми.

6. Консультанти проєкту, з вказівкою розділів проєкту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Сімоненко В. П.		

7. Дата видачі завдання 28 вересня 2020 року

Календарний план

№ з/п	Найменування етапів дипломного проєкту	Терміни виконання етапів проєкту	Примітки
1.	<i>Затвердження теми проєкту</i>	15.12.2020-14.01.2021	
2.	<i>Вивчення та аналіз завдання</i>	15.01.2021-14.03.2021	
3.	<i>Розробка загальної структури системи та її архітектури</i>	15.03.2021-24.03.2021	
4.	<i>Розробка структури компонентів системи</i>	25.03.2021-04.04.2021	
5.	<i>Програмна реалізація системи</i>	05.04.2021-19.04.2021	
6.	<i>Оформлення пояснювальної записки</i>	20.04.2021-19.05.2021	
7.	<i>Захист програмного продукту</i>		
8.	<i>Передзахист</i>		
9.	<i>Захист</i>	18.06.2021	

Студент-дипломник _____
(підпис)

Керівник проєкту _____
(підпис)

Анотація

В даному дипломному проєкті було розроблено систему для розпізнавання облич на базі бібліотеки комп'ютерного зору, призначену для розпізнавання облич.

Програма дозволяє виконувати розпізнавання облич на відео у режимі реального часу.

Для реалізації програми використовувалась бібліотека комп'ютерного зору OpenCV з відкритим кодом, мова програмування Python та інтегроване середовище розробки PyCharm.

Annotation

In this diploma project a facial recognition system based on a computer vision library was developed. The program is designed for face recognition.

The program performs face recognition on video in real time.

OpenCV open source computer vision library, the Python programming language, and the PyCharm integrated development environment were used in the process of development.

Опис альбому
до дипломного проєкту
на тему: «Система розпізнавання обличч»

Київ – 2021 року

[illegible]

Технічне завдання
до дипломного проєкту
на тему: «Система розпізнавання облич»

Київ – 2021 року

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	2
2. ПРИЧИНИ ДЛЯ РОЗРОБКИ.....	2
3. ЦІЛЬ ТА ПРИЗНАЧЕННЯ РОЗРОБКИ	2
4. ДЖЕРЕЛА РОЗРОБКИ	2
5. ТЕХНІЧНІ ВИМОГИ	2
5.1 ВИМОГИ ДО ПРОДУКТУ, ЩО РОЗРОБЛЯЄТЬСЯ	2
5.2 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	2
5.3 ВИМОГИ ДО АПАРАТНОЇ ЧАСТИНИ.....	3
6. ЕТАПИ РОЗРОБКИ.....	3

					ІАЛЦ.467800.002 ТЗ						
Зм.	Арк.	№ докум.	Підпис	Дата							
Розробив		Приймич А.-В. О.			Система розпізнавання облич Технічне завдання						
Перевірів		Каплунов А. В.									
Реценз.											
Н. Контр.		Сімоненко В.П.									
Затв.		Стіренко С.Г.									
					Літ.	Арк.	Аркушів				
						1	3				
					«КПІ імені Ігоря.Сікорського» ФІОТ, гр. ІО-71						

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Це технічне завдання поширюється на розробку системи розпізнавання облич. Область застосування: альтернатива існуючим системам розпізнавання облич.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання бакалаврського проєкту професійної програми “Комп'ютерні системи та мережі” спеціальності 123 “Комп'ютерна інженерія”, затверджене кафедрою Обчислювальної техніки Національного технічного Університету України “Київський Політехнічний інститут імені Ігоря Сікорського”

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою цього проєкту є розробка системи виявлення і розпізнавання облич.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами розробки є науково-технічна література, публікації в спеціалізованих періодичних виданнях, технічна документація, публікації в мережі Інтернет на цю тему.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленого продукту

- Система повинна збирати і опрацьовувати зображення з веб-камери.
- Забезпечення точності виявлення і розпізнавання облич.
- Простота у користуванні.

					ІАЛЦ.467800.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		2

5.2. Вимоги до програмного забезпечення

- Операційна система MS Windows 7, MS Windows 8/8.1, MS Windows 10.

5.3. Вимоги до апаратної частини

- Коп'ютер на базі Intel Pentium 4 і вище.
- Обсяг оперативної пам'яті не менше 1024 Мбайт.
- Наявність веб-камери.

6. ЕТАПИ РОЗРОБКИ

	Дата
6.1 Вивчення необхідної літератури	21.02.2021
6.2 Складання і узгодження технічного завдання	08.03.2021
6.3 Написання вступної частини та огляд рішень	22.03.2021
6.4 Розробка архітектури додатку	04.04.2021
6.5 Написання програмної частини	11.04.2021
6.6 Тестування та виправлення помилок	04.05.2021
6.7 Оформлення документації дипломного проєкту	15.05.2021

					ІАЛЦ.467800.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		3

Пояснювальна записка
до дипломного проєкту
на тему: «Система розпізнавання облич»

Київ – 2021 року

ЗМІСТ

ВСТУП	2
РОЗДІЛ 1. ОГЛЯД ВЖЕ ІСНУЮЧИХ ТЕХНОЛОГІЙ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ	3
1.1 Етапи розвитку технологій розпізнавання облич.....	3
1.2 Огляд програмного забезпечення для розпізнавання облич.....	9
ВИСНОВКИ ДО РОЗДІЛУ 1	14
РОЗДІЛ 2. АЛГОРИТМИ РОЗПІЗНАВАННЯ ОБЛИЧ	15
2.1 Алгоритм Eigenfaces	15
2.2 Алгоритм Fisherfaces	20
2.3 Алгоритм LBPН	21
2.4 Метод Віюли-Джонса.....	24
2.5 Застосування глибокого навчання для розпізнавання облич.....	28
2.6 Порівняння розглянутих алгоритмів розпізнавання облич.....	35
ВИСНОВКИ ДО РОЗДІЛУ 2	38
РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ	39
3.1 Опис засобів реалізації.....	39
3.2 Опис проекту.....	40
3.3 Тестування системи.....	43
ВИСНОВКИ ДО РОЗДІЛУ 3	51
ВИСНОВКИ.....	52
ВИСНОВКИ СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	53

					ІАЛЦ.467800.003 ПЗ					
Зм.	Арк.	№ докум.	Підпис	Дата	Система розпізнавання облич			Літ.	Арк.	Аркуші
Розробив		Приймач А.-В.О.								
Перевірів		Каплунов А. В.							1	55
Реценз.								«КПІ імені Ігоря Сікорського» ФІОТ, гр. ІО-71		
Н. Контр.		Сімоненко В.П.								
Затв.		Стіренко С.Г.								

ВСТУП

Розпізнавання облич є досить тривіальним завданням для людини. Зазвичай вона може впізнати родичів та знайомих навіть якщо вони постаріли, почали носити окуляри чи запустили бороду. Для комп'ютера ця задача не є легкою, незважаючи на те, що винайдено багато нових технологій і методів для виявлення облич на зображеннях.

Компанії та організації проявляють великий інтерес у вдосконаленні систем розпізнавання облич, оскільки вони мають широке коло застосувань, в тому числі комерційне. Пошук злочинців, системи відеонагляду, облік відвідувань, розблокування телефону, визначення облич на фото в соцмережах – це далеко не повний перелік застосувань. Деякі системи розпізнавання знаходять обличчя з дуже великою точністю – близько 98%, цей показник є кращим ніж впізнавання людей людиною. Проте розробники систем розпізнавання не зупиняються на цьому, оскільки навіть невелика можливість похибки для систем, які використовуються, наприклад, у криміналістиці може призвести до негативних наслідків.

Останнє десятиліття відзначилось значним прогресом у сфері комп'ютерного зору і систем розпізнавання образів. Застосування глибокого навчання, особливо згорткових нейронних мереж, значно підвищило точність і ефективність розпізнавання. Проте цей підхід теж має недоліки, тренування моделей глибокого навчання потребує великих обчислювальних потужностей і, відповідно, дорогого обладнання. Для підвищення точності вихідних результатів потрібно обробити велику кількість вхідних даних.

В даній роботі буде проаналізовано алгоритми, які використовуються для розробки аналогічних систем, вибрано найкращі підходи і на їх основі реалізовано власну систему.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		2

РОЗДІЛ 1

ОГЛЯД ВЖЕ ІСНУЮЧИХ ТЕХНОЛОГІЙ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ

1.1 Етапи розвитку технологій розпізнавання облич

Піонерами у сфері розпізнавання облич були американські науковці Вудро Бледсоу, Хелен Чан і Чарльз Бісон, які на початку 1960-их років працювали над проектом з використання комп'ютера для розпізнавання облич [1]. Тільки невелика частина досліджень і результатів роботи була опублікована, оскільки проект фінансувався Центральним розвідувальним управлінням і був засекреченим.

Розроблена система була названа людино-машинною, тому що спочатку людина-оператор визначала координати сукупності рис обличчя на фотографіях, а вже після цього комп'ютер використовував ці координати для виконання розпізнавання. Оператор за допомогою графічного планшету визначав координати рис обличчя: центри зіниць, внутрішній куточок очей, зовнішній куточок очей, і т. д. Виходячи з цих координат обчислювались двадцять відстаней: ширина рота, ширина очей, відстань від центра однієї зіниці до іншої. Вручну одна людина могла обробити 40 фотографій за годину. Після обчислення необхідних відстаней на комп'ютері складалась база даних, у якій імені людини на фото відповідав набір значень обрахованих відстаней. На етапі розпізнавання набір відстаней порівнювався з наборами відстаней в базі даних, комп'ютер обчислював різницю і повертав найменше значення як можливий збіг.

З вищеописаного можна зробити висновок, що перша система була доволі примітивною, але все ж дозволяла частково автоматизувати процес розпізнавання облич і вплинула на створення систем-послідовниць.

У 1970-і роки Голдштейн, Хармон і Леск удосконалили систему, розроблену Бледсоу і його колегами. Були додані маркери для рис обличчя.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		3

Точність розпізнавання обличчя збільшилась у порівнянні із попередньою методикою, але визначення координат і відстаней все ще робилось вручну.

Наприкінці 1980-их Сіровіч та Кірбі застосували лінійну алгебру для вирішення проблеми розпізнавання облич і розробили систему під назвою Eigenfaces [2]. У цій системі для точного кодування нормалізованого зображення обличчя використовувалось менше ста значень. В 1991 році Турк і Пентланд продовжили роботу над Eigenfaces. Вони винайшли спосіб розпізнавання лиця на зображеннях, який був повністю автоматичним. В 1997 році Eigenfaces був вдосконалений використанням лінійного дискримінантного аналізу (LDA) і отримав назву Fisherfaces.

Виявлення облич в режимі реального часу на відео стало можливим у 2001 році завдяки методу Віоли-Джонса. Пол Віола і Майкл Джонс, поєднали свій метод для виявлення облич з ознаками Хаара, що дозволило значно пришвидшити обчислення.

Національний інститут стандартів і технологій (США) у 1993 році розпочав програму FERET. Ця програма мала на меті створити стандартизовану базу даних для дослідження технологій розпізнавання обличчя. До запровадження FERET більшість дослідників самостійно збирали зображення для бази даних облич. Зазвичай такі індивідуальні бази даних були невеликі й містили зображення менш ніж 50 осіб. Відсутність спільної бази даних ускладнювала порівняння результатів досліджень різних методів розпізнавання облич в науковій літературі. Більшість опублікованих робіт не використовували зображення із загальної бази даних і не слідували стандартному протоколу тестування. Як результат, дослідники не могли провести обґрунтоване порівняння результатів роботи різних алгоритмів розпізнавання облич. За час проведення програми було зібрано базу даних із 14126 зображень облич 1199 людей [3]. Учасники програми з різних університетів і дослідницьких організацій отримали можливість протестувати свої системи для розпізнавання облич на зібраній базі зображень, виявити їхні недоліки і порівняти із системами-конкурентами.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		4

На початку 2000-их Національний інститут стандартів і технологій розгорнув FVRT, програму-послідовницю FERET. Метою цієї програми було надання незалежної оцінки системам розпізнавання облич, які були доступні на ринку, а також прототипам таких систем.

Спонсоровані американським урядом дослідження та змагання допомогли суттєво покращити роботу систем розпізнавання облич. Починаючи з 1993 до 2010 року, коефіцієнт помилок автоматичних систем розпізнавання облич зменшився у 272 рази [4].

Починаючи з 2010 року у сфері розпізнавання образів та класифікації зображень починає домінувати глибоке навчання (deep learning, DL). На той час проводилось мало досліджень із застосуванням нейронних мереж. На думку науковців вони вважались безперспективними. Проте група Джефрі Хінтона із університету Торонто, дослідники із Науково-дослідного інституту штучного інтелекту IDSIA у Швейцарії та декілька інших науковців продовжували роботу з нейронними мережами і їм вдалось добитись помітних успіхів. У 2011 році Ден Кірсан з IDSIA переміг в академічному конкурсі по класифікації зображень із застосуванням глибоких нейронних мереж, навчання яких відбувалось з використанням GPU. В 2012 році відбулась ще одна важлива подія, яка підтвердила можливість успішного практичного застосування глибокого навчання. Група Джефрі Хінтона взяла участь у щорічних змаганнях ImageNet по розпізнаванню зображень. Завдання конкурсу було дуже складним на той час, і полягало в тому, щоб класифікувати кольорові зображення з високою роздільною здатністю на 1000 різних категорій після навчання на 1,4 мільйона зображень. В 2011 році точність моделі-переможця, реалізований з використанням класичних підходів, була тільки 74,3%. У 2012 році команда Алекса Крижевські досягла точності 83,6%. Це був значний стрибок. З того часу щорічно перші місця на цих змаганнях займають згорткові нейронні мережі (CNN). В 2015 році переможець досягнув точності 96,4% [5].

Починаючи із 2012 року згорткові нейронні мережі стали передовим алгоритмом для всіх задач розпізнавання образів. Пізніше сфера застосування

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		5

CNN поширилась на задачі розпізнавання взагалі, наприклад, розпізнавання слів людської мови і т. п.

Концепція спрощених нейронних мереж була сформована ще в 50-их роках XX століття. Проте досить довго розвиток цього підходу гальмувався відсутністю методів для навчання великих нейронних мереж, низькою продуктивністю процесорів і труднощами у зборі даних для навчання.

В 2000-их роках компанії NVIDIA і AMD вклали мільйони доларів в розробку графічних процесорів (GPU) для підтримки графіки відеоігор, яка ставала все більш реалістичною. Це мали бути недорогі спеціалізовані процесори, які мали відповідати за відображення на екрані складних тривимірних сцен в режимі реального часу. Інвестиції у розвиток GPU пішли на користь не тільки ігровому співтовариству, а й науковому, коли у 2007 році компанія NVIDIA випустила CUDA – програмний інтерфейс для лінійки своїх GPU. Декілька таких GPU можуть замінити кластери зі звичайними процесорами для виконання різноманітних задач, які дозволяють масове розпаралелювання обчислень. Глибокі нейронні мережі, які в основному виконують множення великої кількості матриць, відносяться до задач, які легко розпаралелюються.

Графічні процесори, здавалось, ідеально підходять для виконання задач глибокого навчання, проте індустрія DL починає виходити за рамки GPU. Компанії вкладають гроші у розвиток ще більш спеціалізованих процесорів, які покажуть більшу ефективність у сфері глибокого навчання. У 2016 році на щорічній конференції Google Innovation in the Open компанія Google презентувала свій проект тензорного процесора (Tensor Processing Unit, TPU): процесор з новою архітектурою, призначений для використання в глибоких нейронних мережах.

Револьюційні зміни відбувались не тільки з процесорами. По-перше, за останні 20 років спостерігається експоненціальне зростання обсягу пристроїв зберігання інформації. По-друге, бурхливо розвинувся інтернет, збільшилась кількість користувачів і кількість інформації в мережі. Для прикладу, зараз

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		6

нараховується близько 1 мільярда користувачів соціальної мережі Instagram, які завантажують 8,5 мільйонів фотографій на день, а загальна кількість фото в Instagram налічує 50 мільярдів. YouTube має 2,3 мільярди користувачів, і близько 720 000 годин відео завантажуються на цей відеохостинг щодня. Це просто грандіозні цифри, які й уявити не можна було 20 років тому.

Ці два фактори сприяли накопиченню і поширенню дуже великої кількості даних для машинного навчання. Сьогодні великі компанії працюють з колекціями зображень, відео і текстових матеріалів, які неможливо було б зібрати без інтернету. Для прикладу, зображення на сайті Flickr, класифіковані користувачами, стали справжнім клондайком для розробників моделей розпізнавання образів. Те саме можна сказати про відеоролики із відеохостингу YouTube. Каталізатором для розвитку глибокого навчання можна вважати колекцію ImageNet, яка містить 1,4 мільйона зображень, класифікованих вручну на 1000 категорій, кожне зображення відноситься тільки до однієї категорії.

Потрібно відзначити, що протягом останніх двох десятиліть покращились технічні характеристики цифрових камер. Також збільшилась їх доступність і кількість. Жоден смартфон зараз не обходиться без камери а то і декількох. Виробники телефонів і цифрової техніки постійно вдосконалюють характеристики своїх камер. Покращення стосуються не тільки лінз і світлочутливої матриці, але й програмного забезпечення, яке обробляє сигнали, що надходять з матриці. Цей факт пояснює велику кількість фото і відео, яка щодня завантажуються в мережу інтернет і може стати потенційним джерелом для різноманітних датасетів. З якіснішими зображення алгоритмам легше працювати, оскільки лиця на них видно чіткіше і їх стає набагато легше виявити і розпізнати.

На додачу до процесорів і даних, до кінця 2000-их не існувало надійного способу навчання дуже глибоких нейронних мереж. Через це нейронні мережі залишались неглибокими. Вони могли мати один або два шари представлення, і не могли конкурувати із досконалішими поверхневими методами. Найбільша проблема полягала в поширенні градієнта крізь велику кількість шарів. Сигнал

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		7

зворотного зв'язку, який використовувався для навчання нейронних мереж, затухав із збільшенням кількості шарів. Ситуація змінилась з появою в 2009 – 2010 роках простих, але важливих алгоритмічних вдосконалень, які дозволили покращити поширення градієнта. Завдяки цим вдосконаленням стало можливим створення моделей з 10 і більше шарами. В 2014 – 2016 роках були винайдені ще досконаліші способи поширення градієнта, що дозволило зробити нейронні мережі ще глибшими. На сьогодні можна створювати і навчати моделі мереж, глибина яких сягає декілька тисяч шарів.

