


Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

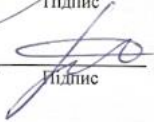
КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

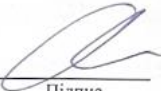
на тему Метод розпізнавання обличчя за зображенням під довільним кутом зору

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 2 курсу, група КНм-22-1  В.О.Добржанський
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: д.т.н., професор, зав. кафедри КН  О.В. Бармак
Науковий ступінь, посада Підпис Ініціали, прізвище


Нормоконтроль: к.т.н., доцент кафедри КН  Р.О. Багрій
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор  О.В. Бармак
Підпис Ініціали, прізвище

10 грудня 2023 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет інформаційних технологій
Кафедра комп'ютерних наук
Освітній ступінь магістр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук



(підпис)
д.т.н., професор О.В. Бармак
« 01 » вересня 2023 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

1. Тема кваліфікаційної роботи магістра: «Метод розпізнавання обличчя за зображенням під довільним кутом зору»
2. Завдання видано студенту Добржанському Володимирі Олеговичу
(прізвище, ім'я, по батькові)
3. Керівник роботи Завідувач кафедри КН Бармак Олександр Володимирович
(прізвище, ім'я, по батькові)

4. Затверджені наказом університету від « 15 » серпня 2023 р. № 30

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета кваліфікаційної роботи магістра – вдосконалення методу розпізнавання обличчя за зображенням нейромережевими засобами. Для досягнення поставленої мети визначено наступні задачі: проаналізувати існуючі способи розпізнавання обличчя за зображенням під довільним кутом зору; здійснити аналіз згорткових нейронних мереж та їх архітектур для вибору найбільш відповідних для поставленого завдання; вдосконалити метод розпізнавання обличчя за зображенням під довільним кутом зору та покращити його для умов недостатньої або надмірної освітленості; підготувати набори даних для донавчання згорткової нейронної мережі; провести валідацію та аналіз отриманих експериментальних результатів запропонованого методу розпізнавання обличчя за зображенням за стандартними показниками якості.

Реферат

Кваліфікаційна робота магістра присвячена розробці методу розпізнавання обличчя за зображенням під довільним кутом зору.

Актуальність теми. Розпізнавання обличчя за зображенням є важливою частиною сучасного життя, особливо у сфері безпеки, як загальної (камери спостереження) так і персональної (особисті файли на пристрої). Більшість сучасних смартфонів використовують фронтальну камеру для ідентифікації особи, але не завжди цей процес є успішним, через такі фактори як: якість отриманого зображення, недостатня попередня навченість моделі, кут огляду, рівень освітлення. Деякі підприємства встановлюють автоматичні системи спостереження на своїх об'єктах, які сигналізують, якщо вони зуміли зафіксувати особу, яка не має певного рівня доступу до приміщення, але нажаль через ті самі фактори, дані системи не можуть працювати на 100 % ефективно.

Метод розпізнавання обличчя за зображенням під довільним кутом зору є актуальною темою дослідження. Згорткові нейронні мережі є дуже ефективними для задач аналізу зображень та класифікації об'єктів на них.

Метод розпізнавання обличчя за зображенням під довільним кутом зору полягає у використанні алгоритмів машинного навчання, для розпізнавання обличчя за багатьма ознаками (формою носа, сітківки ока, відстані між очима, формою вух) та інших навіть не помітних людському оку, до прикладу як текстура шкіри. Перевагою удосконаленого методу є те, що він дозволяє зменшити похибку при розпізнаванні обличчя за зображенням, коли обличчя не повністю присутнє на фото та якщо рівень освітленості зависокий або занижкий. Перевага методу полягає тому, що спеціально створеним датасетом, орієнтованим як на зображення під різними кутами зору так і на різні рівні освітленості можна донавчити попередньо навчену нейромережу.

Отже, метод розпізнавання обличчя за зображенням під довільним кутом зору є одним із перспективних напрямів для дослідження, який може сприяти покращенню існуючих засобів безпеки заснованих на розпізнаванні обличчя за зображенням.

Також, даний метод можна використати, для ідентифікації людей на відео, адже будь який відеозапис – це послідовність зображень.

Мета і задачі роботи. Метою кваліфікаційної роботи магістра є вдосконалення методу розпізнавання обличчя за зображенням під довільним кутом зору.

Для досягнення поставленої мети визначено наступні задачі:

- проаналізувати існуючі способи розпізнавання обличчя за зображенням під довільним кутом зору;
- здійснити аналіз згорткових нейронних мереж та їх архітектур для вибору найбільш відповідних для поставленого завдання;
- вдосконалити метод розпізнавання обличчя за зображенням під довільним кутом зору та покращити його для умов недостатньої або надмірної освітленості;
- підготувати набори даних для донавчання згорткової нейронної мережі;
- провести валідацію та аналіз отриманих експериментальних результатів запропонованого методу розпізнавання обличчя за зображенням за стандартними показниками якості.

Апробація результатів кваліфікаційної роботи магістра та публікації.

Результати роботи опубліковані у науковому журналі – Метод розпізнавання обличчя під довільним кутом зору / Добржанський В.О., Бармак О.В., Скрипник Т.К.// Науковий журнал «Вісник Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2023. №6 (довідка від редакції).

Структура та обсяг роботи. Кваліфікаційна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 51 найменування та 3 додатків. Загальний обсяг дипломної роботи магістра становить 112 сторінок, з них 84 сторінок основного тексту та 28 сторінок додатків. У роботі наведено 49 рисунків, 8 формул та 13 таблиць.

Ключові слова: розпізнавання обличчя, кут зору, глибокі нейронні мережі, нейромережі, згорткова нейронна мережа, CNN, набір даних.

Зміст

Перелік скорочень	4
Вступ.....	5
РОЗДІЛ 1	8
Аналіз сучасного стану використання інформаційних технологій для розпізнавання обличчя за зображенням під довільним кутом зору	8
1.1 Аналіз предметної області	8
1.2 Методи глибокого навчання для розпізнавання обличчя за зображенням під довільним кутом зору	13
1.3 Аналіз існуючих публікацій розпізнавання обличчя за зображенням під довільним кутом зору	15
1.4 Постановка задачі.....	20
Висновки до розділу 1	22
РОЗДІЛ 2	23
Метод виявлення обличчя за зображенням під довільним кутом зору.....	23
2.1 Моделі глибокого навчання для розпізнавання обличчя.....	24
2.2 Підготовка вхідних даних	31
2.3 Навчання та донавчання моделі	35
2.4 Оцінка якості навченої нейромережі	40
Висновки до розділу 2	43
РОЗДІЛ 3	44
Проектування прототипної інформаційної системи для реалізації вдосконаленого методу розпізнавання обличчя за зображенням під довільним кутом зору	44
3.1 Вибір платформи, технологій та бібліотек.....	44
3.2 Проектування прикладних компонентів застосування.....	48

Висновки до розділу 3	50
РОЗДІЛ 4	51
Дослідження методу розпізнавання обличчя за зображенням під довільним кутом зору	51
4.1 Прототипне застосування для дослідження запропонованого вдосконаленого методу розпізнавання обличчя за довільним кутом зору	51
4.2 Підготовка вхідних даних	55
4.3 Результати експериментального дослідження вдосконаленого методу	60
Висновки до розділу 4	76
Загальні висновки.....	77
Перелік посилань.....	78
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
CPU	Central processing unit
GPU	Graphics processing unit
CNN	Convolutional neural network
FLOPs	Floating-point operations per second
RNN	Recurrent neural networks
IOT	Internet of things
ReLU	Rectified linear unit
SGD	Stochastic gradient descent
ROC	Receiver operating characteristic
IDE	Integrated development environment
RPN	Region Proposal Network
SSA	Singular spectrum analysis
FAF	Facial attribute features
FRF	Face recognition features
ГНМ	Глибока нейронна мережа
ІТ	Інформаційні технології
ШНМ	Штучна нейронна мережа

Вступ

Актуальність теми. Розпізнавання обличчя за зображенням є важливою частиною сучасного життя, особливо у сфері безпеки, як загальної (камери спостереження) так і персональної (особисті файли на пристрої). Більшість сучасних смартфонів використовують фронтальну камеру для ідентифікації особи, але не завжди цей процес є успішним, через такі фактори як : якість отриманого зображення, недостатня попередня навченість моделі, кут огляду, рівень освітлення. Деякі підприємства встановлюють автоматичні системи спостереження на своїх об'єктах, які сигналізують, якщо вони зуміли зафіксувати особу, яка не має певного рівня доступу до приміщення, але нажаль через ті самі фактори, дані системи не можуть працювати на 100 % ефективно.

Метод розпізнавання обличчя за зображенням під довільним кутом зору є актуальною темою дослідження. Згорткові нейронні мережі є дуже ефективними для задач аналізу зображень та класифікації об'єктів на них.

Метод розпізнавання обличчя за зображенням під довільним кутом зору полягає у використанні алгоритмів машинного навчання, для розпізнавання обличчя за багатьма ознаками (формою носа, сітківки ока, відстані між очима, формою вух) та інших навіть не помітних людському оку, до прикладу як текстура шкіри. Перевагою удосконаленого методу є те, що він показує дозволяє зменшити похибку при розпізнаванні обличчя за зображенням, коли обличчя не повністю зображене на фото та якщо рівень освітленості зависокий або занижкий. Перевага методу полягає тому, що спеціально створеним датасетом, орієнтованим як на зображення під різними кутами зору так і на різні рівні освітленості можна донавчити попередньо навчену нейромережу.

Отже, метод розпізнавання обличчя за зображенням під довільним кутом зору є одним із перспективних напрямів для дослідження, який може сприяти покращенню існуючих засобів безпеки заснованих на розпізнаванні обличчя за зображенням.

Також, даний метод можна використати, для ідентифікації людей на відео, адже будь який відеозапис – це послідовність зображень.

Мета і задачі роботи. Метою кваліфікаційної роботи магістра є вдосконалення методу розпізнавання обличчя за зображенням під довільним кутом зору.

Для досягнення поставленої мети визначено наступні задачі:

- проаналізувати існуючі способи розпізнавання обличчя за зображенням під довільним кутом зору;
- здійснити аналіз згорткових нейронних мереж та їх архітектур для вибору найбільш відповідних для поставленого завдання;
- вдосконалити метод розпізнавання обличчя за зображенням під довільним кутом зору та покращити його для умов недостатньої або надмірної освітленості;
- підготувати набори даних для донавчання згорткової нейронної мережі;
- провести валідацію та аналіз отриманих експериментальних результатів запропонованого методу розпізнавання обличчя за зображенням за стандартними показниками якості.

Об'єкт дослідження – процес розпізнавання обличчя за зображенням під довільним кутом зору.

Предмет дослідження – метод розпізнавання обличчя за зображенням під довільним кутом зору.

Методи дослідження, застосовані для вирішення поставлених завдань: методи та засоби штучного інтелекту (моделі глибокого навчання, згорткові нейромережі), статистичні показники якості моделей глибокого навчання.

Наукова новизна одержаних результатів. В результаті проведеної роботи було вдосконалено метод розпізнавання обличчя за зображенням під довільним кутом зору в умовах недостатнього чи надмірного освітлення, шляхом додаткового навчання початкової моделі, розробленим набором даних, що підвищило статистичні показники якості для усіх розглянутих архітектур моделей глибокого навчання.

Найкращий результат точності у 99,3% було отримано з використанням архітектур «SeNet-50» та «ResNet-50».