Ще одним важливим фактором стали інвестиції в галузь глибокого навчання, які були найбільшими за всю історію розвитку штучного інтелекту. Для порівняння, в 2011 році, перед тим як глибоке навчання зайняло лідируючу позицію, інвестиції в розвиток штучного інтелекту становили близько 19 мільйонів доларів. В 2014 році ця цифра вже дорівнювала 394 мільйонам доларів. Такі технологічні гіганти як Google, Facebook, Baidu і Microsoft вклали гроші в дослідження, які проводились внутрішніми підрозділами, і обсяги цих інвестицій майже напевно перевищують інвестиції венчурного капіталу. Для прикладу, в 2013 році компанія Google купила стартап DeepMind, який займався глибоким навчанням, за 500 мільйонів доларів і це одна з найдорожчих покупок в історії AI.

Останнім часом технології розпізнавання облич активно використовуються соціальними мережами. Facebook почав використовувати розпізнавання облич у 2010 році в США [6]. Фото користувачів сканувались на наявність облич. Якщо обличчя було знайдено, юзер отримував ім'я людини на зображенні. Застосування технології викликало багато суперечок. Для застосування функції розпізнавання не потрібен був дозвіл користувача. З 2019 року функція розпізнавання облич працюватиме, якщо юзер дасть на це дозвіл.

Системи розпізнавання облич все частіше зустрічаються в повсякденному житті. Так у 2017 році компанія Apple презентувала телефон із системою Face ID, яка дозволяє розблоковувати телефон обличчям. Після цього інші компанії-виробники телефонів почали випускати пристрої з аналогічним функціоналом.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		8

З одного боку, системи розпізнавання облич можуть використовуватись для ідентифікації злочинців, проходження контролю в аеропортах та ін. Проте, наприклад, в Китаї кількість камер з функцією розпізнавання облич досягає 626 мільйонів. Зйомка зазвичай відбувається без згоди громадян, а великі обсяги конфіденційних біометричних даних часто не захищаються належним чином. З одного боку це явне порушення прав людини, але з іншого боку через великий попит на системи розпізнавання облич китайські компанії постійно удосконалюють свої технології в цій сфері і саме Китай є найбільшим експортером систем розпізнавання облич [7].

1.2 Огляд програмного забезпечення для розпізнавання облич

Amazon Rekognition – це хмарний сервіс для розпізнавання облич запущений компанією Amazon у 2016 році. Rekognition використовує глибоке навчання для розпізнавання зображень. Цей сервіс застосовувала низка урядових установ США, включаючи Міграційну та митну службу США, поліцію Орlando, штат Флорида, а також приватні організації.

Можливості, які надає Rekognition можна поділити на дві категорії: алгоритми, навчені на даних, зібраних Amazon, і алгоритми, які користувач може тренувати на власному наборі даних [8].

В категорії навчених алгоритмів є такий набір можливостей:

- розпізнавання знаменитостей на зображення
- розпізнавання облич з визначенням статі, віку, емоцій, чи на лиці є борода або вуса, чи на обличчі є окуляри для зору або від сонця, чи людина на фото усміхається та ін.
- відстежування людей на відео. Наприклад, відстежування гравців на полі для післяматчевого аналізу
- виявлення тексту на зображеннях
- виявлення невідповідного, небажаного чи образливого вмісту.

В категорії алгоритмів, які можна вчити є наступні:

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		9

- SearchFaces – дозволяє завантажувати власну базу зображень для тренування моделі машинного навчання на ній. Після навчання користувач може завантажити зображення і отримати інформацію про обличчя, які є на зображенні.
- верифікація користувачів за обличчям.

Face ID – це система розпізнавання облич розроблена Apple для iPhone і iPad Pro. Face ID надає можливість біометричної автентифікації для розблокування пристрою, а також відстеження виразу обличчя. Вперше система з'явилась в iPhone X, випущеному в 2017 році. З того часу вона отримала оновлення в нових моделях iPhone і iPad Pro.

Апаратне забезпечення Face ID складається з датчика з трьома модулями: точковий проектор, який проектує сітку маленьких інфрачервоних точок на обличчя користувача, кількість яких перевищує 30000 штук; модуль, який випромінює інфрачервоне світло в обличчя; інфрачервона камера, яка знімає інфрачервоне зображення користувача, зчитує отриманий шаблон і створює 3D-карту обличчя. Ця карта порівнюється із зареєстрованим обличчям за допомогою захищеної підсистеми, і користувач аутентифікується, якщо обидва обличчя збігаються. Інфрачервона камера забезпечує роботу системи при поганому освітленні або в темноті.

Під час початкового налаштування обличчя користувача сканується двічі з різних кутів, щоб створити повну контрольну карту обличчя. Apple зазначила, що карта не буде відправлятися в клауд, а зберігатиметься у зашифрованому вигляді на локальному співпроцесорі Secure Enclave [9]. Задля безпеки очі користувача під час сканування мають бути відкритими. Після 5 невдалих спроб автентифікації, неактивності пристрою протягом 48 годин, перезавантаження FaceID тимчасово відключається і користувач зможе розблокувати гаджет тільки за допомогою паролю. Система може розпізнавати обличчя в окулярах, з макіяжем, бородою та адаптується до змін зовнішнього вигляду з часом. Цей функціонал обслуговує нейронний процесор Neural Engine розроблений Apple для пришвидшення виконання задач машинного навчання і глибокого навчання.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		10

Компанія випустила 4 покоління Neural Engines, щоразу покращуючи продуктивність та енергоефективність цих процесорів. FaceID дозволяє здійснювати оплату за допомогою Apple Pay, також ця система дозволяє з високою точністю захоплювати вираз обличчя користувача і використовує цю функцію для накладання різноманітних фільтрів і масок на обличчя в режимі реального часу.

Незважаючи на те, що FaceID дуже добре справляється із виявленням та розпізнаванням облич, вона не є ідеальною. Слабкою стороною системи є розпізнавання близнюків та родичів. Apple в “білій книзі” попереджає, що у таких випадках ймовірність помилкового збігу зростає, і рекомендує використовувати для аутентифікації пароль [10]. Варто зауважити, що це є проблемою для всіх систем розпізнавання обличчя, оскільки більшість із них базується на знаходженні відмінностей одного лиця від іншого.

DeepFace – це система розпізнавання облич, розроблена Facebook. Компанія випустила DeepFace на початку 2015 року і з того часу постійно розширює її застосування. Система використовує алгоритми глибокого навчання і складається із дев’ятишарової нейронної мережі яка має більш ніж 120 мільйонів вагових з’єднань. Навчання цієї нейронної мережі проводилось на 4 мільйонах зображень, завантажених користувачами соцмережі Facebook. Оскільки люди продовжують завантажувати фото, датасет DeepFace постійно поповнюється і зростає точність алгоритму. Дослідницька група Facebook заявила, що точність їхньої системи досягає 97,35 % на датасеті LFW (Labeled Faces in the Wild) [11]. При цьому точність розпізнавання лиць людиною становить близько 97 %. Це означає, що DeepFace повинна справлятися із задачею розпізнавання облич краще ніж людина.

DeepFace аналізує фото у Facebook, на якому зображений користувач, наприклад основну світліну на сторінці і створює унікальний числовий код для цього користувача, який ще називається шаблоном. Потім код використовується для розпізнавання обличчя юзера на інших світлинах. Система розпізнавання має функцію захисту користувачів від зловмисників, які неправомірно

використовують особисті дані або видають себе за іншу особу [12]. Також DeepFace може повідомляти людям із порушенням зору, хто зображений на фото чи відео.

Процедура розпізнавання обличчя для DeepFace включає вирівнювання обличчя. Після виявлення обличчя на зображенні з допомогою методу опорних векторів (SVM) знаходяться спеціальні координатні мітки (рис. 1.1). Метод опорних векторів визначає їх виходячи із дескриптора зображення, створеного за допомогою гістограм локальних бінарних шаблонів (LBPH) [11].

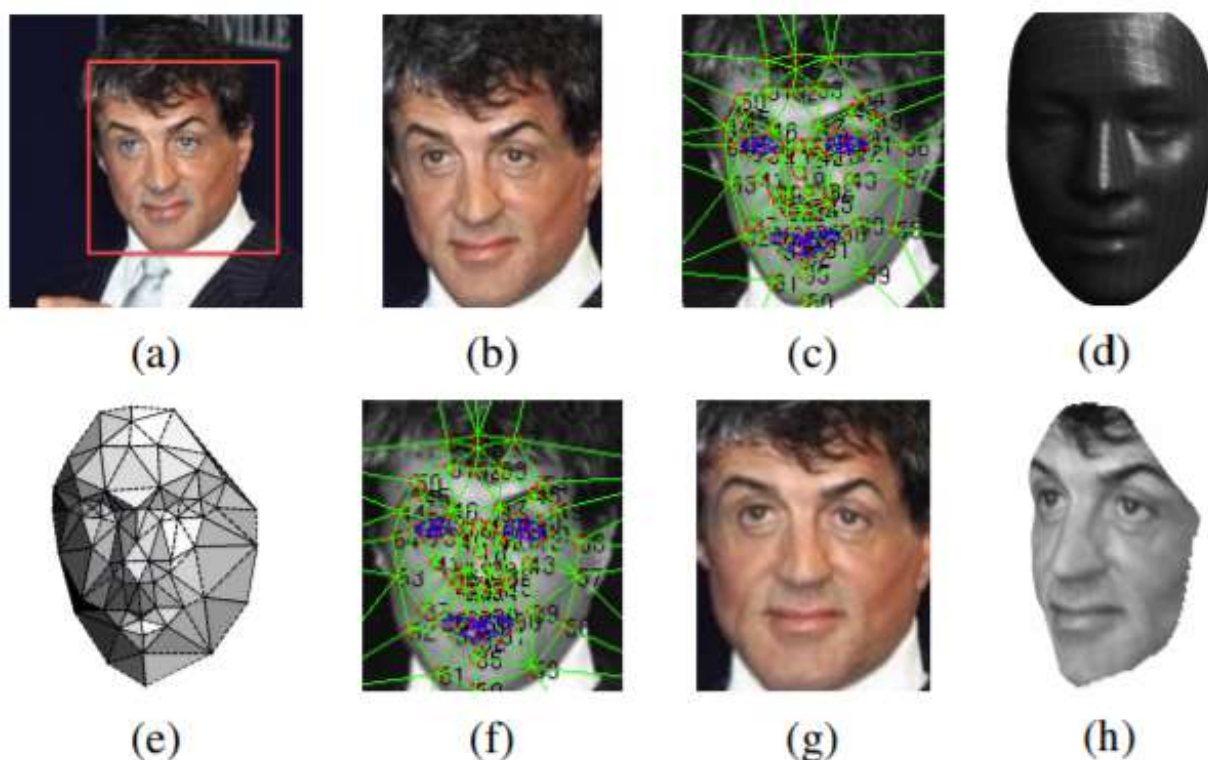


Рисунок 1.1 – Покроковий процес вирівнювання обличчя

Після знаходження координатних міток відбувається процес двовимірного вирівнювання. Далі слідує тривимірне вирівнювання, а саме накладання координатних міток на 3-D форму голови. Після проведення ряду математичних операцій одержуємо обрізане тривимірне обличчя як на рис. 1.1.

Наступним кроком є передача отриманого результату на вхід глибокої нейронної мережі (DNN). Нейронна мережа складається із двох шарів згортки: C1 і C3. Шар C1 містить 32 фільтри, з допомогою яких відповідно можна одержати 32 карти ознак. Шар C3 складається із 16 фільтрів. Між шарами згортки міститься шар пулінгу M2 з вибором максимального значення. Після

шару C3 йдуть три локально зв'язані шари (L4, L5, L6) і два повнозв'язні шари (F7, F8). На виході нейронна мережа видає id обличчя, тобто визначає кому воно належить [11]. Архітектуру DNN наведено на рис. 1.2.

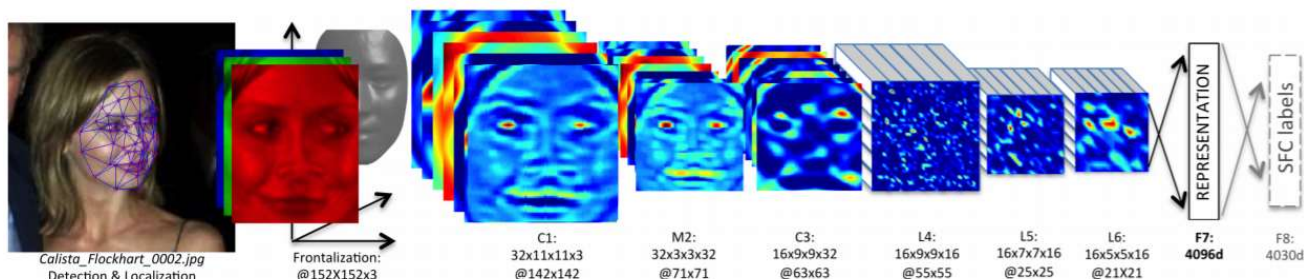


Рисунок 1.2 – Архітектура DNN, яку використовує DeepFace

ВИСНОВКИ ДО РОЗДІЛУ 1

У першому розділі було розглянуто етапи розвитку алгоритмів та методів, які застосовувались і дотепер застосовуються в системах розпізнавання облич. Спочатку були примітивні підходи, які намагались автоматизувати процес розпізнавання, наступні системи ставали все точнішими. Для розробки систем, точність яких перевищує 90 % знадобилось майже 60 років. За останні десять років стався справжній прорив – точність систем розпізнавання досягла рівня можливостей людини впізнавати інших людей завдяки застосуванню глибокого навчання. Цьому посприяло багато факторів, серед них створення потужних GPU і спеціалізованих процесорів, збільшення ємності накопичувачів, наявність великої кількості зображень та відео в інтернеті для створення датасетів, а також поява нових ефективних алгоритмів.

Приклади програмного забезпечення для розпізнавання облич, наведені у розділі, показують що компанії та розробники також роблять ставки на глибоке навчання, проте не обмежуються тільки ним. Наприклад, FaceID та DeepFace застосовують для розпізнавання 3D моделі обличчя. Для створення та обробки цих моделей використовуються засоби аналітичної геометрії і методи традиційного комп'ютерного зору. Тобто розробники і науковці комбінують різні підходи для отримання кращих результатів, і застосовують нові підходи для вирішення задачі розпізнавання обличчя.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		14

РОЗДІЛ 2

АЛГОРИТМИ РОЗПІЗНАВАННЯ ОБЛИЧ

2.1 Алгоритм Eigenfaces

В 1991 році Метью Турк і Алекс Пентланд запропонували новий підхід для вирішення проблеми розпізнавання обличчя під назвою Eigenfaces [13]. Для початку 90-их це був доволі ефективний і простий в застосуванні метод розпізнавання облич.

В основі алгоритму лежить використання фундаментальних статистичних характеристик, таких як математичне очікування і коваріаційна матриця, а також методу головних компонент (РСА). Останній був запропонований ще в 1901 році Карлом Пірсоном, але й сьогодні застосовується для зменшення розмірності даних.

Зображення в градаціях сірого розмірністю $p \times q$ можна представити у $p \cdot q$ -вимірному векторному просторі. Якщо взяти зображення 100×100 пікселів, то воно буде належати 10000-вимірному простору. Проведення операцій з подібними значеннями вимагає багато обчислювальних ресурсів. Зі збільшенням розмірів зображення кількість вимірів теж збільшиться, проте далеко не всі виміри містять корисну для подальшого аналізу інформацію. Метод головних компонент застосовується для перетворення набору пов'язаних змінних у менший набір непов'язаних змінних. Ідея полягає в тому, що багатовимірні датасети часто можна описати пов'язаними змінними, тому тільки декілька вимірів містять найбільш цінну інформацію.

Покроковий опис алгоритму виглядає так:

1. Створення датасету із зображеннями облич. Для прикладу будуть використовуватись зображення у відтінках сірого (grayscale) із LFW people's dataset (рис. 2.1) [14]. На відміну від кольорових зображень, які мають три канали – червоний зелений і синій, зображення в градаціях сірого має тільки один.

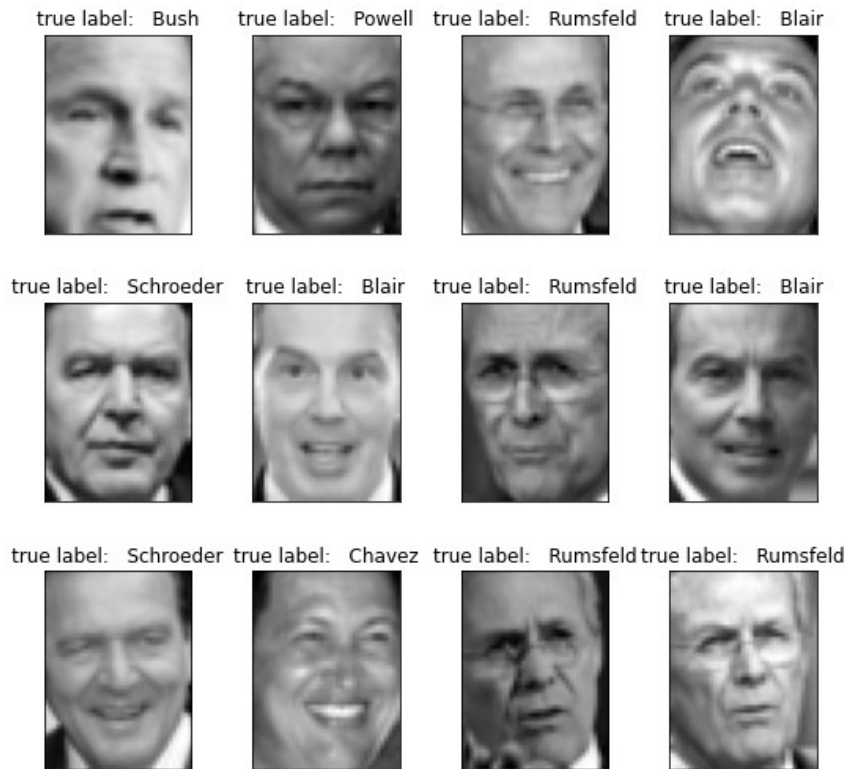


Рисунок 2.1 – Зображення для датасету

2. Конвертація зображень облич у вектори облич. Кожне зображення в градаціях сірого це матриця пікселів, значення кожного пікселя знаходиться в межах від 0 до 255, де 0 – це чорний колір, 255 – білий. Таку матрицю легко перетворити у вектор.