Апробація результатів кваліфікаційної роботи магістра та публікації.

Результати роботи опубліковані у науковому журналі – Метод розпізнавання обличчя під довільним кутом зору / Добржанський В.О., Бармак О.В., Скрипник Т.К. // Науковий журнал «Вісник Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2023, №6 (довідка від редакції).

Структура та обсяг роботи. Кваліфікаційна робота магістра складається з завдання, реферату, змісту, переліку скорочень, вступу, 4 розділів, висновків, переліку посилань із 51 найменування та 3 додатків. Загальний обсяг дипломної роботи магістра становить 112 сторінок, з них 84 сторінок основного тексту та 28 сторінок додатків. У роботі наведено 49 рисунків, 8 формул та 13 таблиць.

РОЗДІЛ 1

Аналіз сучасного стану використання інформаційних технологій для розпізнавання обличчя за зображенням під довільним кутом зору

1.1 Аналіз предметної області

В сучасному світі розпізнавання певних об'єктів на зображеннях є дуже розповсюдженим та використовується у багатьох сферах життя, таких як медицина, для виявлення певних захворювань на кардіограмах, та знімках МРТ, на виробництвах для виявлення бракованої продукції, та найбільше у сфері безпеки.

Здебільшого, ці процеси виконуються ручними методами, а саме людиною, і хоча вони можуть мати свої переваги, вони супроводжуються недоліками, такими як:

- низька швидкість, особливо при великих об'ємах даних;
- висока вартість залученого працівника для цього процесу;
- людський фактор, адже людина більш здатна до помилок, їй притаманна втома та суб'єктивність аналізу та судження.

Для вирішення цієї проблеми, впроваджуються технічні засоби, які дозволяють прибрати людський фактор, та автоматизувати багато процесів. Наприклад, дорожні камери розташовані над проїжджою частиною, можуть розпізнати номерний знак автомобіля, і в разі порушення ним швидкісного режиму, поліція може ідентифікувати власника авто [1].

Великий інтерес викликає тема розпізнавання саме обличчя. Сучасні електронні пристрої обладнанні камерами, які дозволяють розблокувати пристрій за обличчям власника, здійснити платіж, провести авторизацію на інтернет ресурсах чи в застосунках. Розробляються та тестуються інформаційні системи безпечних міст, обладнаних камерами відеоспостереження, які в режимі реального часу можуть розпізнавати предмети, певні дії, обличчя тощо [2]. Розпізнавання обличчя здійснюється інформаційно технічними засобами які постійно вдосконалюються.

Будь який відео запис, це деяка кількість кадрів які робляться за певний проміжок часу, тому процес розпізнавання зводиться до аналізу зображення.

Для розпізнавання обличчя за зображенням взятим із камери відеоспостереження, велике значення має якість зображення, а саме його роздільна здатність, чіткість, освітленість та кут огляду.

На сьогодні, найефективнішим інструментом для розпізнавання обличчя за зображенням під довільним кутом зору можна вважати штучні нейронні мережі.

Штучна нейронна мережа (ШНМ) є математичною моделлю, що імітує роботу людського мозку [3]. Вона складається з взаємопов'язаних штучних нейронів, що передають інформацію один одному за допомогою зв'язки з різною вагою.

Даний підхід має ряд переваг:

- ШНМ можуть працювати дуже швидко та ефективно, що дозволяє їм аналізувати великі обсяги даних (зображень);
- ШНМ можуть працювати автономно, , що дозволяє автоматизувати процес та зменшує витрати на персонал;
- ШНМ мають можливість навчатися на нових даних та покращувати свою точність розпізнавання обличчя за зображенням з часом.

ШНМ можуть бути використані у багатьох галузях, наприклад, безпека, правоохоронні органи, медіа та інші, що дозволяє вважати їх універсальними засобами розпізнавання обличчя за зображенням під довільним кутом зору.

Глибоке навчання – це спеціалізована галузь машинного навчання, де робочий процес розпочинається із автоматичного вилучення відповідних характеристик із зображень, замість використання ручного методу [3, 4]. У глибокому навчанні також застосовується «наскрізне навчання», це означає, що мережа отримує вихідні дані та завдання, до прикладу класифікація, і мережа може автоматично навчатися, як отримати відповідний результат. Під час «наскрізного навчання» глибока нейронна мережа самостійно визначає характеристики та параметри зображення, які їй необхідні для розпізнавання об'єктів, що присутні на зображенні.

Алгоритми глибокого навчання можуть масштабуватися зі збільшенням обсягу даних, тоді як неглибоке навчання сходиться після досягнення певного рівня продуктивності.

Неглибоке навчання досягає певного рівня продуктивності, оскільки до мережі додається більше зразків і навчальних даних [5].

Однак головна перевага глибокого навчання полягає в тому, що ці мережі, як правило, вдосконалюються зі збільшенням обсягу даних.

Машинне навчання передбачає ручний вибір функцій і класифікаторів для ідентифікації зображення. Проте, при глибокому навчанні ці кроки виконуються автоматично, завдяки можливості моделі глибокого навчання глибокого навчання самостійно виокремлювати ознаки та будувати моделі.

Глибокі нейронні мережі складаються із трьох типів шарів нейронів. Зв'язки між нейронами мають певну вагу, що визначає важливість вхідних даних.

Нейрони в моделях глибокого навчання є вузлами, через які протікають дані та обчислення [7].

Нейрони працюють таким чином:

- отримують один або декілька вхідних сигналів; ці вхідні сигнали можуть надходити або від набору необроблених даних, або від нейронів, розташованих на попередньому рівні нейронної мережі;

- вони виконують деякі розрахунки;

- вони надсилають деякі вихідні сигнали нейронам глибше в нейронній мережі через синапс [8];

На рисунку 1.1 зображена діаграма функціональності нейрона у нейромережі.

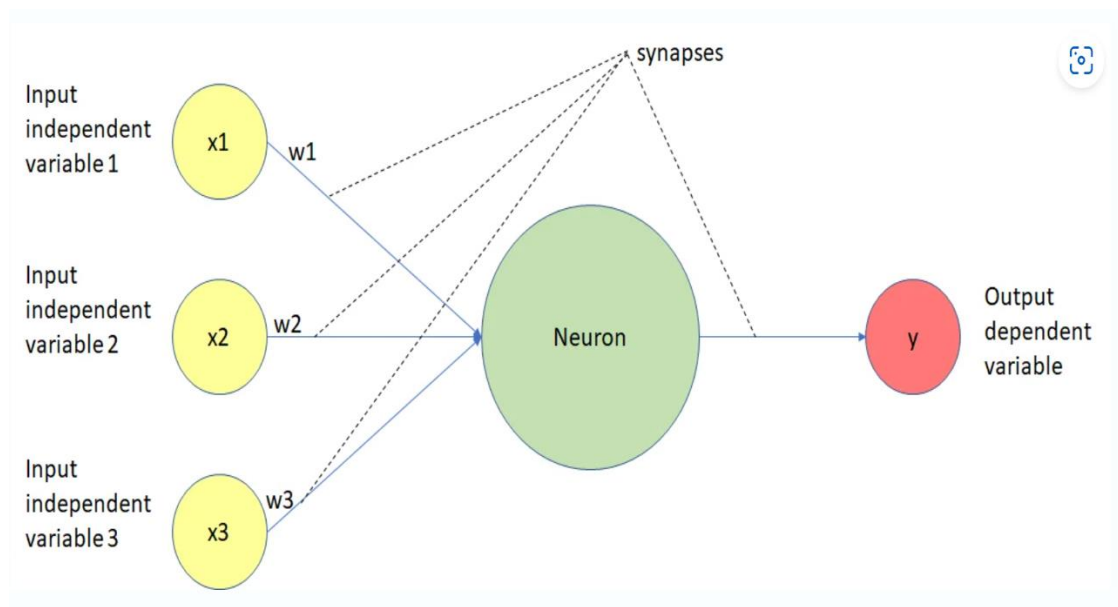


Рисунок 1.1 – Діаграма функціональності нейрона [9]

Глибока нейронна мережа (ГНМ) це – мережа, що включає в себе 2 і більше прихованих шарів. Навчання ГНМ потребує навчання на великому об’ємі вхідних даних, адже від кількості цих даних, залежить ефективність роботи ГНМ у завданнях прогнозування чи класифікації.

У глибинних нейронних мережах нейрони організовані у три види шарів:

- вхідний шар приймає вхідні дані та направляє їх до 1-го прихованого шару, який забезпечує приймання вхідних сигналів та подальшу передачу сигналів у мережу [10];

- прихований шар здійснює обчислення на вхідних даних. Приховані шари займаються обробкою та аналізом вхідних сигналів, проведенням обчислень на цих даних. При створенні мереж такого типу, потрібно визначити загальну кількість прихованих шарів, та скільки нейронів буде міститись в кожному із них;

- вихідний шар відповідальний за повернення опрацьованих даних. Цей шар генерує вихідний сигнал та повертає результат на основі опрацьованих даних [11].

Така структура шарів із нейронами дозволяє глибоким нейронним мережам виконувати складні математичні обчислення та інші завдання в залежності від її призначення.

Існує декілька способів використання нейронних мереж, зокрема:

– навчання ШНМ на власних даних передбачає, що мережа навчається на власному наборі даних, а алгоритм повністю самостійно встановлює ваги нейронів, для відповідності поставленій задачі.

– використання готової архітектури ШНМ з попередньо навченими вагами передбачає використання вже попередньо навченої моделі, яка тренувалась на вхідних даних, що підходять під завдання дослідження. Цей підхід можна використати, коли обчислювальні ресурси обмежені, та/або недостатня кількість вхідних даних для навчання;

– донавчання ШНМ із використанням власних даних передбачає використання попередньо навченої моделі і за допомогою власного датасету – донавчити; цей підхід дозволить швидше отримати високу точність розпізнавання, бо модель вже розуміє певні характеристики і має змогу швидко адаптуватись до нових задач.

Для того, щоб навчити нейронну мережу за допомогою власних даних, потрібно мати коректні мітки, які мережа має прогнозувати. Модель, яка була навчена на обмеженому наборі даних, може виявитися перенавченою, що означає, що вона може не показувати достатньо добрих результатів для прикладів, які не були включені в тренувальні дані, і має обмежену спроможність до узагальнення на нові умови отримання зображення. Тому тренувальний набір даних повинен бути різноманітним і включати різні варіації, такі як освітлення, кут зору, колір і т.д. Самостійне тренування моделі може потребувати багато часу та значних обчислювальних ресурсів, але це дозволяє обмежити класи лише до тих, що необхідні для конкретної задачі. Найкращим варіантом може бути донавчання вже готової моделі за допомогою власних даних. Це дозволяє використовувати вже навчені ваги моделі та скоріше досягти високої точності та збалансованості, налаштовуючи модель до нових даних.

Глибокі нейронні мережі навчаються з нуля, використовуючи набори даних, які містять дуже велику кількість зображень. Дані нейронні мережі здебільшого добре узагальнюються для великої кількості класів. Хоча використовуються всього декілька класів з усіх доступних, більшість зображень об'єктів різних класів мають спільні

ознаки [12]. Функція детектора об'єктів одного класу може ефективно працювати при пошуку об'єктів іншого класу. Це дозволяє перенавчати останні кілька шарів нейронної мережі, а решта шарів вважаються вже створеними засобами вилучення ознак[13].