Нехай є m зображень розмірності $N \times N$, тоді отримаємо вектори x_1, x_2, \dots, x_m розмірністю $N^2 \times 1$ кожен. Вектор після перетворення буде мати вигляд як на рис. 2.2.

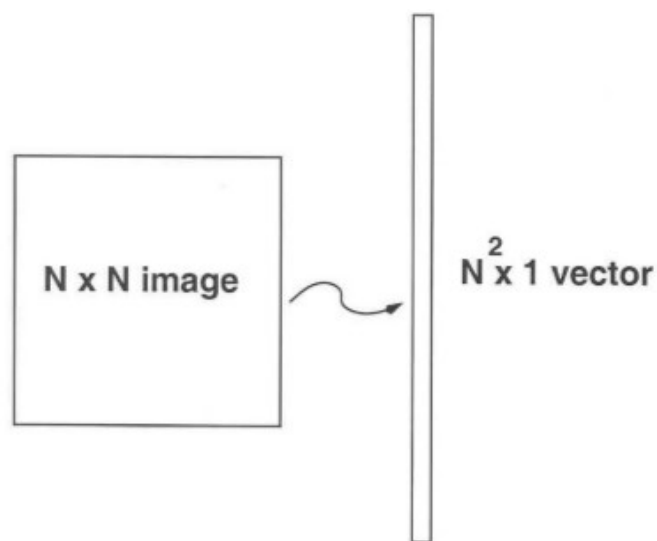


Рисунок 2.2 – Конвертація зображення у вектор

3. Нормалізація вектора облич. Вона відбувається у два етапи. Спочатку знаходиться “усереднене обличчя” (average face) (рис. 2.3) за формулою:

$$\psi = \frac{1}{m} \sum_{i=1}^m x_i$$

де m – кількість зображень в датасеті,

x_i – отримані на попередньому кроці вектори облич.

Усереднене лице містить у собі риси, спільні для всіх облич в датасеті.

Далі воно віднімається від кожного вектора обличчя:

$$a_i = x_i - \psi$$

З векторів різниці формується новий вектор A розмірністю $N^2 \times M$:

$$A = [a_1, a_2, \dots, a_m]$$

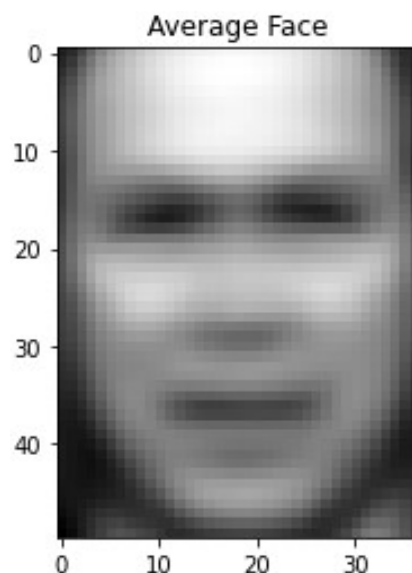


Рисунок 2.3 – Усереднене обличчя

4. Знаходження коваріаційної матриці шляхом множення A на транспоновану матрицю A^T . Розмірність матриці A $N^2 \times M$, отже розмірність A^T $M \times N^2$. Розмірність результату множення A на A^T рівна $N^2 \times N^2$, що дає N^2 власних векторів (eigenvectors) розмірністю N^2 . Із збільшенням N обчислення матриць такої великої розмірності потребуватиме багато обчислювальних ресурсів. Тому матриця коваріації обчислюється як добуток A^T і A . Її розмірність буде дорівнювати $M \times M$ (вважається, що $M \ll N^2$).

5. Знаходження власних значень (eigenvalues) і власних векторів (eigenvectors) для коваріаційної матриці C' :

$$C'u_i = \lambda_i u_i, i = 1, 2, \dots, m$$

де $C' = AA^T$.

6. Сортуювання власних векторів у порядку спадання їхніх власних значень. Вибір K власних векторів – eigenfaces (рис. 2.4), які відповідають K найбільшим власним значенням ($K < M$).

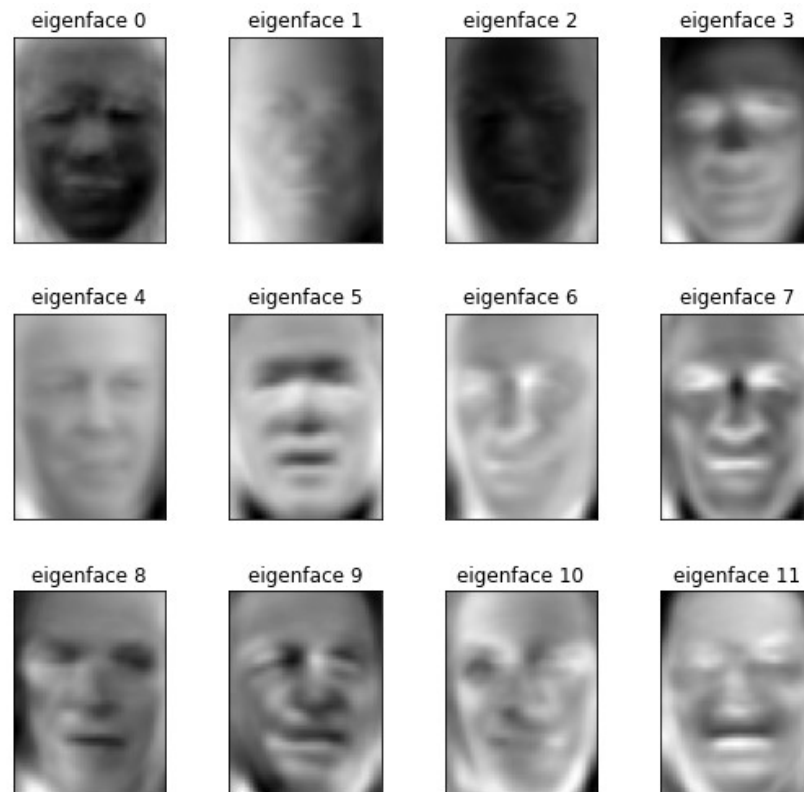


Рисунок 2.4 – Eigenfaces

7. Конвертація K власних векторів меншої розмірності у початкову розмірність використовуючи формулу

$$u_i = Av_i$$

8. Вираження кожного обличчя як лінійної комбінації K власних векторів. Кожне лице із датасету можна записати як суму зважених сум K eigenfaces і усередненого обличчя ψ :

$$x_i = \sum_{j=1}^K w_j u_j + \psi$$

де u_j – це Eigenfaces (рис. 2.4), w_j – ваги.

Приклад такої лінійної комбінації наведено на рис. 2.5.

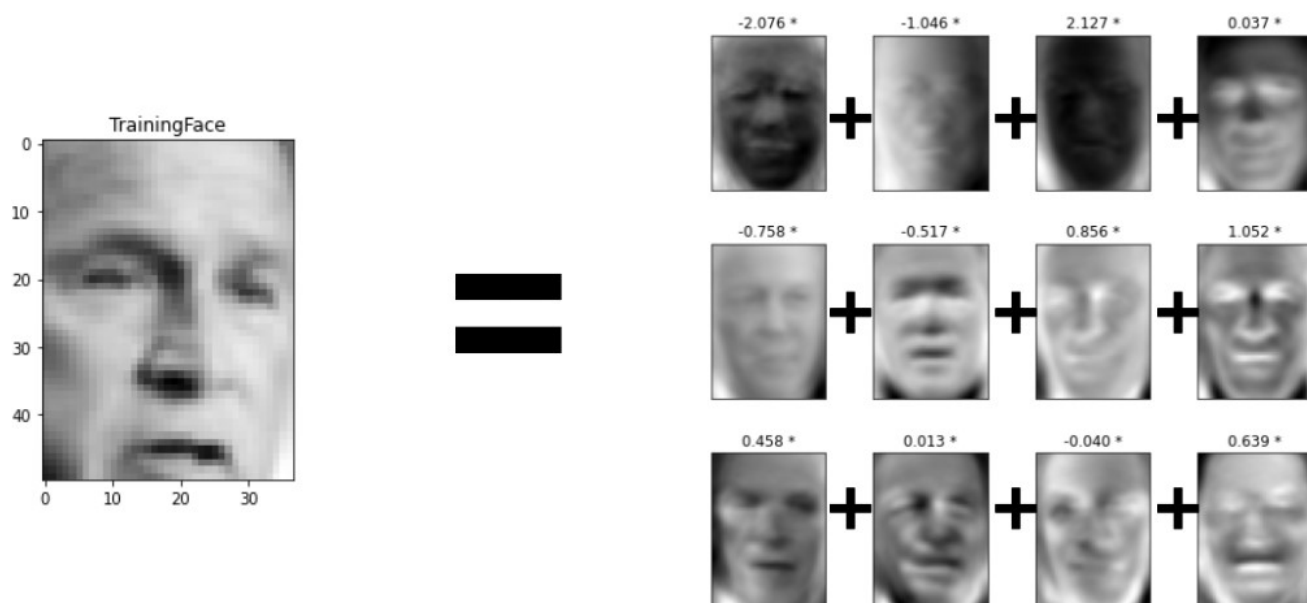


Рисунок 2.5 – Обличчя виражене як лінійна комбінація Eigenfaces

Після того як завершився етап навчання можна приступити до розпізнавання невідомих облич. Алгоритм розпізнавання обличчя за допомогою Eigenfaces наступний:

1. Зображення невідомого обличчя потрібно попередньо обробити: відцентрувати лице, підігнати розміри зображення до тих, що є в датасеті.
2. Перетворення підготовленого зображення лиця у вектор обличчя.
3. Нормалізація вектора обличчя аналогічно пункту 3 в алгоритмі.
4. Проектування нормалізованого вектора обличчя на eigenspace для отримання лінійної комбінації eigenfaces і зваженого вектора.

$$\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_k \end{bmatrix}$$

5. Знаходження різниці між векторами для облич із датасету і вектором для невідомого обличчя. Якщо вона менша ніж встановлена похибка, вважається, що обличчя розпізнане. Якщо більша, тоді обличчя залишається невідомим [15].

Для додавання нових зображень до датасету, процедуру знаходження eigenfaces необхідно виконати заново.

2.2 Алгоритм Fisherfaces

Алгоритм Fisherfaces є удосконаленням алгоритму Eigenfaces. Недоліком методу головних компонент, який використовується в Eigenfaces, є виявлення компонент, які насправді є пов'язаними. Наприклад, є два зображення однієї і тієї ж людини під різним освітленням. Алгоритм Eigenfaces може не розпізнати, що це одна й та сама людина.

В 1997 році Петер Бельхюмеор та ін. запропонували застосувати лінійний дискримінантний аналіз для розпізнавання облич і їхній алгоритм Fisherfaces показав кращі результати ніж Eigenfaces [16]. Ідея алгоритму Fisherfaces дуже проста: об'єкти, які належать до одного класу повинні бути тісно згруповані, тоді як об'єкти різних класів мають бути розташовані якомога далі одне від одного. Класом у цьому випадку є конкретна людина, об'єктами, які належать до класу є світлини, на яких вона зображена.

Покроковий опис алгоритму виглядає так:

1. Створення датасету.
2. Перетворення вхідних зображень у вектори. Нехай c класів, вектор X відображає всі класи, x_i відображає вектор одного зображення:

$$X = \{X_1, X_2, \dots, X_c\}$$

$$X_i = \{x_1, x_2, \dots, x_n\}$$

3. Обчислення середніх значень: загального середнього значення μ і середнього значення для кожного класу μ_i :

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\mu_i = \frac{1}{|X_i|} \sum_{x_j \in X_i} x_j$$

4. Обчислення значень матриці розсіювання між класами S_B і матриці розсіювання в межах класу S_W :

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

$$S_W = \sum_{i=1}^c \sum_{x_j \in X_i} (\mu_i - \mu)(\mu_i - \mu)^T$$

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		20

5. Знаходження проекції, в якій відстань між класами буде максимальною.
6. Знаходження власних значень, власних векторів і матриці трансформації.

На рис. 2.6 зображено матриці розсіювання для 3 класів. S_{B1} , S_{B2} , S_{B3} показують розсіювання між класами, S_{W1} , S_{W2} , S_{W3} – показують розсіювання всередині відповідного класу. Алгоритм Fisherfaces дозволяє збільшити відстань між різними класами, і згрупувати об'єкти, що належать до одного класу. Під час етапу розпізнавання близько розташовані об'єкти вважаються такими, що належать до одного класу. У випадку розпізнавання облич на фотографіях, які належать до одного класу, зображена одна й та ж людина [17].

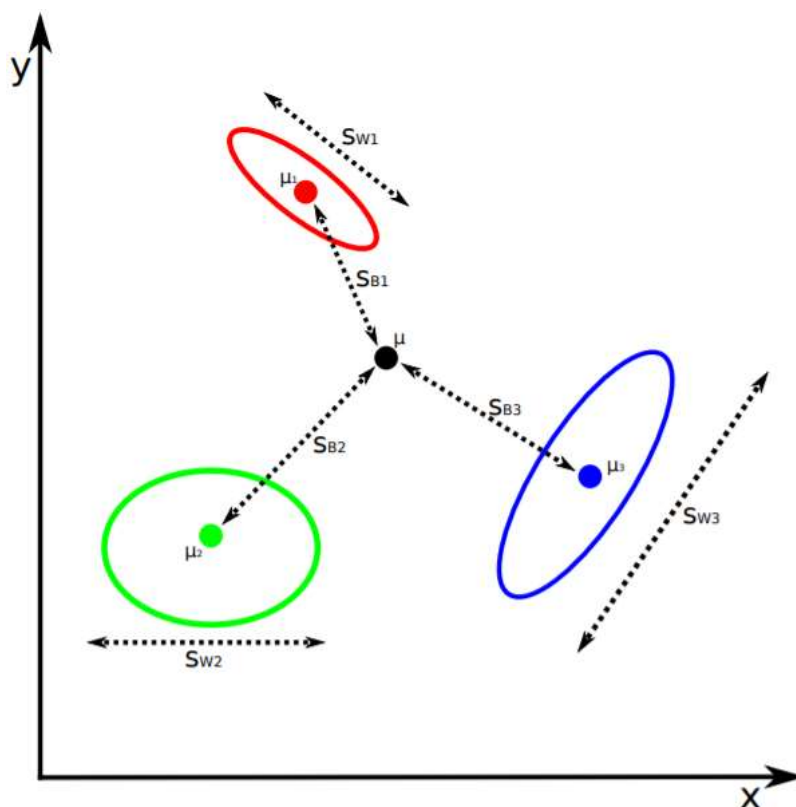


Рисунок 2.6 – Приклад розташування матриць розсіювання

2.3 Алгоритм LBPН

Локальні бінарні шаблони (Local Binary Patterns, LBP) – це простий та ефективний оператор, який використовується для класифікації текстур. Вперше алгоритм був описаний в 1994 році. Згодом було виявлено, що поєднання LBP і гістограм напрямлених градієнтів підвищує ефективність алгоритму. У порівнянні з вищерозглянутими Eigenfaces і Fisherfaces, які працюють із

зображенням як єдиним цілим, LBPН розглядає зображення попідсильно. Якщо порівнювати три вищезгадані алгоритми, то LBPН менш чутливий до змін освітлення, ніж Eigenfaces і Fisherfaces [18].

LBPН використовує чотири основні параметри:

1. Радіус: використовується для створення кругового локального бінарного шаблону і являє собою радіус навколо пікселя в центрі. Зазвичай він рівний 1. На рис. 2.7 наведено випадки, коли радіус дорівнює 2 або 3.
2. Кількість сусідів: кількість точок для побудови кругового локального бінарного шаблону. Потрібно зазначити, що чим більша кількість цих точок, тим більше обчислень прийдеться виконати. Зазвичай кількість сусідів рівна 8. Але вона може бути рівна 12, як, наприклад, на рис. 2.7.

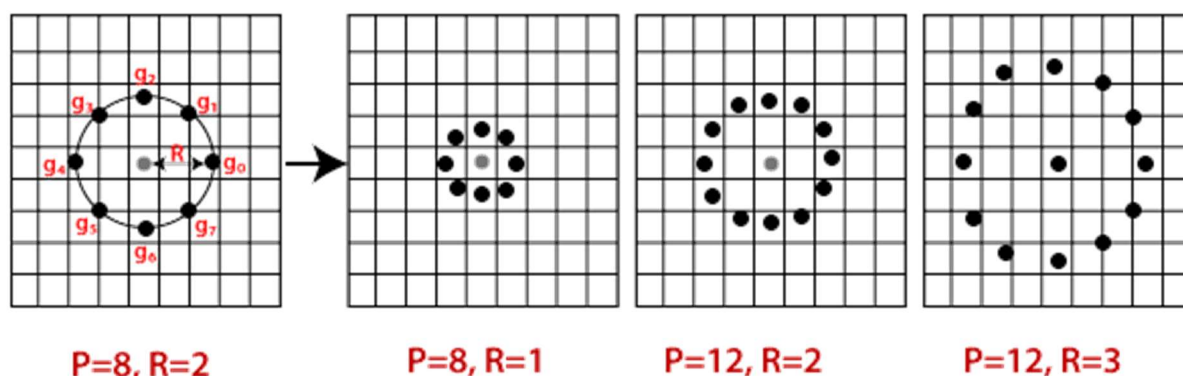


Рисунок 2.7 – LBPН з різними параметрами

3. Горизонтальна сітка: кількість клітинок у напрямку осі ОХ. Чим більше клітинок, тим дрібнішою буде сітка і тим більшою розмірність результуючого вектора ознак.
4. Вертикальна сітка: кількість клітинок у напрямку осі ОУ. Аналогічно горизонтальній: більше клітинок, отже дрібніша сітка і більша розмірність результату.