1.2 Методи глибокого навчання для розпізнавання обличчя за зображенням під довільним кутом зору

У сфері комп'ютерного зору за останні десятиріччя відбувся значний прогрес методів глибокого навчання, які включають рекурентну нейронну мережу (RNN) та згорткову нейронну мережу (CNN). Дані алгоритми на основі глибокого навчання використовуються для виявлення, класифікації та розпізнавання ознак. Основною перевагою згорткової нейронної мережі є те, що вона повністю усуває або суттєво знижує залежність від фізичних характеристик моделей та інших методів попередньої обробки, використовуючи навчання «від кінця до кінця», яке базується безпосередньо на вхідних зображеннях [12]. З цієї причини CNN досягла найвищих результатів у багатьох сферах, в тому числі у розпізнаванні обличчя під будь-яким кутом зору.

CNN (згорткова нейронна мережа) – це штучна нейронна мережа, яка застосовується для аналізу візуальних даних, в тому числі зображень. Основною характеристикою згорткової нейронної мережі є використання згорток, які дозволяють автоматично розпізнавати локальні шаблони або характеристики у зображенні. Згортки - це фільтри, які проходячи по зображенню, виконують функцію згортки, яка полягає у множенні значень пікселів у вікні на відповідні ваги і додаванні результатів. Цей процес виявляє різні характеристики, такі як гострі кути, рельєфи, текстури і т.д.

CNN містить декілька шарів, таких як згорткові, пулінгові та повністю з'єднані шари. Згорткові шари розпізнають характеристики на різних рівнях абстракції, пулінгові шари зменшують розмірність зображення та кількість параметрів, а повністю з'єднані шари виконують кінцевий аналіз та класифікацію [19, 20] CNN

широко застосовується для багатьох задач обробки зображень, таких як класифікація зображень, детектування об'єктів, сегментація зображень, ідентифікація обличч, стилізація зображень та інше. Вона також може бути поєднана з іншими видами нейронних мереж для розв'язання складніших задач аналізу даних. CNN є потужним і ефективним засобом для обробки великих масивів візуальних даних та встановлення складних зв'язків у зображеннях. Вона дозволяє автоматично розпізнавати характеристики, які людському оку можуть бути невидимими, і забезпечує високу точність у багатьох задачах комп'ютерного зору. На рисунку 1.2 показано загальний вигляд згорткової нейронної мережі (CNN).

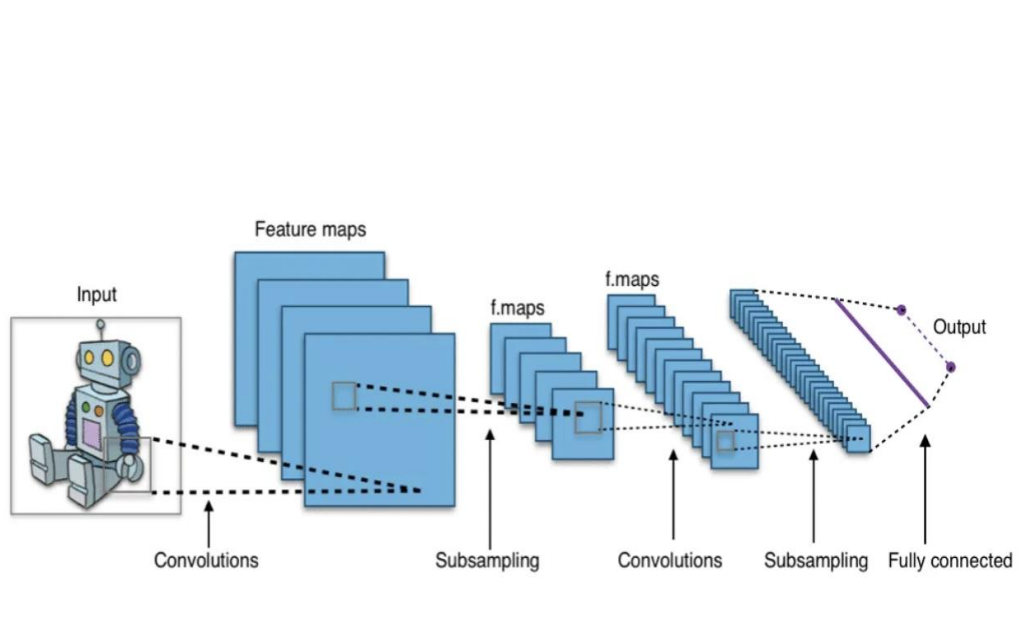


Рисунок 1.2 – Згорткова нейронна мережа [15]

Для задач розпізнавання об'єктів на зображеннях, нейромережі рідко використовуються у початковому вигляді, на сьогодні їх розробляють у вигляді модифікованих рішень, тобто розширюють та змінюють існуючу архітектуру або комбінують декілька архітектур [19].

1.3 Аналіз існуючих публікацій розпізнавання обличчя за зображенням під довільним кутом зору

Розпізнавання обличчя є складним завданням для науковців у всьому світі. Через зміну в освітленні, кута зору чи зовнішнього вигляду та фону може бути дуже складно порівняти схожі зображення за допомогою комп'ютерів, навіть якщо вони так легко розрізняються людьми. Багато дослідників у галузі комп'ютерного зору присвятили свої зусилля поліпшенню основних методів геометричних елементів або узгодження шаблонів. Процес автоматичного виявлення обличчя складається з трьох головних кроків: виявлення обличчя та груба нормалізація, вилучення та точна нормалізація обличчя та ідентифікація або перевірка обличчя. Іноколи задачі виявлення обличчя та виділення об'єктів можуть виконуватися одночасно.

Складність завдання залежить від різних параметрів, наприклад, від розміру бази даних обличчя, виразу обличчя, оклюзії, мінливості фону, шуму та швидкості роботи комп'ютера [15].

Не так давно, одним із традиційних методів розпізнавання обличчя за зображенням був метод Віоли-Джонса, що детально описаний у статті «Multi-View Faces Detection Using Viola-Jones Method» [16].