Першим етапом алгоритму є його навчання на датасеті. В датасет включаються зображення людей, які на етапі тестування будуть розпізнаватись. Далі вхідна світлина перетворюється у проміжне зображення. Для цього використовується “рухоме вікно”, параметри якого визначаються радіусом і кількістю сусідів.

Нехай маємо вхідне зображення у відтінках сірого, яке водночас є матрицею $n \times m$, заповненою значеннями інтенсивності кожного пікселя. Розмірність “рухомого вікна” дорівнює 3×3 . Значення інтенсивності центрального пікселя приймається за порогове значення (threshold). Далі значення кожного пікселя сусіднього до центрального порівнюється з пороговим значенням. Якщо значення більше ніж порогове, то у відповідну клітинку записується 1, якщо менше, то 0. На рис. 2.8 порогове значення дорівнює 82. Якщо розглянути першу клітинку в першому рядку із значенням 137, то воно є більшим ніж порогове 82. Тоді на місці 137 записується 1. Аналогічні дії виконуються із рештою сусідів. Далі потрібно об'єднати всі отримані бінарні значення у нове двійкове число. На рис. 2.8 це буде 11100001. Дане число необхідно перевести в десяткову систему числення і записати в початкову матрицю замість центрального значення. Вікно пересувається і процедура повторюється заново.

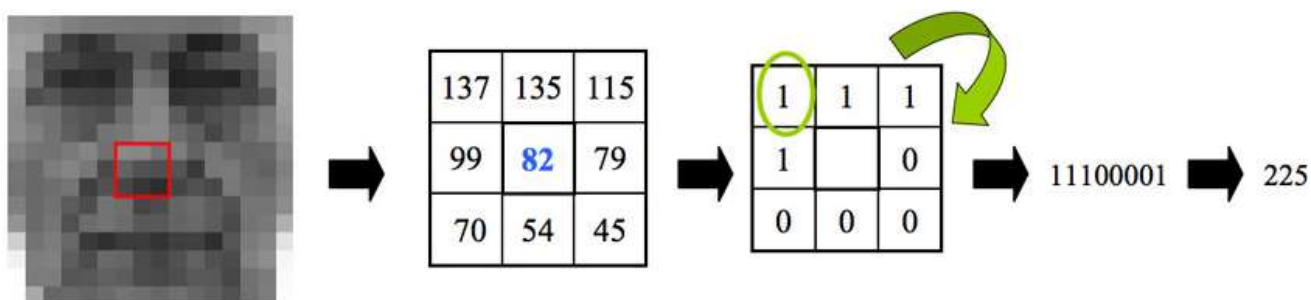


Рисунок 2.8 – Процедура знаходження бінарних значень

Зображення, отримане після перетворень, використовується для побудови гістограми. Навчання алгоритму можна вважати завершеним. В результаті для кожного зображення із датасету має бути створена гістограма.

На етапі тестування вхідне зображення проходить ту ж процедуру, що і дані з навчального набору. Далі гістограма невідомого зображення порівнюється із гістограмами навчальних зображень. Найбільш подібні гістограми вважаються такими, що належать одній людині. Для знаходження відстані між гістограмами використовується евклідова відстань, яка вираховується за формулою:

$$D = \sqrt{\sum_{i=1}^n (hist1_i - hist2_i)^2}$$

Відстань між діаграмами також використовується для вимірювання рівня “довіри” до результатів. Чим нижчим є значення “довіри”, тим ближчими є дві гістограми. Також це значення дозволяє визначити наскільки правильним є отриманий результат. Розпізнавання можна вважати успішним, якщо “довіра” є меншою ніж граничне значення [18].

2.4 Метод Віоли-Джонса

Метод Віоли-Джонса був розроблений Полом Віолою і Майклом Джонсом у 2001 році [19]. Цей алгоритм дозволяє виявляти об’єкти на зображеннях в режимі реального часу. Незважаючи на те, що метод був розроблений 20 років тому, він все ще залишається потужним засобом для виявлення облич на зображеннях. До початку активного застосування глибокого навчання для виявлення і розпізнавання облич, даний метод мав високі показники ефективності та швидкості роботи і вважався одним із найкращих.

Потрібно звернути увагу, що метод Віоли-Джонса призначений для виявлення облич, тобто він визначає місцезнаходження обличчя на зображенні. Якщо стоїть завдання розпізнавання обличчя, тобто знаходження конкретного власника обличчя, тоді цей метод можна комбінувати з алгоритмами розпізнавання облич. Оскільки метод Віоли-Джонса є доволі ефективним, при комбінуванні з іншими методами можна отримати значний приріст продуктивності.

Основні принципи методу Віоли-Джонса такі:

- Використання інтегральних зображень для пришвидшення виконання обчислень.
- Для виявлення обличчя використовуються ознаки Хаара.

- Для вибору ознак, які найкраще підходять для пошуку об'єкта, використовується бустинг.
- Знайдені під час сканування ознаки потрапляють на вхід класифікатора, який приймає їх чи відкидає.
- Для швидкого відбракування вікон використовуються каскади ознак.

Для зручності обчислень в методі Віоли-Джонса використовуються зображення в інтегральному вигляді. Представлення світлин у такому вигляді дозволяє швидко обчислювати сумарну інтенсивність довільного прямокутника на зображенні, причому яким би не був розмір прямокутника, час розрахунку завжди однаковий.

Інтегральне зображення являє собою матрицю, розміри якої ідентичні розмірам вхідного зображення. Кожен елемент цієї матриці дорівнює сумі інтенсивностей всіх пікселів, які знаходяться лівіше і вище від даного елемента, а також значення самого елемента. Час розрахунку інтегральної матриці пропорційний кількості пікселів зображення.

На рис. 2.9 зображена матриця початкового зображення, а також його інтегральна матриця. Значення елемента на перетині першого рядка і другого стовпця обчислюється як сума елемента зліва, який рівний 4, і значення поточного елемента, який рівний 1. В сумі виходить 5. Якщо взяти елемент на перетині другого рядка і другого стовпця, то він обчислюється так: $4 + 1 + 0 + 4 = 9$. Аналогічно знаходяться решта елементів інтегральної матриці.

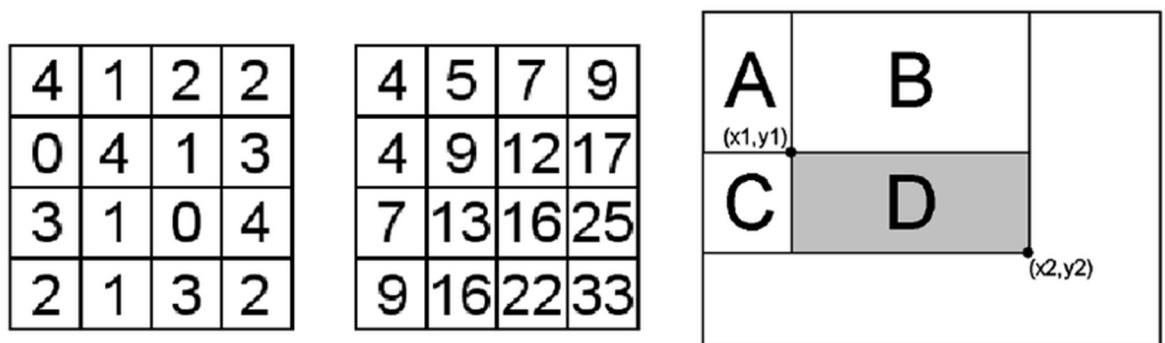


Рисунок 2.9 – Отримання інтегрального зображення

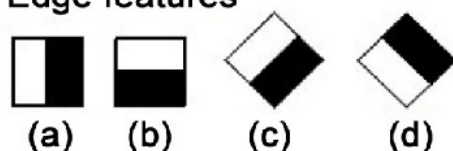
По інтегральній матриці можна швидко знаходити суму пікселів довільного прямокутника з довільною площею. Нехай потрібно знайти суму пікселів виділеного сірим прямокутника (рис. 2.9): $\text{Grey_Area} = D - C - B + A$.

В методі Віоли-Джонса використовуються прямокутні ознаки Хаара (Haar-like-features) для отримання даних про перепад інтенсивності на зображенні (рис. 2.10). Значення ознаки Хаара знаходиться за формулою:

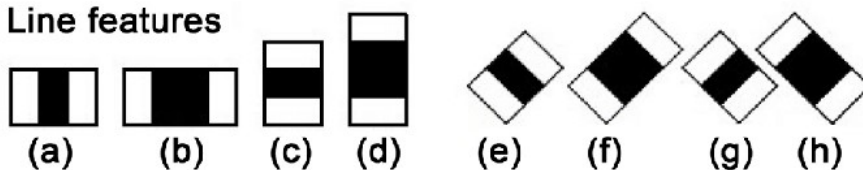
$$F = X - Y$$

де X – це сума значень інтенсивностей, які знаходяться під білою частиною ознаки, Y – сума значень інтенсивностей, які знаходяться під чорною частиною. Ознаки Хаара накладаються на інтегральне зображення.

1. Edge features



2. Line features



3. Center-surround features

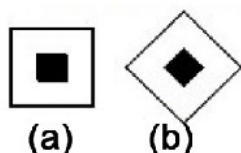


Рисунок 2.10 – Ознаки Хаара

В методі Віоли-Джонса використовується принцип скануючого вікна, який полягає у наступному:

1. Для початку вибирається зображення, розмір вікна сканування, ознаки Хаара, які будуть застосовуватись. Для зображення знаходиться інтегральне представлення.
2. Вікно сканування поступово пересувається по зображенню з кроком, наприклад, 1 піксель.
3. В кожному вікні обчислюються значення прямокутних ознак Хаара. При цьому розташування ознак у вікні та їхній розмір змінюються.
4. Пункти 2 і 3 повторюються для скануючих вікон різних розмірів.

5. Всі знайдені значення ознак попадають на вхід до класифікатора, який визначає, чи є у вікні обличчя [20].

Процедура застосування всіх ознак Хаара у різних масштабах до зображення потребує немало обчислювальних ресурсів. Для того, щоб їх зменшити, алгоритм навчається і визначає, які ознаки доречно застосовувати.

Далі необхідно розглянути модель машинного навчання, яка вирішуватиме проблему класифікації. В даному випадку алгоритм має встановити певні закономірності, за якими можна буде визначити чи на зображенні присутнє обличчя чи його немає.

Класифікатори можна поділити на слабкі – такі класифікатори роблять багато помилок під час розділення об’єктів на класи, і сильні – вони практично безпомилково визначають до якого класу належить об’єкт. Бустинг – це процедура побудови композиції класифікаторів, у якій кожен наступний класифікатор покликаний компенсувати недоліки поєднання всіх попередніх. Класифікатори поєднуються в ланцюг, кожен наступний класифікатор, крім першого, вчиться на помилках попереднього. Метою бустингу є поєднання слабких класифікаторів в один сильний класифікатор.

Під час навчання в методі Віоли-Джонса використовується удосконалений алгоритм бустингу AdaBoost (adaptive boosting). AdaBoost використовує довільну кількість класифікаторів. Для демонстрації роботи алгоритму бустингу можна скористатись таким рівнянням:

$$F(x) = w_1 f_1(x) + w_2 f_2(x) + w_3 f_3(x) + \dots$$

де w_1, w_2, w_3 – це ваги, $f_1(x), f_2(x), f_3(x)$ – це ознаки, які виступають в ролі слабких класифікаторів. Сума класифікаторів $F(x)$ – це сильний класифікатор.

Нехай є 10 зображень, на 5 із них зображені обличчя, на решті 5 обличчя немає. З допомогою, наприклад, першої ознаки $f_1(x)$ модель знаходить обличчя на 3 фото із 5, де воно дійсно є, і на 2 фото із 5, де його немає. На наступному кроці вибирається ще одна ознака, наприклад, друга $f_2(x)$, для того щоб доповнити першу. Вага зображень, які були неправильно визначені, зростає, а

друга ознака вибирається так, щоб точність знаходження обличчя на фото зросла.

Послідовно з'єднані класифікатори утворюють каскад. Каскадування класифікаторів дозволяє пришвидшити процес виявлення обличчя і збільшити його точність. Нехай вибране скануюче вікно і в ньому йде пошук найхарактернішої ознаки Хаара, яка вказує на наявність обличчя у вікні. Якщо цієї ознаки немає, тоді вікно автоматично відкидається, а наступні ознаки для цього вікна не обраховуються. Для прикладу, на рис. 2.11 є класифікатор F1. Якщо він визначає, що у вікні немає обличчя, то класифікатор F2 уже не перевіряє дане вікно [21].

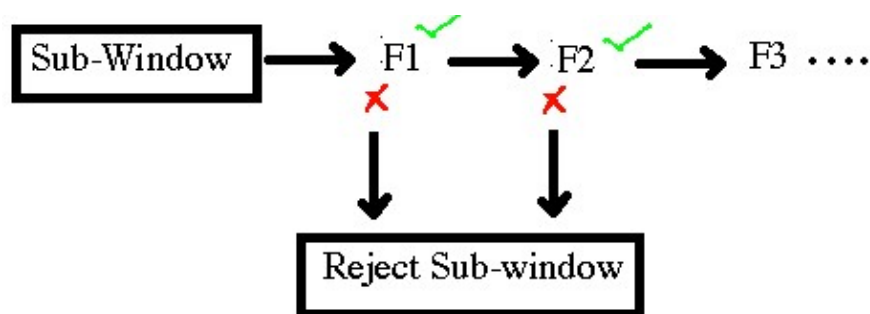


Рисунок 2.11 – Каскад класифікаторів

2.5 Застосування глибокого навчання для розпізнавання облич

Застосування глибокого навчання для класифікації зображень, в тому числі для виявлення і розпізнавання обличч показало кращі результати у порівнянні з традиційними методиками комп'ютерного зору.

Згорткова нейронна мережа (CNN, ConvNet) – це різновид штучних нейронних мереж. Нейронну мережу можна вважати CNN, якщо вона має хоча б один згортковий шар. Згорткові мережі показали високу ефективність під час вирішення задач розпізнавання і класифікації зображень. Крім розпізнавання обличч CNN застосовують в системах комп'ютерного зору для роботів та самокерованих автомобілів.

Однією з перших згорткових нейронних мереж була мережа LeNet, структуру якої запропонував Ян Лекун у 1989 році [22]. Першими практичними

застосуваннями LeNet було розпізнавання поштових індексів та цифр. Протягом останніх років було запропоновано кілька нових архітектур згорткових мереж, які є кращими ніж LeNet, проте основні принципи їх функціонування залишились практично незмінними ще з часів першої мережі.

На рис. 2.12 зображено згорткову нейронну мережу, архітектура якої подібна до архітектури LeNet. Ця мережа визначає до якої категорії належить вхідне зображення. Як видно з рис. 2.12 згорткова мережа включає в себе 4 основні операції:

1. Згортка (Convolution).
2. Застосування нелінійної функції ReLU (rectified linear unit).
3. Об'єднання (Pooling)
4. Класифікація (Classification)

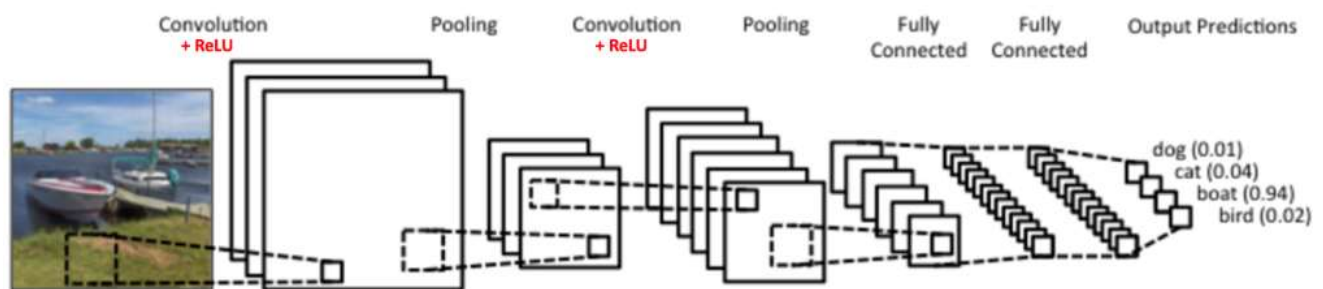


Рисунок 2.12 – Згорткова нейронна мережа

Згортка (convolution) – перша операція, якій піддаються вхідні дані. Основною метою згортки є отримання ознак з вхідного зображення. Згортка зберігає просторовий зв'язок між пікселями шляхом вивчення особливостей зображення за допомогою фільтрів.

Кожне зображення можна представити у вигляді матриці значень пікселів. Для прикладу береться світлина в градаціях сірого, яку можна представити за допомогою однієї матриці. Зазвичай для розпізнавання лиця достатньо зображення у відтінках сірого.

Якщо розглядати матрицю зображення розмірністю 5 x 5 і фільтр або детектор ознаки, розмірністю 3 x 3, то застосування фільтра виглядає так: в матриці вибирається квадрат розмірністю 3 x 3 і поелементно виконується множення числа із матриці і числа із детектора ознаки, всі добутки додаються і

записуються в результуючу матрицю 3 x 3, яка ще називається картою ознак (feature map). Приклад матриці і фільтра наведений на рис. 2.13.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

Рисунок 2.13 – Матриця зображення і фільтр

Квадрати в матриці вибираються починаючи із першого елемента зліва. Це можна представити як накладання фільтра на матрицю. Після того як сума добутків записана, детектор ознаки пересувається вправо на одну клітинку. Якщо рядок закінчився, тоді він пересувається на початок і опускається на одну клітинку вниз, виконання множення і додавання продовжується. Покрокове застосування фільтра зображене на рис. 2.14.

Figure 1: A 5x5 grid of numbers (1, 0) with associated multiplication factors (x1, x0) and a 3x3 grid of numbers (4, 3, 4).

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

4		

Figure 2: A 5x5 grid of numbers (1, 0) with associated multiplication factors (x1, x0) and a 3x3 grid of numbers (4, 3, 4).

1	1 _{x1}	1 _{x0}	0 _{x1}	0
0	1 _{x0}	1 _{x1}	1 _{x0}	0
0	0 _{x1}	1 _{x0}	1 _{x1}	1
0	0	1	1	0
0	1	1	0	0

4	3	

Figure 3: A 5x5 grid of numbers (1, 0) with associated multiplication factors (x1, x0) and a 3x3 grid of numbers (4, 3, 4).

1	1	1 _{x1}	0 _{x0}	0 _{x1}
0	1	1 _{x0}	1 _{x1}	0 _{x0}
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1	1	0
0	1	1	0	0

4	3	4

Figure 4: A 5x5 grid of numbers (1, 0) with associated multiplication factors (x1, x0) and a 3x3 grid of numbers (4, 3, 4).

1	1	1	0	0
0 _{x1}	1 _{x0}	1 _{x1}	1	0
0 _{x0}	0 _{x1}	1 _{x0}	1	1
0 _{x1}	0 _{x0}	1 _{x1}	1	0
0	1	1	0	0

4	3	4
2		

Figure 5: A 5x5 grid of numbers (1, 0) with associated multiplication factors (x1, x0) and a 3x3 grid of numbers (4, 3, 4).

1	1	1	0	0
0	1 _{x1}	1 _{x0}	1 _{x1}	0
0	0 _{x0}	1 _{x1}	1 _{x0}	1
0	0 _{x1}	1 _{x0}	1 _{x1}	0
0	1	1	0	0

4	3	4
2	4	

Figure 6: A 5x5 grid of numbers (1, 0) with associated multiplication factors (x1, x0) and a 3x3 grid of numbers (4, 3, 4).

1	1	1	0	0
0	1	1	1	0
0	0	1 _{x1}	1 _{x0}	1 _{x1}
0	0	1 _{x0}	1 _{x1}	0 _{x0}
0	1	1 _{x1}	0 _{x0}	0 _{x1}

4	3	4
2	4	3
2	3	4

Рисунок 2.14 – Покрокове застосування фільтра

Очевидно, що застосування фільтрів з різними значеннями для одного й того ж зображення дасть різні карти ознак. За допомогою фільтрів можна виконувати такі операції як виявлення країв, загострення рис, розмиття і т. д. На рис. 2.15 продемонстровано застосування різних детекторів ознак до одного зображення. На практиці згорткові нейронні мережі вчать самостійно визначати значення для фільтрів впродовж процесу навчання, хоча деякі параметри все ж доводиться встановлювати програмісту. Такими параметрами можуть бути кількість фільтрів, розмірність фільтра, архітектура мережі і т. п. Чим більшою є кількість фільтрів тим більше ознак можна виявити і тим краще згорткова мережа буде справлятися із розпізнаванням образів на нових зображеннях [23].

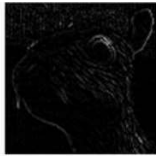

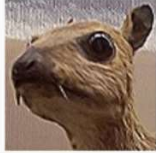

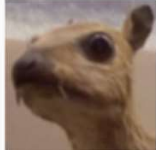
Edge detection	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Рисунок 2.15 – Застосування різних фільтрів до одного зображення

Розмір карти ознак залежить від 3 параметрів, які потрібно вказати перед виконанням етапу згортки:

1. Глибина: відповідає кількості фільтрів, які використовуватимуться під час виконання згортки. Варто звернути увагу, що кількість карт ознак відповідає кількості фільтрів.

2. Крок: дорівнює кількості пікселів, на яку необхідно перемістити фільтр на оригінальному зображенні. Чим більший крок, тим меншою буде карта ознак.
3. Заповнення нулями: в деяких випадках буває зручно доповнити матрицю вхідного зображення нулями по краях для того, щоб було простіше застосовувати фільтр до пікселів на межі. Цей параметр буває корисним, якщо потрібно збільшити вихідну карту ознак [24].

Після операції згортки застосовується функція ReLU (rectified linear unit) або інакше функція випрямлення (рис. 2.16). Вона відіграє роль функції активації і має наступний вигляд:

$$f(x) = \max(0, x)$$

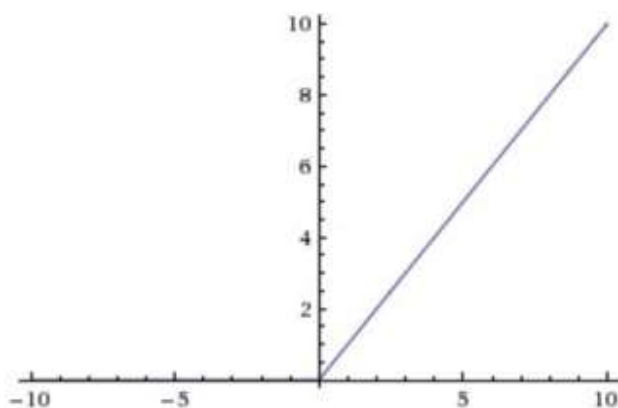


Рисунок 2.16 – Графік функції ReLU

Функція ReLU застосовується поелементно, у випадку роботи із зображеннями – попівсельно. В результаті всі пікселі з негативними значеннями на карті ознак замінюються нулями. ReLU застосовується для введення нелінійності в згорткову нейронну мережу, оскільки більшість реальних даних, на яких вчиться нейронна мережа, є нелінійними. Етап згортки, де виконується поелементне множення матриць і додавання, є лінійною операцією. І для того щоб врахувати нелінійність вводиться функція ReLU.

Інші нелінійні функції, такі як сигмоїда або гіперболічний тангенс, можуть застосовуватись замість ReLU, але було встановлено, що в більшості випадків вона дає кращий результат.

Операція об'єднання або пулінгу (pooling) зменшує розмірність кожної карти ознак і залишає найбільш цінну інформацію. Існує декілька видів пулінгу: вибір максимального значення, середнього значення, сумування та ін.

У випадку пулінгу з вибором максимального значення, беруться сусідні значення в матриці, наприклад, вікно розміром 2 x 2 (рис. 2.17), і серед них вибирається максимальне. Замість того, щоб вибирати найбільший елемент, можна взяти середнє значення, або суму всіх елементів у вікні. На практиці пулінг з вибором максимального елемента працює краще.

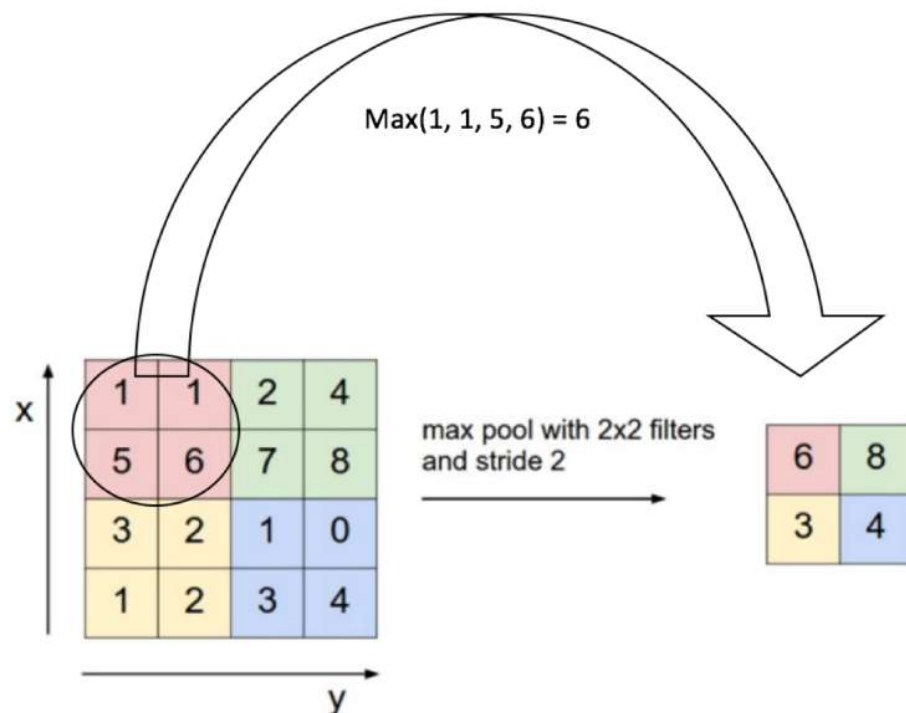


Рисунок 2.17 – Застосування пулінгу із вибором Max

Як і у випадку із застосуванням згортки, вікно переміщається вправо і вниз на значення кроку. На рис. 2.17 крок вздовж осі x, як і крок вздовж осі y, дорівнює дві клітинки. Ще однією схожістю з операцією згортки є зменшення розмірності карти ознак. На рис. 2.17 матриця до пулінгу мала розмірність 4 x 4, після пулінгу її розмірність зменшилась до 2 x 2. Варто звернути увагу на те, що пулінг застосовується до кожної карти ознак окремо.

Загалом пулінг є дуже важливою операцією, він дозволяє:

- зменшити кількість параметрів та обчислень в згортковій мережі, і тим самим контролювати можливість виникнення перенавчання.

Перенавчання – це явище, коли модель добре класифікує дані, на яких

вона навчалась, але показує гірші результати на даних, які вона обробляє вперше.

- зробити згорткову мережу нечутливою до незначних спотворень, переміщень на вхідному зображенні. Спотворення не змінять результатів пулінгу, оскільки ця операція визначає максимум або середнє значення сусідніх пікселів.
- отримати незмінне представлення зображення. Це дає можливість успішного розпізнавання об'єктів, незалежно від того, де вони знаходяться на зображенні [23].

Структура мережі LeNet є наступною: спочатку йде одна черга із операцій згортки, ReLU і пулінгу. Потім ці ж операції повторюються, але виконуються вони вже з результатами, отриманими після першої черги. Загалом дані операції можуть повторюватись декілька разів, залежно від архітектури згорткової нейронної мережі. Для прикладу, сучасні згорткові нейронні мережі можуть мати десятки шарів згортки, ReLU і пулінгу. Також можливий варіант, коли декілька разів підряд виконуються операції згортки і ReLU, а вже після цього застосовується пулінг. У цілому, чим більше разів застосовується операція згортки, тим складніші ознаки зможе розпізнавати нейронна мережа. Наприклад, згорткова мережа при застосуванні згортки до вхідного зображення буде виявляти краї, наступна операція згортки буде виявляти прості форми, які будуть використовуватись для виявлення складніших рис, таких як форма обличчя.

Отримані за допомогою операцій згортки і пулінгу ознаки переходять до повністю зв'язаного шару (Fully Connected Layer). Цей шар являє собою багатошаровий перцептрон (multilayer perceptron) – різновид нейронних мережі, який має принаймні три шари: вхідний, прихований і вихідний шар і не має зворотних зв'язків. Повністю зв'язаний означає, що кожен нейрон попереднього шару з'єднаний з кожним нейроном із наступного шару. Перцептрон використовує функцію активації Softmax, хоча загалом можуть використовуватись інші функції. Карти ознак, створені на етапах згортки і

пулінгу, передаються на входи повністю зв'язаного шару. Цей шар вирішує до якого класу належить вхідне зображення. Дане рішення приймається виходячи із застосування “знань”, отриманих на етапі навчання.

Важливу роль в процесі навчання згорткової нейронної мережі відіграє метод зворотного поширення помилки (backpropagation). З допомогою цього методу виконується корекція значень ваг і фільтрів. Правильне налаштування цих значень збільшує точність результатів, які видає нейронна мережа.

2.6 Порівняння розглянутих алгоритмів розпізнавання облич

Для того, щоб порівняти вищеописані алгоритми, необхідно спочатку визначити їхні сильні і слабкі сторони.

Застосування методу головних компонент в алгоритмі Eigenfaces дозволяє зменшити розмірність матриць над якими виконуються обчислення. Через це для виконання алгоритму потрібно мало обчислювальних ресурсів і його легко реалізувати програмно. Оскільки алгоритм приймає на вхід матрицю інтенсивностей кожного пікселя, зображення не потрібно заздалегідь обробляти. Серед недоліків алгоритму можна зазначити такі: чутливість до освітлення, масштабу, фону, повороту обличчя. Найкращі результати Eigenfaces показував на фото, зроблених в “ідеальних” умовах з рівномірним освітленням і однотонним фоном. Під час реальної фотозйомки ці умови не завжди виконуються. Оскільки зображення обробляється цілісно, неоднотонний фон може спотворювати результати.

Алгоритм Fisherfaces є менш чутливим до поганого освітлення і зміни виразу обличчя ніж Eigenfaces завдяки застосуванню лінійного дискримінантного аналізу. Але загалом переваги і недоліки цих алгоритмів практично однакові.

Алгоритм LBPН, на відміну від Eigenfaces і Fisherfaces, аналізує окремі ділянки зображення. Оскільки LBPН шукає різницю інтенсивностей пікселів, вона не змінюється якщо обличчя погано освітлене. Це дозволяє розпізнавати

затінені обличчя краще ніж Fisherfaces. Проте такий спосіб аналізу зображення робить алгоритм LBPН чутливим до шуму на фото. Якщо зображення є бляклим, тобто не має яскраво виражених перепадів інтенсивностей, LBPН не розпізнає на ньому обличчя.

Метод Віюли-Джонса опрацьовує зображення досить ефективно і швидко завдяки каскадам Хаара і бустингу. Перевагою методу є можливість виявлення обличчя в якій би частині фото воно не знаходилося. Точність знаходження облич, повернутих на кут більший ніж 30° , набагато менша у порівнянні з обличчями, які дивляться прямо в камеру.

Згорткові нейронні мережі не дарма є дуже популярними, оскільки точність розпізнавання ними облич є дуже високою у порівнянні з вищезазначеними методами. Проте для підвищення точності модель нейронної мережі потрібно навчити на великому датасеті. Процес навчання потребує багато обчислювальних ресурсів і часу.

Для наочності вищеописані характеристики було занесено в таблицю 2.1. Тож виходячи із даної таблиці ефективними і простими в реалізації є алгоритм LBPН і метод Віюли-Джонса. Також високий показник точності розпізнавання у згорткових нейронних мереж, проте їх важче реалізувати.

Таблиця 2.1. – Порівняння алгоритмів

Алгоритм	Переваги	Недоліки
Eigenfaces	Простий в реалізації, потребує мало обчислювальних ресурсів, зображення обробляється цілісно, не виокремлюючи окремі ознаки.	Для алгоритму підходять тільки зображення, де обличчя дивиться вперед і розташоване по центру, алгоритм чутливий до освітленості на фото, тіней і масштабу.

Продовження таблиці 2.1

Fisherfaces	Простий в реалізації, потребує мало обчислювальних ресурсів. Показує хороші результати при поганій освітленості.	Найкраще підходять зображення обличчя в анфас. Чутливість до освітленості невелика, але все ж присутня.
LBPН	Простий в реалізації, потребує мало обчислювальних ресурсів. Показує хороші результати при поганій освітленості.	Алгоритм працює краще, коли обличчя на зображенні дивиться вперед. Точність розпізнавання для фото низької якості суттєво знижується.
Viola-Jones	Простий в реалізації, потребує мало обчислювальних ресурсів. Швидкий. Виявляє обличчя незалежно від його місцезнаходження на зображенні.	При повороті обличчя на більше ніж 30° точність знаходження обличчя значно зменшується. Призначений для знаходження облич, необхідно комбінувати з методами для розпізнавання.
CNN	Висока точність розпізнавання облич. Модель CNN можна натренувати на розпізнавання людей в окулярах, масках і т. п.	Для підвищення точності модель необхідно натренувати на великій кількості зображень. Невдало вибрані параметри можуть призвести до перенавчання. Навчання може зайняти багато часу. Складніший в реалізації.

ВИСНОВОК ДО РОЗДІЛУ 2

У другому розділі було розглянуто існуючі алгоритми для розпізнавання облич. Eigenfaces, Fisherfaces, LBPН, метод Віоли-Джонса – це алгоритми традиційного комп’ютерного зору, згорткові нейронні мережі – це представник глибоких нейронних мереж, які застосовуються для глибокого навчання.

Eigenfaces для свого часу був досить потужним алгоритмом розпізнавання облич. Застосування методу головних компонент значно зменшувало обчислювальні витрати. Для Eigenfaces підходять тільки фото обличчя в анфас, зроблені при хорошому освітленні. Зважаючи на ці обмеження на зміну Eigenfaces прийшов алгоритм Fisherfaces. Він включав застосування лінійного дискримінантного аналізу, який дозволив зробити алгоритм менш чутливим до змін освітлення на зображеннях.

Одним із найпростіших алгоритмів для розпізнавання є LBPН. Якщо порівнювати його із двома вищезгаданими алгоритмами, то він є точнішим і ефективнішим. Недоліком LBPН є чутливість до шуму на зображеннях.

Метод Віоли-Джонса має багато позитивних ознак, він може виявляти обличчя де б воно не було розташоване на світлині. Процес виявлення займає мало часу. Каскади Хаара, які застосовуються в методі, можна натренувати на знаходження облич в профіль. Точність методу залежить від величини датасету. Чим більше зображень на одну людину зібрано для навчання класифікатора, тим точнішими будуть результати.

Застосування згорткових нейронних мереж для розпізнавання облич є найпопулярнішим на сьогоднішній день. Якщо модель натренована на великому датасеті, то точність результатів може досягати 97%. Але варто пам’ятати, що тренування такої моделі може зайняти декілька годин, а то й днів.

Тож враховуючи такі показники як час розробки системи, точність, кількість зображень для навчання, швидкість роботи, час, витрачений на навчання, лідирують метод Віоли-Джонса, LBPН і CNN.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		38

РОЗДІЛ 3

РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ

3.1 Опис засобів реалізації

Виходячи з аналізу, проведеного в попередньому розділі, за основу беруться метод Віюлі-Джонса і алгоритм LBPН через такі позитивні якості як: простота в реалізації, навчання алгоритму займає небагато часу, для навчання потрібно відносно небагато зображень на одну людину.

Для того, щоб система могла розпізнавати обличчя, потрібно пройти декілька етапів:

1. Збір зображень для датасету
2. Навчання класифікатора
3. Розпізнавання

Є ще один важливий проміжний етап – це знаходження обличчя на зображенні. Спочатку потрібно знайти обличчя, а потім його можна використовувати для навчання класифікатора і розпізнавання.

В ролі класифікатора будуть виступати каскади Хаара із методу Віюлі-Джонса. Алгоритм LBPН буде використовуватись для розпізнавання облич. Обидва алгоритми були розглянуті в попередньому розділі.