Ще одним традиційним підходом можна вважати метод зворотного поширення помилки, який детально описаний у статті «Back Propagation in Neural Network: Machine Learning Algorithm» [17].

За останні роки, підвищився розвиток виявлення обличчя за зображенням за допомогою методів глибокого навчання, які суттєво перевершують традиційні методи комп'ютерного зору [18].

У роботі «Multi view face pose estimation based on the supervised ISA learning» наведено метод виявлення обличчя, який інтегрує CNN та 3D-модель середнього обличчя в кінцеву багатозадачну дискримінаційну навчальну структуру [19]. Схема методу зображена на рисунку 1.3.

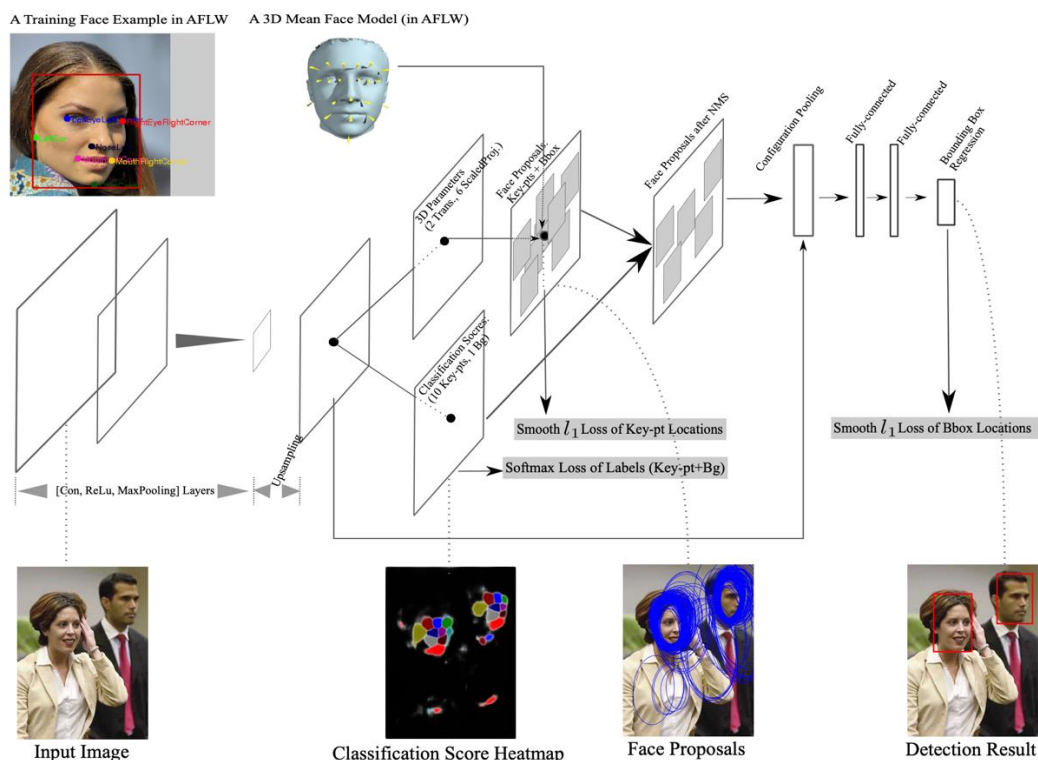


Рисунок 1.3 – Схема роботи 3D Model Based Pose Regularization [20]

Основна ідея полягає в тому, щоб навчити CNN оцінювати параметри 3D-трансформації (обертання) відносно 3D-моделі пересічного обличчя для кожної виявленої ключової точки обличчя (ліве вухо, губи тощо). На 2018 рік, даний метод демонстрував високі результати посеред інших методів, та його недоліком було, те, що модель вимагала навчання на великій кількості обличч різних типів, кольору шкіри, та не показував ефективності, коли зображення було не найкращої якості.

У дослідній роботі «A multimodal deep learning framework using local feature representations for face recognition» пропонують новий мультимодальний підхід до вилучення локальних ознак на обличчі, заснований на об'єднанні переваг різноспрямованих перетворень [21].

Основним внеском цього підходу є застосування перетворення «Curvelet» як швидкого і потужного методу для подання ознак обличчя, а потім оброблення коефіцієнтів «Curvelet» в різних частотних діапазонах з використанням двох різних підходів фрактального виміру для ефективного відображення текстури обличчя.

Пропонований метод здебільшого дає точність 99,0 %, але дослідження не охоплює ситуації, коли обличчя за зображенням розташоване під довільним кутом.

Достойні результати розпізнавання на зображеннях із не ідеальною якістю, показав покращений метод підходу Fast RCNN (Region-Based Convolutional Neural Network) який наведений у статті [20]. Цей метод використовує такий самий підхід як і у Faster RCNN, та складається із двох частин:

- Region Proposal Network (RPN), для генерації списку меж, які ймовірно містять обличчя або так звані області інтересу (Rols) [22, 23];
- мережі Faster RCNN для класифікації межі зображення обличчя та фону.

Спочатку модель Faster RCNN тренується на наборі даних «WilderFace», і цей самий набір використовується для тестування попередньо навченої моделі для складання зразків близьких до позитивних (hard negatives) [24]. Після цього, ці зразки передаються в модель як другий етап навчання, та здійснюємо тонке налаштування на наборі «FRMDB» [25].

Останнім етапом є тонке налаштування (fine tuning) де ми виконуємо навчання на багатьох масштабах (multi-scale) і використовуємо стратегію конкатенації ознак для підвищення продуктивності моделі [26].

Схема запропонованого методу зображена на рисунку 1.4.