Для реалізації буде використовуватись бібліотека OpenCV і мова програмування Python.

OpenCV – це бібліотека комп'ютерного зору з відкритим кодом. Це досить популярна бібліотека, яка містить декілька сотень оптимізованих алгоритмів. Серед них є класичні і сучасні алгоритми.

Python – це мова програмування високого рівня, яка теж на сьогодні є дуже популярною. Головними перевагами цієї мови є простий синтаксис, наявність великої кількості бібліотек і кросплатформність.

В якості середовища розробки було обрано IDE PyCharm Community Edition через те, що воно безкоштовне і зручне у користуванні. PyCharm надає

такі можливості, як автодоповнення, підсвічування синтаксису і помилок, навігація по проєкту, рефакторинг коду, вбудований дебагер, інструменти для юніт-тестування та ін.

3.2 Опис проєкту

Проект складається із чотирьох файлів на мові Python, папок, де зберігаються навчений класифікатор, стандартні класифікатори із бібліотеки OpenCV та зображення для навчання класифікатора. Структуру проєкту можна побачити на рис. 3.1.

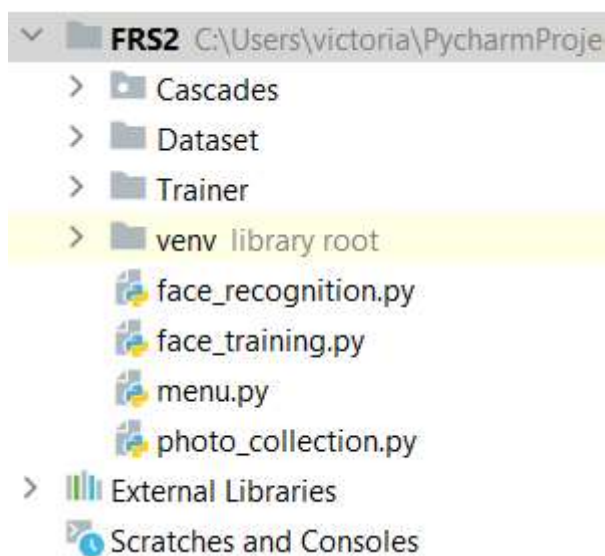


Рисунок 3.1 – Файли проєкту в середовищі PyCharm

Файл `photo_collection.py` містить клас `PhotoCollector`, який робить фото користувача з веб-камери і додає їх до датасету, на якому потім буде навчатись класифікатор.

В конструкторі класу застосовується такі функції:

- `cv2.VideoCapture(0, cv2.CAP_DSHOW)` – вмикає камеру.
- `cv2.CascadeClassifier('Cascades/haarcascade_frontalface_default.xml')` – дана функція завантажує класифікатор, який знаходиться в каталозі `Cascades`. В OpenCV є близько 20 попередньо навчених каскадів Хаара на різних об'єктах. Серед них є каскади для обличчя в анфас, профіль, для пошуку посмішки, очей, людини у повний зріст та ін. Файли з каскадами завантажуються за замовчуванням разом з бібліотекою.

Метод `photos_taker()` робить фотографії обличчя. В ньому використовуються такі функції бібліотеки OpenCV:

- `cam.read()` – зчитує кадр із камери.
- `cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)` – перетворює кольорове зображення на зображення у відтінках сірого.
- `face_detector.detectMultiScale(gray, 1.3, 5)` – функція, яка знаходить обличчя на зображенні. Повертає список прямокутників, в межах яких знаходяться знайдені обличчя. Перший параметр `gray` позначає зображення в градаціях сірого, на якому функція буде шукати лиця. Другий параметр називається `scaleFactor` і відповідає за те, у скільки разів буде зменшуватись зображення. Чим меншим буде його значення тим ретельніше буде йти пошук обличчя, а час обробки одного зображення збільшиться. В даному випадку `scaleFactor=1.3`, що означає зменшення зображення на 30%. Ще один параметр `minNeighbors` визначає скільки сусідів повинен мати прямокутник, в якому можливо міститься обличчя. Чим більшим є значення цього параметра, тим менше прямокутників буде виявлено, але в них з більшою точністю будуть знаходитись обличчя.
- `cv2.imwrite(dir_name + str(count) + ".jpg", gray[y:y + h, x:x + w])` – запис зображення до файлу, заданого першим параметром. Другий параметр – це власне зображення, яке буде зберігатись. В даному випадку – це прямокутник, який “вирізається” із зображення у відтінках сірого.
- `cv2.destroyAllWindows()` – закриває вікно із відео.

Ще один метод класу `PhotoCollector` має назву `dir_creation()` і призначений для створення нової папки в каталозі `Dataset`, де будуть зберігатись зроблені фото. В методі використовуються функції з модуля `OS`.

Файл `face_training.py` містить клас `Trainer`, який відповідає за навчання класифікатора. У класі міститься конструктор і два методи.

Метод `get_images_and_labels()` знаходить обличчя на зображеннях із зібраного датасету, а також присвоює кожному обличчю `id`. В методі

використовувався функціонал бібліотеки Pillow і NumPy для обробки зображення у вигляді масиву значень інтенсивностей.

Метод `train_recognizer()` навчає класифікатор і використовує такі функції OpenCV:

- `cv2.face.LBPHFaceRecognizer_create()` – створює розпізнавач на основі алгоритму LBPH.
- `lbph_recognizer.train(faces, np.array(ids))` – навчає класифікатор на зібраному датасеті. Першим параметром функція приймає зображення облич, другий параметр – це асоційовані із зображеннями ідентифікатори у вигляді масиву numpy.
- `lbph_recognizer.write('Trainer/trainer.yml')` – зберігає розпізнавач і стан його моделі. В якості параметру функція приймає ім'я файлу у форматі xml або yaml, в який буде збережено результат навчання.

У файлі `face_recognition.py` міститься клас `Recognizer`, призначений для розпізнавання облич. Клас має в своєму складі конструктор і один метод.

Конструктор ініціалізує каскадний класифікатор, який потім буде використовуватись для пошуку лиця на відео, і розпізнавач на основі алгоритму LBPH, який буде використовуватись для розпізнавання обличчя, знайденого за допомогою каскадів Хаара. Розпізнавач за допомогою функції `lbph_recognizer.read('Trainer/trainer.yml')` зчитує файл з навченою моделлю класифікатора.

Метод `face_recognizer()` перетворює кожен кадр із відеопотоку у зображення в градаціях сірого і шукає на цьому зображенні обличчя. Функції бібліотеки OpenCV, використані в методі:

- `cv2.rectangle()` малює прямокутник навколо регіону, де було знайдено обличчя (рис 3.2). Перший параметр функції – це зображення, на якому буде намальований прямокутник. У змінних `x` та `y` містяться значення координат верхнього лівого кута прямокутника, що обрамлює обличчя, `w` – це значення ширини цього прямокутника, а `h` – висоти. Тому другий і третій параметри функції `cv2.rectangle()` це ліва верхня вершина

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		42

прямокутника і нижня права відповідно. Передостанній параметр вказує на колір ліній прямокутника, останній параметр визначає їх товщину.

- `lbph_recognizer.predict()` повертає ідентифікатор особи, яка може бути зображена на фото, і значення “довіри” до результату. Чим нижча “довіра” тим точніший результат. В якості параметра функція приймає прямокутний фрагмент зображення, який містить обличчя.
- `cv2.putText()` пише текст на зображенні. В даному випадку пише ім’я людини, чие обличчя було знайдене на зображенні. Приймає такі параметри: зображення, на якому потрібно написати текст, текст надпису, координати, де буде починатись надпис, шрифт, масштаб шрифту, товщина шрифту.

```
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 255, 0), 2)
    id_, confidence = self.lbph_recognizer.predict(gray[y:y + h, x:x + w])
```

Рисунок 3.2 – Фрагмент коду методу `face_recognizer()`

3.3 Тестування системи

На заваді точного розпізнавання обличч стоять такі причини як погане освітлення, окуляри на обличчі, зміна зачіски, присутність на обличчі емоцій та ін. Також, зазвичай, моделі розпізнавачів навчають на зображеннях людей європеоїдної раси, тому деякі системи розпізнавання погано справляються із знаходженням і визначенням обличч людей монголоїдної або негроїдної раси.

Для початку буде протестовано здатність розробленої системи знаходити обличчя, а потім здатність розпізнавати.

Як видно з рис. 3.3 система чудово справляється із виявленням обличчя в анфас без окулярів, бороди та емоцій.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		43

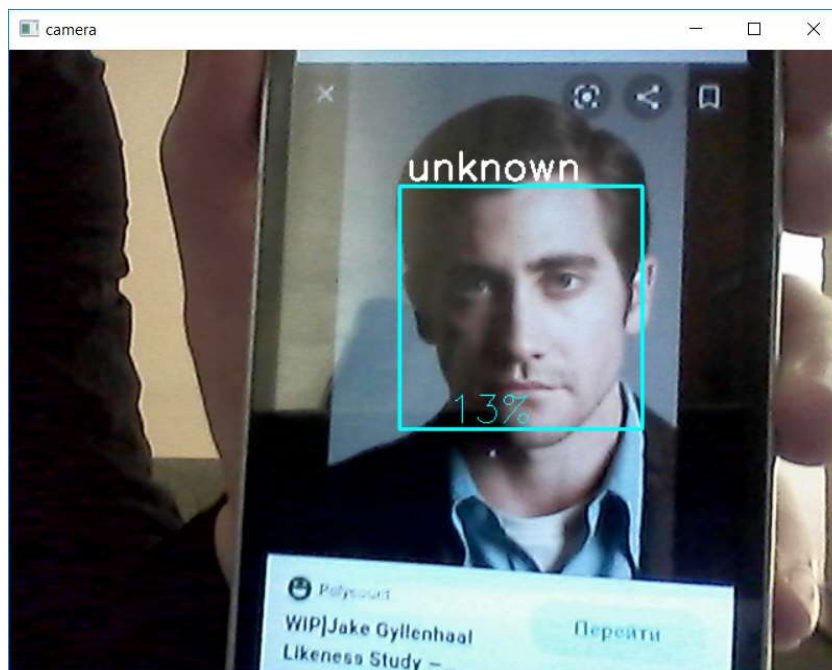


Рисунок 3.3 – Скріншот зображення з камери

Система також знаходить обличчя, якщо на ньому є окуляри для зору (рис. 3.4) та окуляри від сонця (рис 3.5). З цього можна зробити висновок, що стандартний класифікатор із OpenCV навчали на датасеті, який включав зображення обличчя в окулярах.

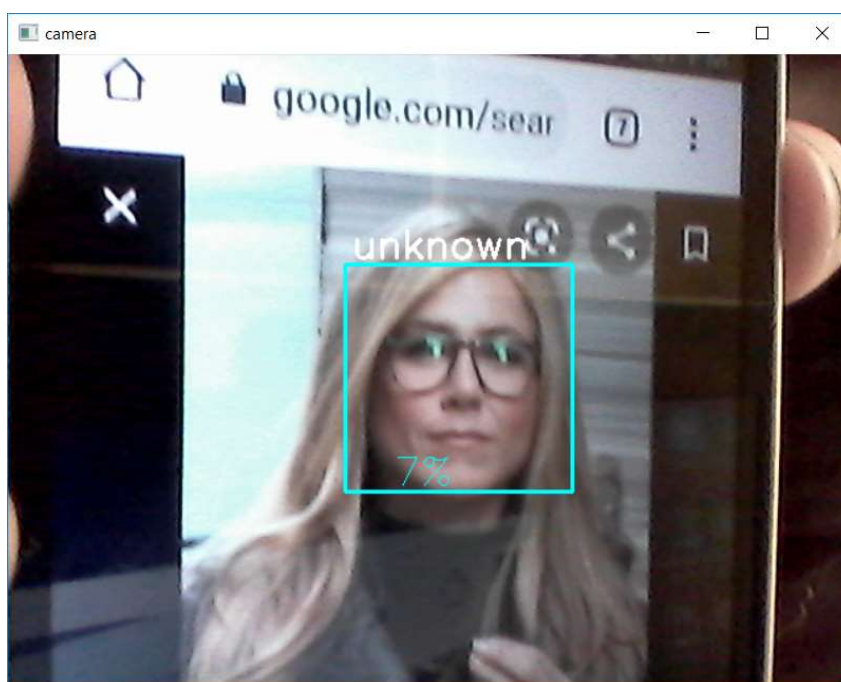


Рисунок 3.4 – Скріншот зображення з камери

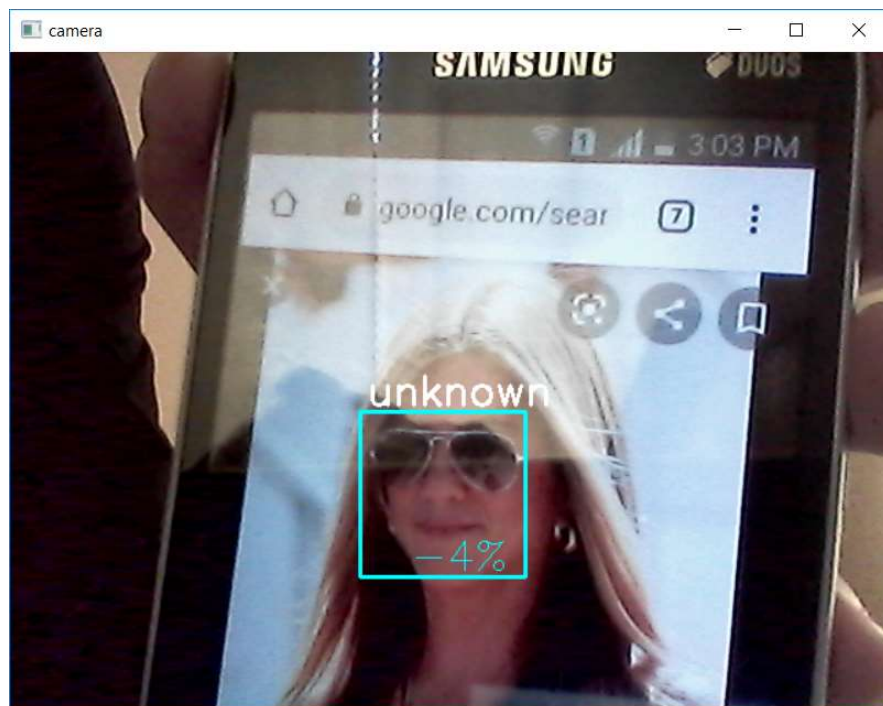


Рисунок 3.5 – Скріншот зображення з камери

Як видно із рис. 3.6 система знаходить обличчя з бородою і вусами. Якщо на обличчі є і окуляри від сонця і борода, як на рис. 3.7, то система без проблем виявляє обличчя з ними.

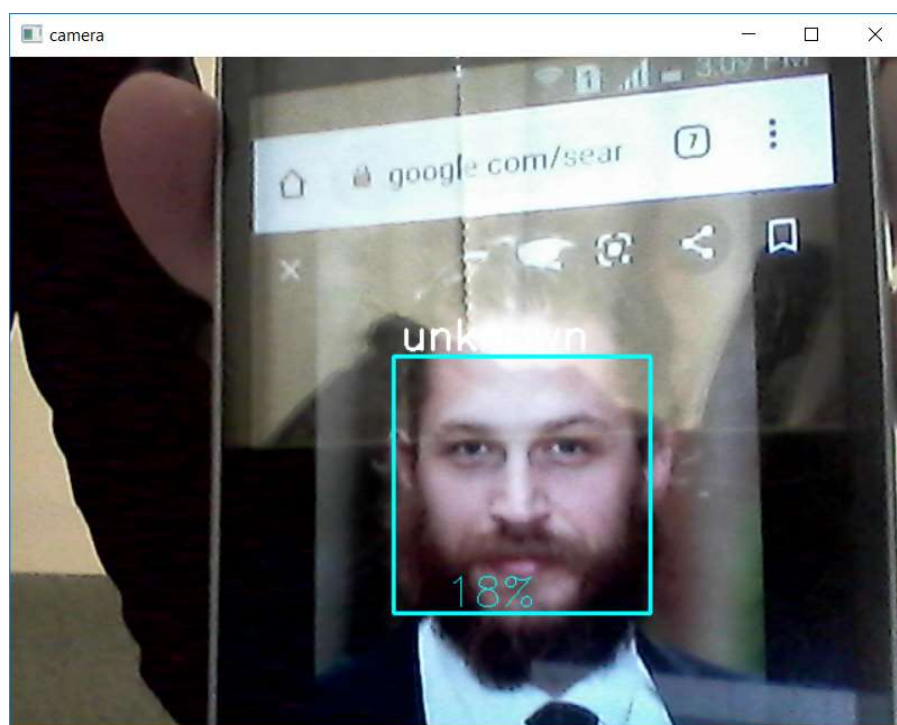


Рисунок 3.6 – Скріншот зображення з камери

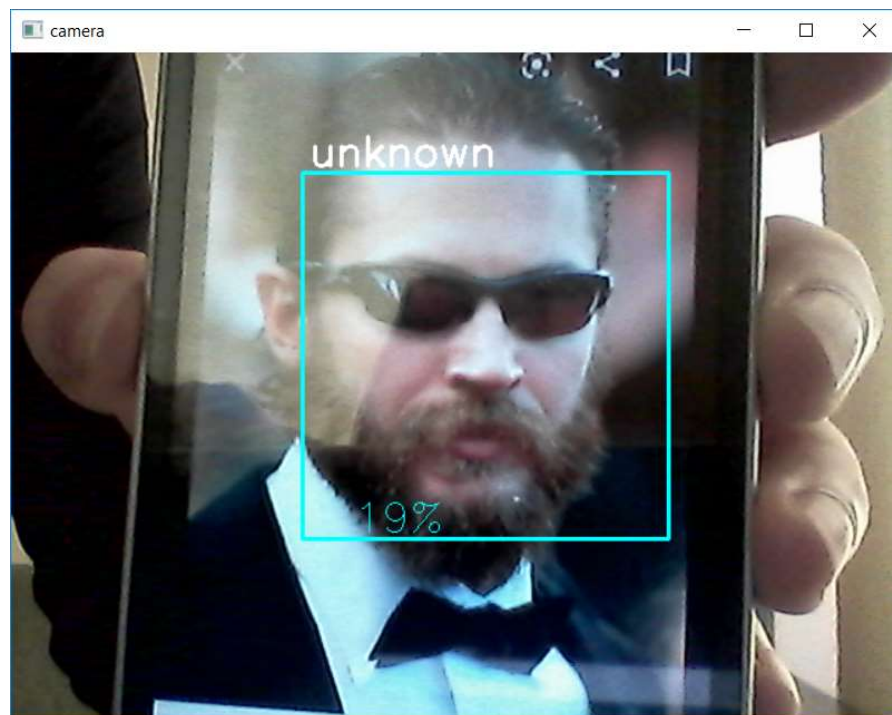


Рисунок 3.7– Скріншот зображення з камери

Обличчя з усмішкою система також знаходить (рис. 3.8). Не складає великих труднощів і знаходження обличчя чорношкірого чоловіка на рис. 3.9.

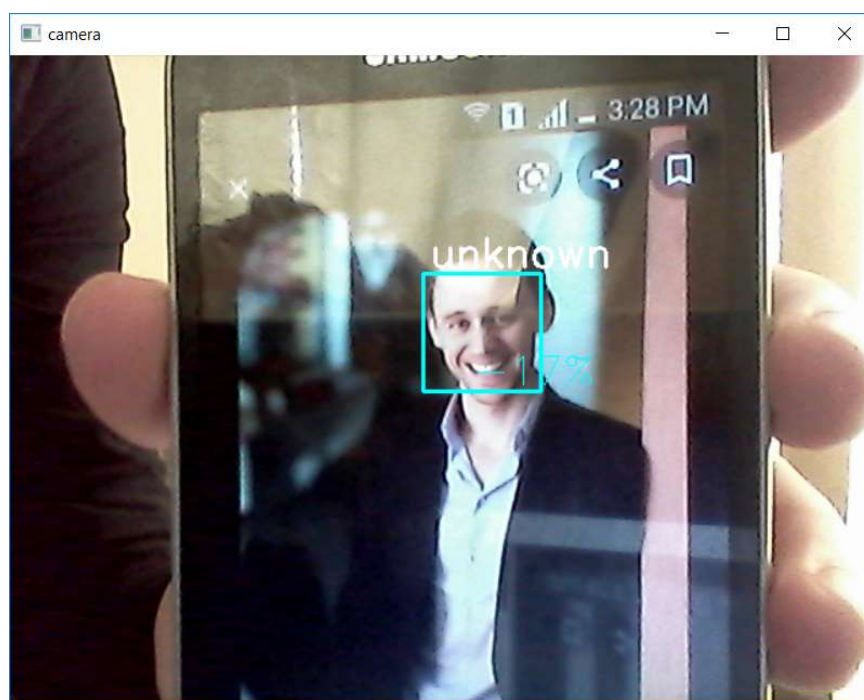


Рисунок 3.8– Скріншот зображення з камери

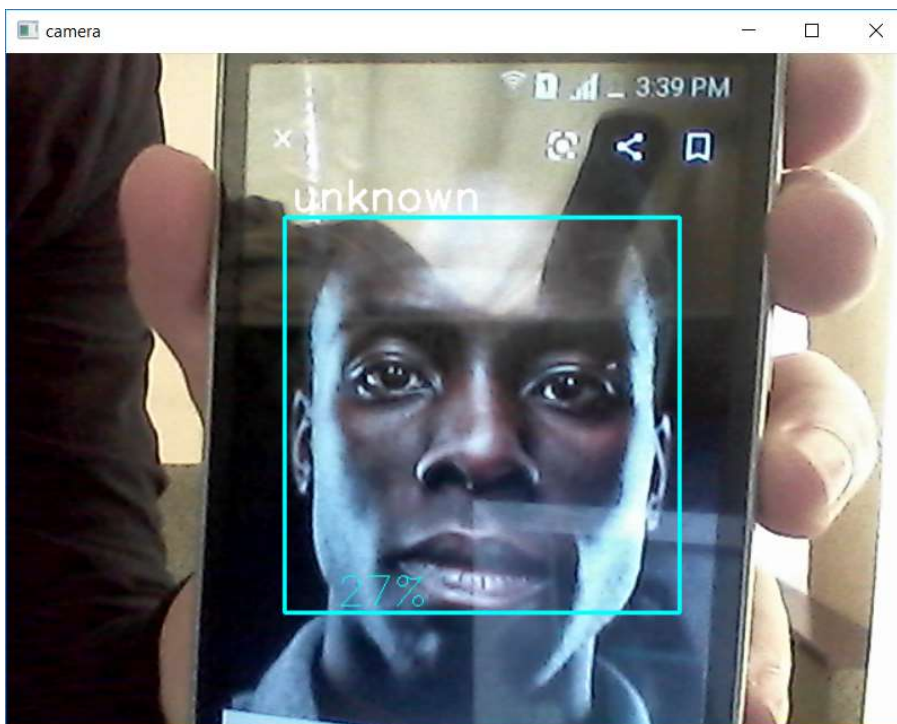


Рисунок 3.9– Скріншот зображення з камери

Обличчя чорношкірої дівчини з пишною зачіскою на рис 3.10 система теж виявила. Варто зазначити, що розміри облич на фото відрізняється, і все-таки це не заважає системі їх знаходити.

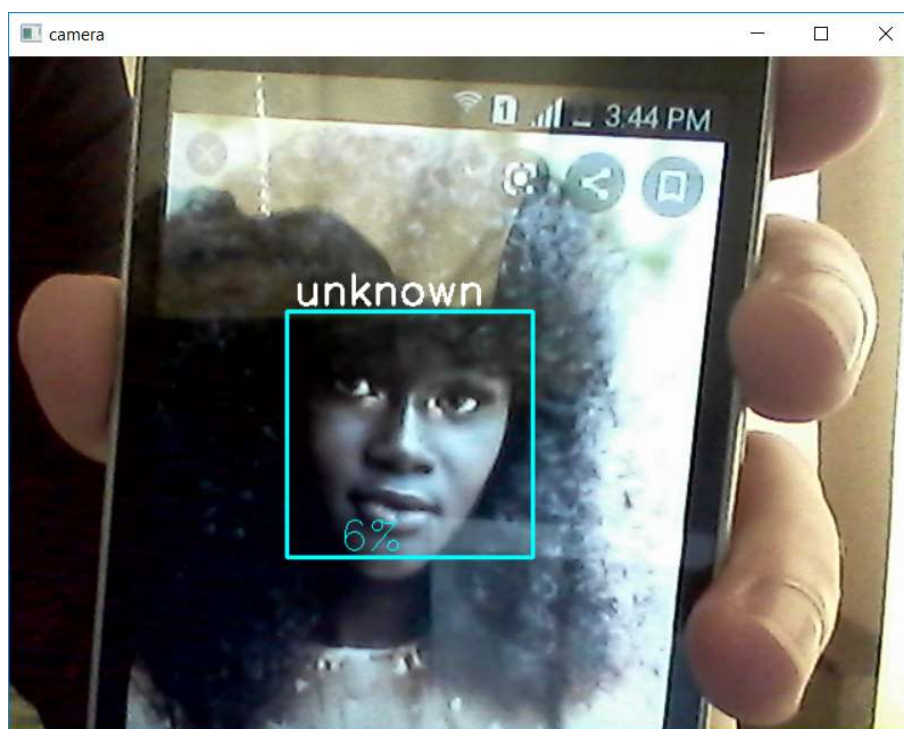


Рисунок 3.10– Скріншот зображення з камери

Далі класифікатор системи було навчено на зображеннях 4 людей. Для кожного датасет містив як мінімум 20 фото. Всі фото були знайдені в мережі Інтернет. У більшості випадків система розпізнавала обличчя правильно (рис. 3.11, рис. 3.12, рис. 3.13, рис. 3.14).

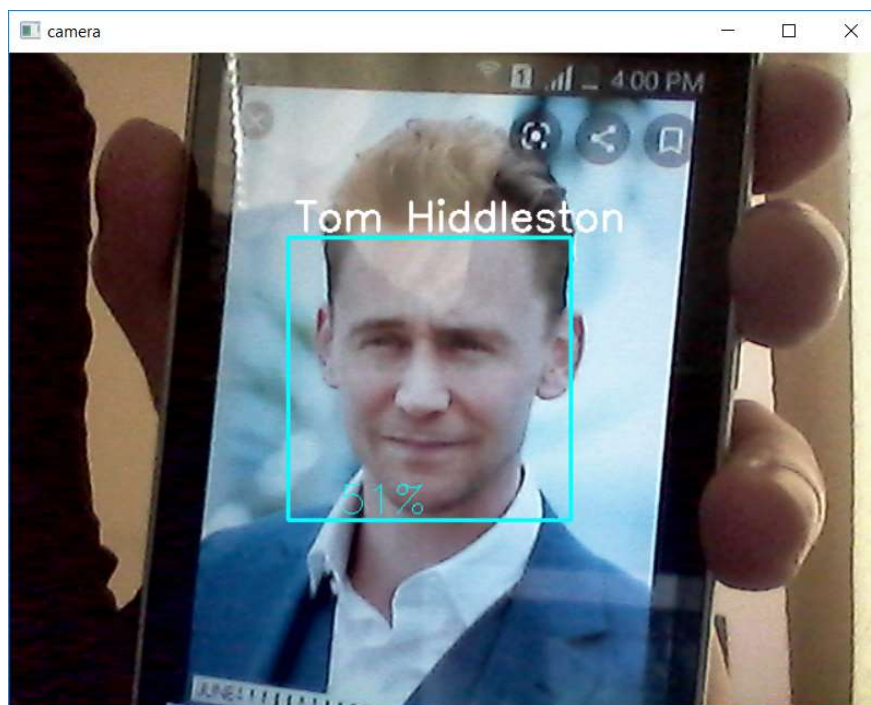


Рисунок 3.11– Скріншот зображення з камери

Система правильно визначила, що на фото зображений Том Хідлстон (рис. 3.11) і Наталі Портман (рис.3.12).

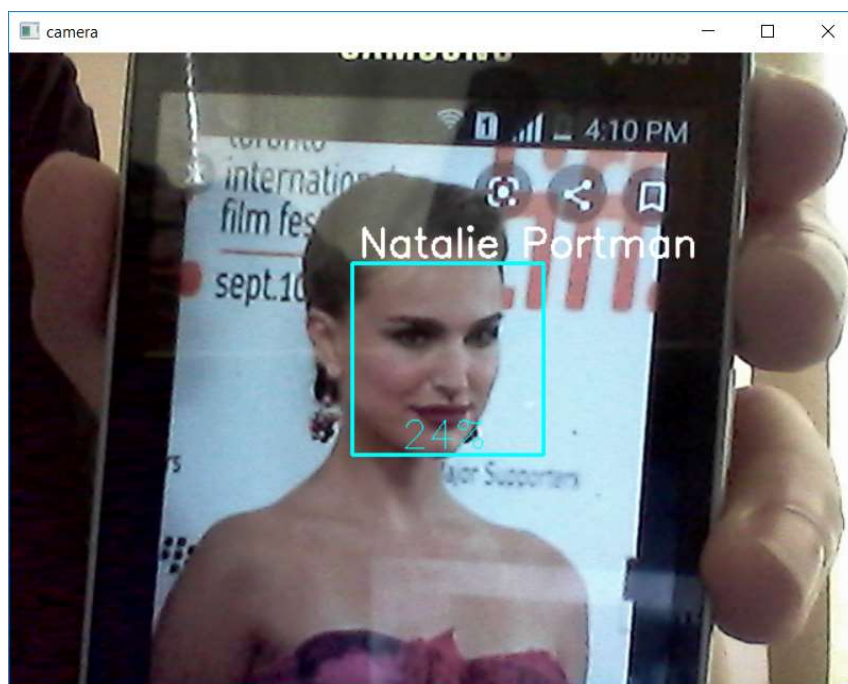


Рисунок 3.12– Скріншот зображення з камери

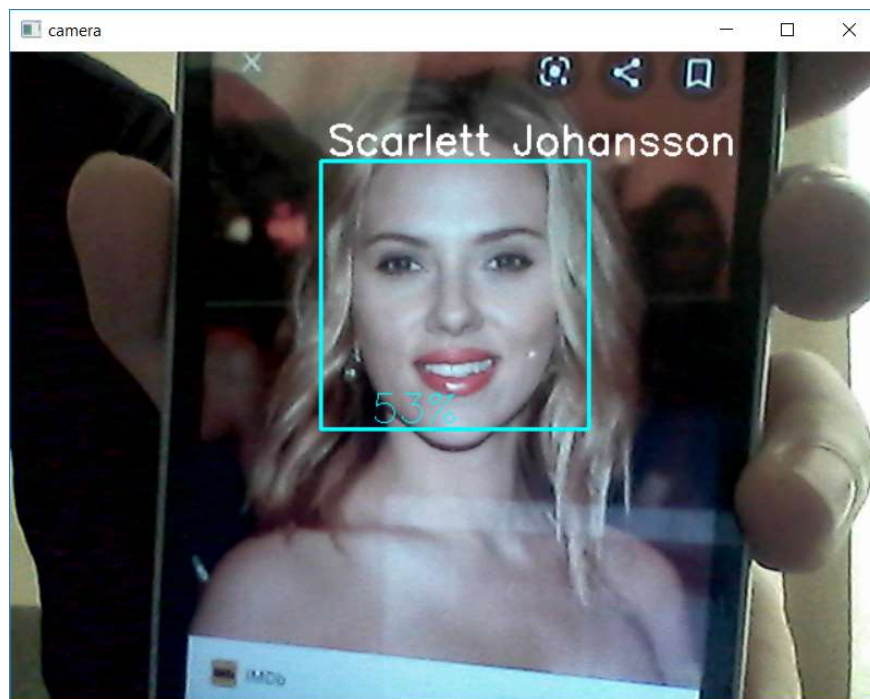


Рисунок 3.13– Скріншот зображення з камери

На рис. 3.13 і рис. 3.14 наведено скріншоти кадрів з камери, на яких система вірно визначила імена людей на фото. Варто звернути увагу на те, що незалежно від того де знаходиться обличчя в кадрі, система знаходить і розпізнає його.

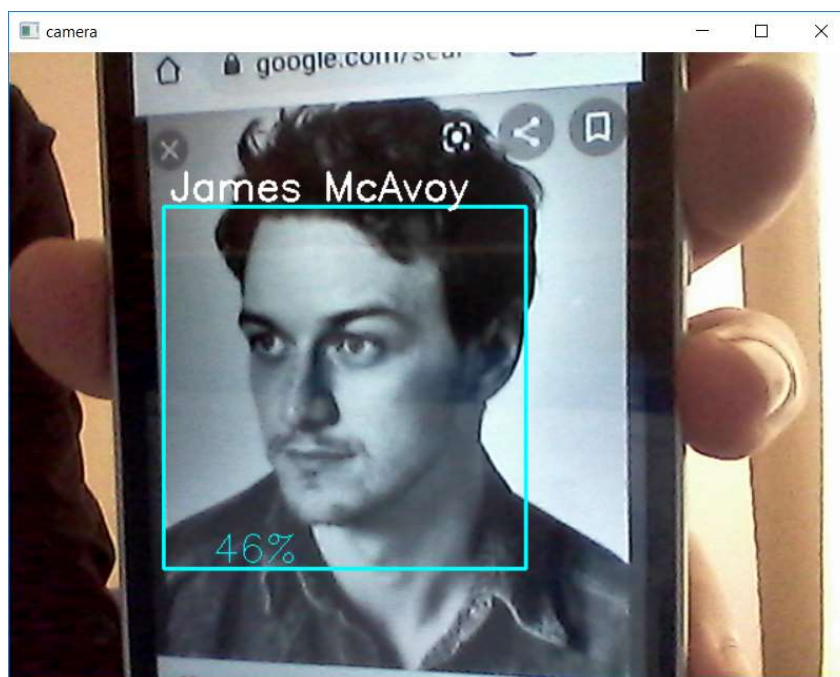


Рисунок 3.14– Скріншот зображення з камери

Проте деколи система хибно визначала ім'я людини, чиє обличчя було зображено на світлині, як от на рис. 3.15 і рис. 3.16. Найчастіше вона хибно припускала, що обличчя належить Тому Хідлстону.

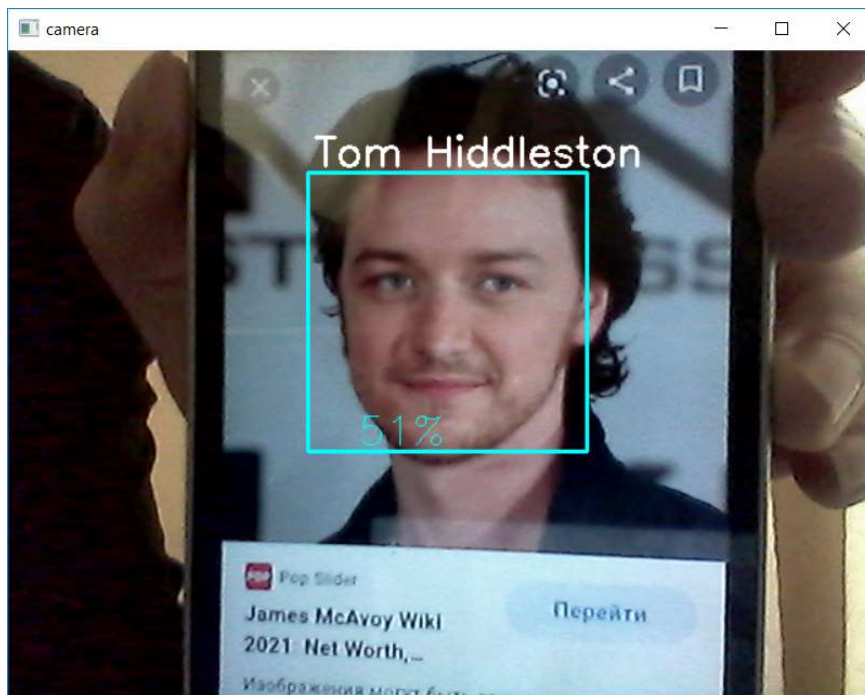


Рисунок 3.15– Скріншот зображення з камери

Причиною хибного розпізнавання може бути порівняно невелика кількість різних фото в датасеті для однієї людини.