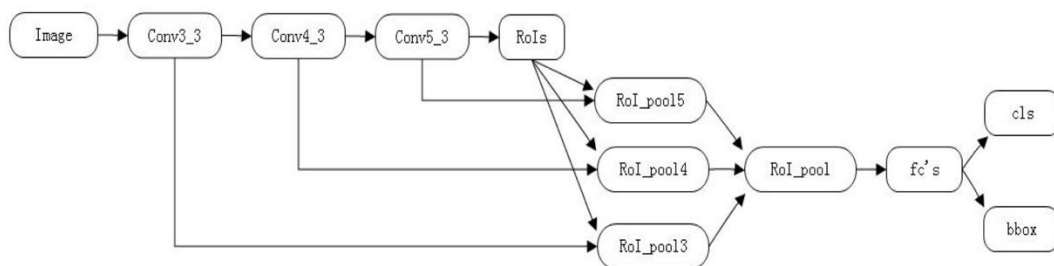


Рисунок 1.4 – Схема Faster RCNN [27]

Даний метод демонструє середню точність у 94,23%, навіть в умовах, коли певна частина обличчя закрита.

Цікавий підхід до розпізнавання обличчя під довільним кутом зору, було запропоновано авторами публікації «Face-Iris Multimodal Biometric Identification System» [28]. У даній роботі, було запропонована мультимодальна біометрична система, яка використовує поєднання загальних рис обличчя та райдужної оболонки ока, для підвищення розпізнавання, уникаючи тим самим, обмежень пов'язаних із одноmodalними біометричними системами.

У методології виділення ознак використовується 2D-фільтр Log-Gabor із різною роздільною здатністю для характеристик райдужної оболонки ока, який фіксує текстурну інформацію в різних масштабах і орієнтаціях [29].

Риси обличчя, з іншого боку, обчислюються за допомогою сингулярного спектрального аналізу (SSA) [30].

Схема сегментація та нормалізації райдужної оболонки зображена на рисунку 1.5.

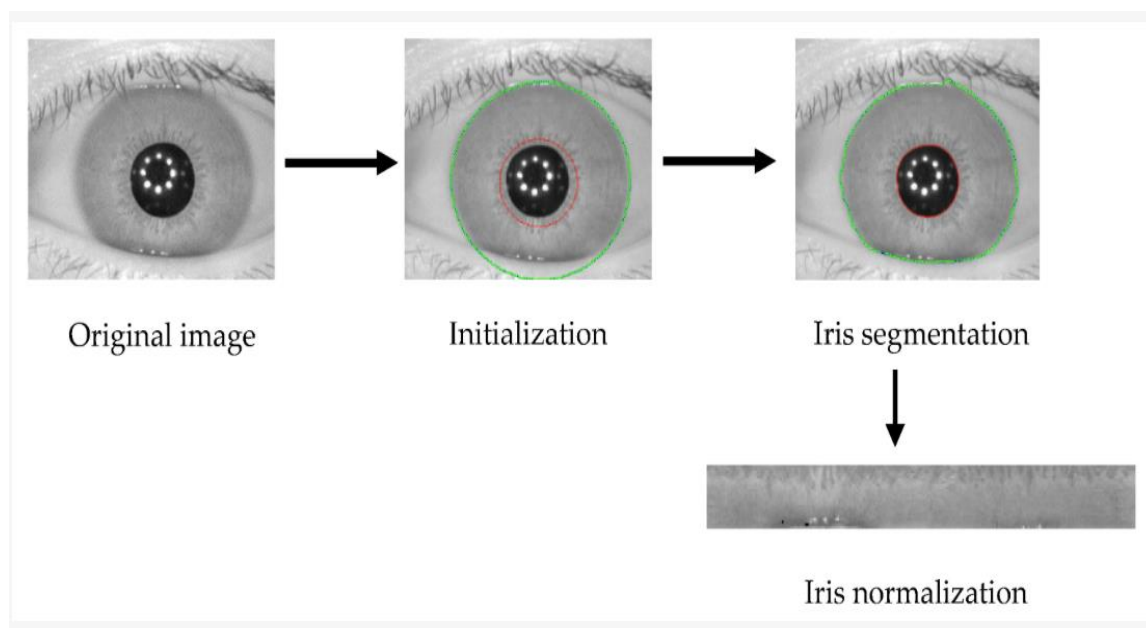


Рисунок 1.5 – Схема сегментація та нормалізації райдужної оболонки [30]

Результати демонструють стійкість системи, та досить високу точність розпізнавання, за умови, що хоча б одне око присутнє на вхідному зображенні. Нажаль, в умовах коли обличчя на вхідному зображенні знаходиться в профіль, ефективність

методи сильно знижується. Середній результат точності, при використанні мережі «VGG-Face» становив 82,62 %. Якщо ж тестувати лише на фото де є чітке зображення ока, середня точність зростає до 94,89%.

Також, варта уваги публікація [31], де був описаний метод, який базується на CNN, та показує високу точність розпізнавання обличчя під довільним кутом зору. Авторами було запропонована стійка нелінійна структура синтезу на основі тензорів для розпізнавання обличчя який може оптимізувати характеристику розпізнавання обличчя FRF та характеристику атрибутів обличчя FAF, використовуючи характеристику тензору низького рангу [32]. Спочатку вони систематично дослідили та перевірили різноманітні сценарії розпізнавання обличчя, такі як поза та освітлення. Потім автори застосували декомпозицію Такера низького рангу [33]. В результаті утворилась нова архітектура, яку назвали «Gated Two-stream Neural Network» (GTNN). Назва була зумовлена тим, що вона на вхід приймає 2 потоки даних, та використовує «gating» механізми, які визначають, яка частина вхідних даних є важливою для поставленої задачі. Для тестування використовувались зображення обличчя під різними кутами огляду, та максимальної середньої точності у 98.8 % вдалось досягнути лише при куті 45 градусів.