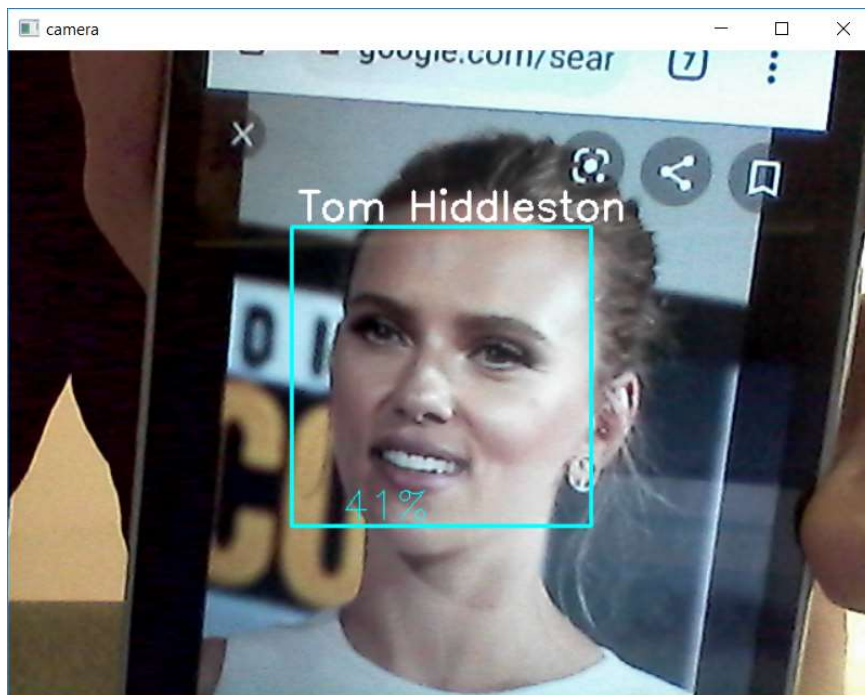


Рисунок 3.16– Скріншот зображення з камери

ВИСНОВОК ДО РОЗДІЛУ 3

У даному розлілі було описано засоби, які використовувались під час розробки системи, файли проекту і функції, які в них використовувались, а також тестування працездатності системи.

Тестування було поділено на два етапи. Спочатку перевірялась здатність системи знаходити обличчя на світлині. На тестових фото були зображені обличчя в окулярах для зору і від сонця, з бородою і вусами і алгоритм чудово справився із виявленням таких облич. Також було протестовано обличчя з емоціями, та обличчя з темним кольором шкіри. І знову ж таки, система правильно вказала місцезнаходження обличчя на зображенні. Варто зауважити, що точність алгоритму є досить високою. Ймовірність хибного знаходження облич, тобто виявлення лиця там, де його насправді немає, практично дорівнює нулю. Вищеназвана характеристика є однією з причин популярності методу Віюлі-Джонса і його каскадів Хаара. Під час тестування програми вона підтвердилась на практиці.

Другим етапом тестування було розпізнавання облич. Система показала хороші результати, проте були випадки помилкового розпізнавання. Причиною цього може бути мала кількість фото на одну людину (20-30 штук), недостатньо велика роздільна здатність веб-камери, нерівномірне освітлення. Також варто зауважити, що для виявлення і розпізнавання використовувались зображення з мобільного телефону, а не реальне людське обличчя. Наявність бліків від екрану на відео могла погіршити точність розпізнавання.

Загалом можна сказати, що система показала доволі непогані результати, особливо під час знаходження облич на світлинах. Покращити точність розпізнавання можна збільшенням кількості фото в датасеті на одну людину і використанням кращої камери.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		51

ВИСНОВКИ

Даний дипломний проєкт присвячено розробці системи розпізнавання облич в режимі реального часу.

У першому розділі було розглянуто етапи розвитку систем розпізнавання облич, характеристики та специфіку роботи існуючих реалізацій систем розпізнавання. Проведено аналіз методів, які компанії застосовують у своїх продуктах, та визначено напрямки у яких зараз рухається індустрія комп'ютерного зору та розпізнавання облич.

В другому розділі було описано наявні методи для розпізнавання лиць. Описувались алгоритми традиційного комп'ютерного зору і більш сучасний підхід з використанням глибокого навчання. Було виділено слабкі і сильні сторони кожного алгоритму і вибрано найбільш оптимальні по часу реалізації, точності, обчислювальних затратах і кількості зображень для навчання.

Третій розділ присвячено опису розробленої програми і її тестуванню. Останнє було виконано з урахуванням ситуацій, в яких система розпізнавання облич може стикнутись із труднощами. Прикладом таких ситуацій є окуляри на обличчі, борода, посмішка, темний колір шкіри і т. п. Також було проведено навчання класифікатора і перевірено точність роботи системи.

Результатом виконання дипломного проєкту є система розпізнавання облич, яка може знаходити і розпізнавати обличчя незалежно від його розміру і положення в кадрі. Точність виявлення обличчя системою є досить високою, тому її можна використовувати, наприклад, для трекінгу облич на відео.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		52

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Woodrow Bledsoe Originates of Automated Facial Recognition [Електронний ресурс] – Режим доступу до ресурсу: <https://www.historyofinformation.com/detail.php?id=2126>.
2. Sirovich L. Low-dimensional procedure for the characterization of human faces [Електронний ресурс] / L. Sirovich, M. Kirby // Journal of the Optical Society of America A. – 1987. – Режим доступу до ресурсу: <https://www.face-rec.org/interesting-papers/General/ld.pdf>.
3. The FERET evaluation methodology for face-recognition algorithms / P. Phillips, M. Hyeonjoon, S. Rizvi, P. Rauss. // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2000. – №22. – С. 1090–1104.
4. Crawford M. Facial recognition progress report [Електронний ресурс] / Mark Crawford // SPIE. – 2011. – Режим доступу до ресурсу: <https://spie.org/news/facial-recognition>.
5. Chollet F. Deep learning with Python / Francois Chollet., 2017. – 384 с.
6. Facebook settles facial recognition dispute [Електронний ресурс] // BBC News. – 2020. – Режим доступу до ресурсу: <https://www.bbc.com/news/technology-51309186>.
7. Ng A. How China uses facial recognition to control human behavior [Електронний ресурс] / Alfred Ng // CNET. – 2020. – Режим доступу до ресурсу: <https://www.cnet.com/news/in-china-facial-recognition-public-shaming-and-control-go-hand-in-hand/>.
8. What is Amazon Rekognition? [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html>.
9. Hardware security overview [Електронний ресурс] – Режим доступу до ресурсу: <https://support.apple.com/uk-ua/guide/security/secf020d1074/1/web/1>.
10. Apple explains how iPhone X facial recognition with Face ID works (and fails) in security paper [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://9to5mac.com/2017/09/27/face-id-iphone-x-white-paper/>.

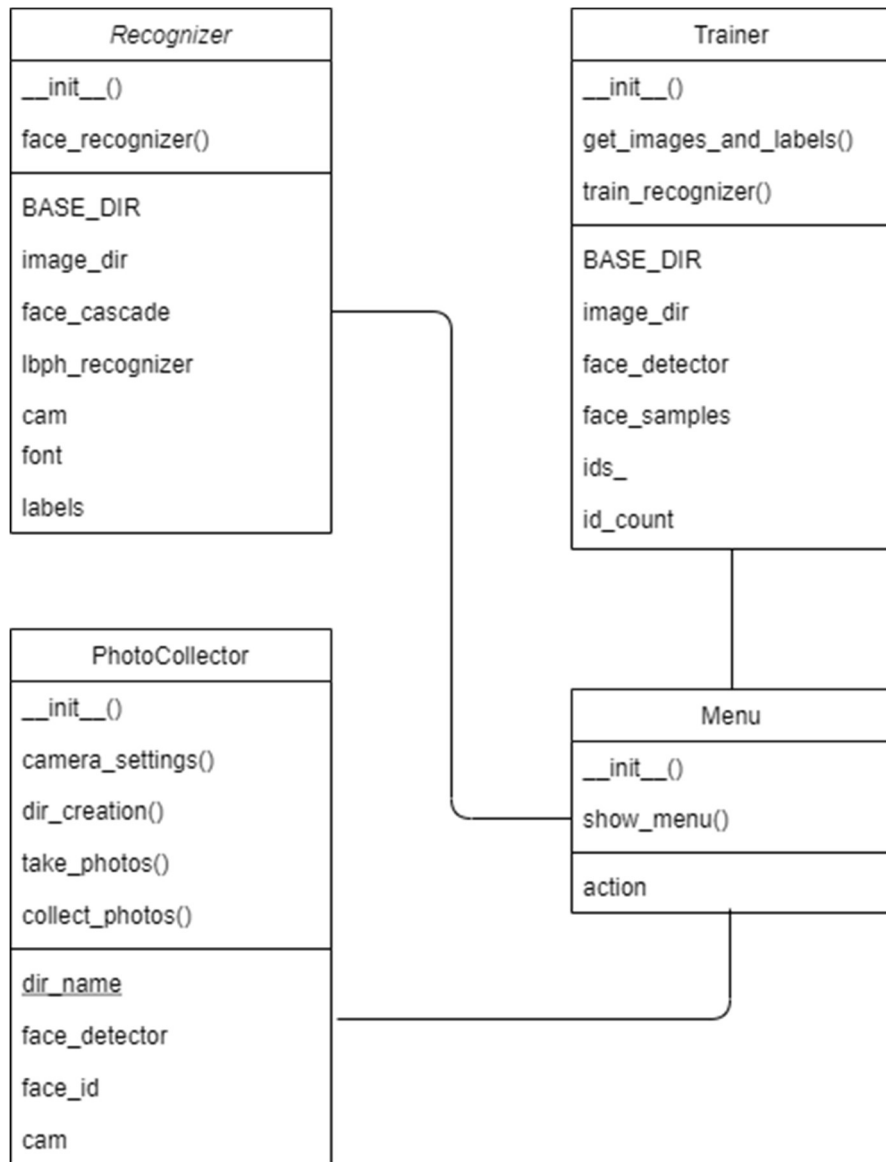
					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		53

11. DeepFace: Closing the Gap to Human-Level Performance in Face Verification [Електронний ресурс] / Y. Taigman, M. Yang, M. Ranzato, L. Wolf // Facebook Research. – 2014. – Режим доступу до ресурсу: <https://research.fb.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/>.
12. Що таке розпізнавання обличчя у Facebook і як працює ця функція? [Електронний ресурс] – Режим доступу до ресурсу: https://www.facebook.com/help/122175507864081?helpref=faq_content.
13. Turk M. Eigenfaces for Recognition / M. Turk, A. Pentland. // Journal of Cognitive Neuroscience. – 1991. – №3. – С. 71–86.
14. Labeled Faces in the Wild [Електронний ресурс] – Режим доступу до ресурсу: <http://vis-www.cs.umass.edu/lfw/index.html>.
15. ML | Face Recognition Using Eigenfaces (PCA Algorithm) [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/>.
16. Belhumeur P. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection / P. Belhumeur, J. Hespanha, D. Kriegman. // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1997. – №19. – С. 711–720.
17. Wagner P. Fisherfaces [Електронний ресурс] / Philipp Wagner. – 2012. – Режим доступу до ресурсу: <https://www.bytefish.de/blog/fisherfaces.html>.
18. Salton do Prado K. Face Recognition: Understanding LBPH Algorithm [Електронний ресурс] / Kelvin Salton do Prado. – 2017. – Режим доступу до ресурсу: <https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>.
19. Viola P. Rapid object detection using a boosted cascade of simple features / P. Viola, M. Jones. // Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. – 2001.

- 20.Метод Виолы-Джонса (Viola-Jones) как основа для распознавания лиц [Электронный ресурс]. – 2011. – Режим доступа до ресурсу: <https://habr.com/ru/post/133826/>.
- 21.Breaking Down Facial Recognition: The Viola-Jones Algorithm [Электронный ресурс]. – 2019. – Режим доступа до ресурсу: <https://towardsdatascience.com/the-intuition-behind-facial-detection-the-viola-jones-algorithm-29d9106b6999>.
- 22.Gradient-based learning applied to document recognition / Y.LeCun, L. Bottou, Y. Bengio, P. Haffner. // Proceedings of the IEEE. – 1998. – №86. – С. 2278–2324.
- 23.An Intuitive Explanation of Convolutional Neural Networks [Электронный ресурс]. – 2016. – Режим доступа до ресурсу: <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>.
- 24.CS231n Convolutional Neural Networks for Visual Recognition [Электронный ресурс] – Режим доступа до ресурсу: <https://cs231n.github.io/convolutional-networks/#fc>.

Додаток А
ФУНКЦІОНАЛЬНА СХЕМА
до дипломного проєкту
на тему: «Система розпізнавання обличч»

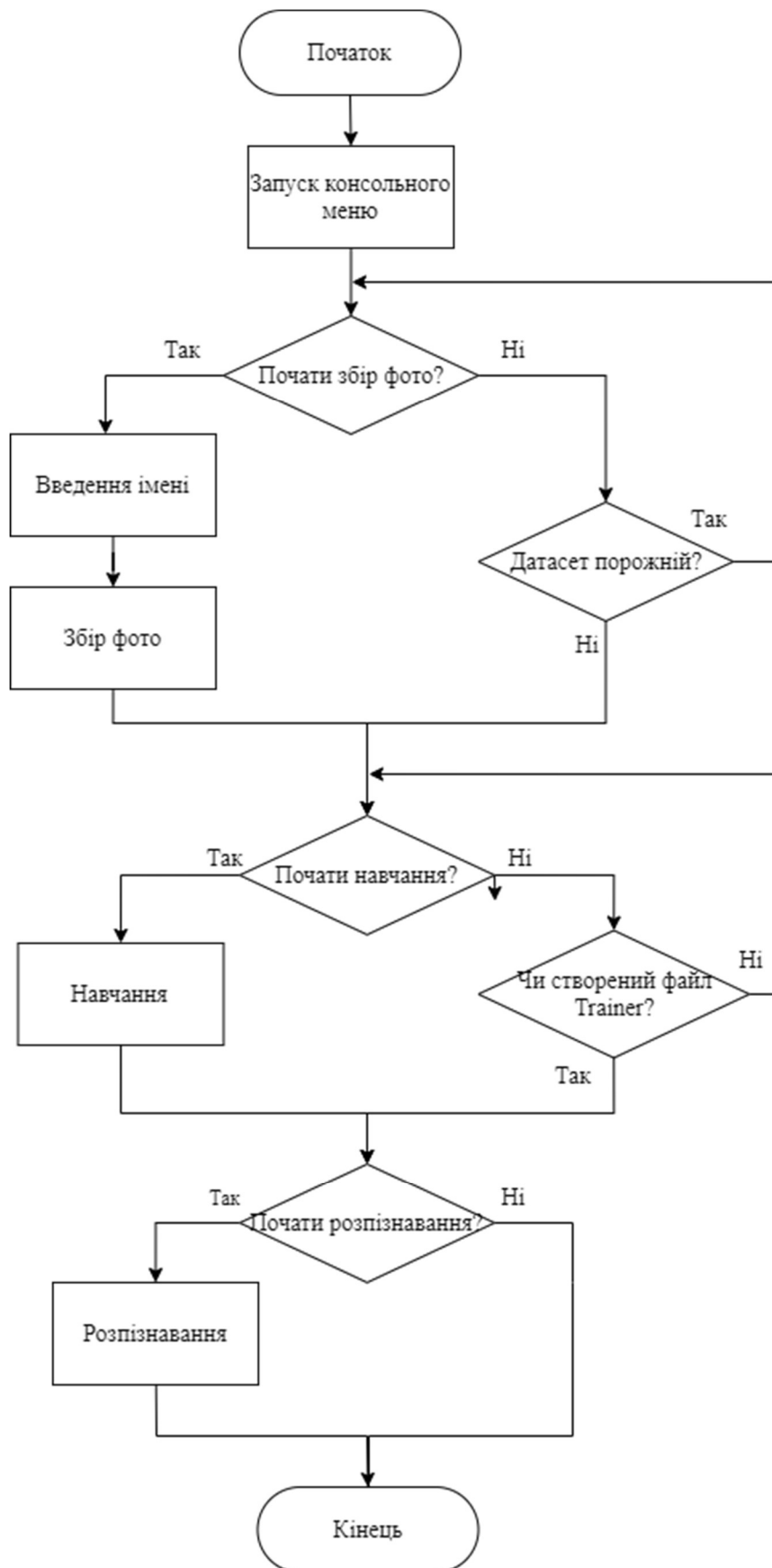
Київ – 2021 року



					ІАЛЦ.467800.004 ДІ						
Зм.	Арк.	№ докум.	Підпис	Дата	Система розпізнавання облич	Лім.		Арк.	Аркуші		
Розробив		Приймич А.-В.О.						1	1		
Перевірів		Каплунов А. В.									
Реценз.											
Н. Контр.		Сімоненко В.П.									
Завв.		Стіренко С.Г.			Функціональна схема	«КПІ імені Ігоря.Сікорського» ФІОТ, гр. ІО-71					

Додаток Б
ПРИНЦИПОВА СХЕМА
до дипломного проєкту
на тему: «Система розпізнавання облич»

Київ – 2021 року



					ІАЛЦ.467800.005 Д2		
Зм.	Арк.	№ докум.	Підпис	Дата	Система розпізнавання облич Принципова схема		
Розробив	Приймач	А.-В.О.					
Перевірів	Каплунов	А. В.					
Реценз.							
Н. Контр.	Сімоненко	В.П.					
Затв.	Стіренко	С.Г.			Літ.		
					Арк.	Аркуші	
					1	1	
					«КПІ імені Ігоря Сікорського» ФІОТ, гр. ІО-71		

Додаток В
СТРУКТУРНА СХЕМА
до дипломного проєкту
на тему: «Система розпізнавання облич»

Київ – 2021 року



					ІАЛЦ.467800.006 ДЗ					
Зм.	Арк.	№ докум.	Підпис	Дата	Система розпізнавання облич			Літ.	Арк.	Аркуші
Розробив		Приймич А.-В.О.								
Перевірів		Каплунов А. В.							1	1
Реценз.								«КПІ імені Ігоря Сікорського» ФІОТ, гр. ІО-71		
Н. Контр.		Сімоненко В.П.								
Затв.		Стіренко С.Г.			Структурна схема					