Один із підходів до вирішення задачі розпізнавання обличчя під довільним кутом зору був розглянутий у статті [34]. Авторами було проведено дослідження при частковому затемненні верхньої та нижньої частини голови на фото та зміні кута нахилу.

У цьому експерименті використовуються два популярних набори даних: «Extended YouTube Faces» (E-YTF) та «Labelled Faces in the Wild» (LFW) [35, 36].

Мережа «DeerFace» показала середню точність у 94,3 % на наборі даних E-YTF та 92,16 % на наборі даних LFW. Точність розпізнавання була покращена за рахунок попередньо навченої моделі «VGG-Face» та «Lightened CNN» [37, 38]. Автори стверджують, що моделі досягли середньої точності 95,12% на наборі даних E-YTF та 97,63 % на наборі даних LFW.

У роботі [39] запропонували модель розпізнавання обличчя низької роздільної здатності, яка може працювати при роздільній здатності всього 32×32 пікселі. Автори надали однаковий пріоритет як кутовій розбіжності, так і розбіжності величини (розриву величини між парами обличь) з високою роздільною здатністю і відповідними парами обличь з низькою роздільною здатністю. Зображення обличь із низькою роздільною здатністю збільшують кут і розрив величини елементів. Автори стверджували, що всі методи, засновані на надвисокій роздільній здатності, зменшують кутовий розрив і розрив величини між функціями. Ось чому мережа з надвисокою роздільною здатністю досягла 98,46% точності на «LightCNN-v9» але такий показник - при куті нахилу голови 43 градуси [45]. При зміні куту нахилу, точність падає, чим більше кут відрізняється в меншу чи більшу сторону від 43 градусів, тим менший показник середньої точності.

На даний час, існує велика кількість методів та архітектур на основі CNN, які можуть виконувати завдання розпізнавання обличчя під довільним кутом зору. Більшість сучасних методів розпізнавання обличчя базуються на зміні певних параметрів існуючих методів, та перевіряють їх на різних вхідних даних [40]. Нажаль, жоден із оглянутих методів, не дозволяє з великою точністю розпізнавати обличчя під довільним кутом зору в умовах недостатнього чи надмірного освітлення.

1.4 Постановка задачі

Метою кваліфікаційної роботи магістра є вдосконалення методу виявлення обличчя за зображенням під довільним кутом зору. Для виконання поставленого завдання потрібно:

- провести аналіз нейромережових моделей та існуючих підходів для виявлення обличчя за зображенням під довільним кутом зору;
- удосконалити метод виявлення обличчя за зображенням під довільним кутом зору з використанням згорткової нейронної мережі;

- підготувати набір даних для навчання та донавчання згорткової нейронної мережі;
- донавчити попередньо навчену згорткову нейронну мережу виявляти обличчя за зображенням під довільним кутом зору в умовах недостатньої або надмірної освітленості;
- визначити якість, отриманого за запропонованим методом, класифікатора за статистичними показниками.

Успішним виконанням завдання можна вважати більш високу точність запропонованого методу, в порівнянні із існуючим, у процесі виявлення обличчя за зображенням під довільним кутом зору в умовах недостатньої або надмірної освітленості.

Висновки до розділу 1

У розділі проаналізовано предметну область і доцільність використання нейромереж для задач комп'ютерного зору. Розглянуто сучасні підходи та методи розпізнавання обличчя. Аналіз публікацій, дотичних до теми дослідження, підтверджує доцільність використання нейронних мереж з глибоким навчанням для виконання завдань роботи.

За результатами роботи над першим розділом було обґрунтовано актуальність роботи, сформульовано мету та поставлені завдання дослідження.

РОЗДІЛ 2

Метод виявлення обличчя за зображенням під довільним кутом зору

На сьогоднішній день розв'язок задачі виявлення обличчя, зафіксованих на зображеннях під різним кутом зору, ґрунтується на модифікації існуючих підходів або комбінуванні декількох рішень з метою отримання більш ефективного способу. Найбільш розповсюдженні методи розпізнавання обличчя, за основу використовують моделі глибокого навчання, тобто згорткові нейронної мережі різної архітектури.

Реалізація, запропонованого у цій роботі, методу розпізнавання обличчя за зображенням під довільним кутом зору складається з наведених нижче основних кроків.

– *Вибір згорткової нейронної мережі.* Аналіз існуючих згорткових нейронних мереж та вибір відповідної до поставленої задачі.

– *Підготовка вхідного набору даних.* Вибір набору даних (датасету) який найбільш вдало підходять для навчання моделей, та додаткова обробка його програмними засобами для донавчання.

– *Навчання та донавчання глибоких нейронних мереж та оцінка отриманих результатів.* Якість отриманих результатів (навченої нейромережі) оцінюється за відомими статистичними показниками, вибір яких обумовлюється предметною областю задачі та аналізом набору даних. До таких показників відносяться точність (accuracy) та прецизійність (precision), повнота (recall) та аналіз за ROC кривою.

– *Порівняння отриманих результатів;* Порівняльний аналіз отриманих результатів за навчальним, тестовим та іншими наборами даних.

Далі розглянемо зазначені кроки більш детально.

2.1 Моделі глибокого навчання для розпізнавання обличь

Згорткові нейронні мережі (Convolutional Neural Networks, CNN), також відомі як «звичайні» та складаються з серії згорткових та субдискретизуючих шарів, за якими слідують один або кілька повнозв'язаних шарів. Кожен згортковий шар у такій мережі виконує згортки з використанням фільтрів, які навчаються, для вилучення ознак з вхідного зображення. Прикладом традиційної CNN є архітектура Lenet-5, яка була однією з перших успішних нейронних згорткових мереж для розпізнавання рукописних цифр. Вона складається з двох наборів згорткових та субдискретизуючих шарів, за якими йдуть два повнозв'язні шари. Архітектура Lenet-5 продемонструвала ефективність CNN в ідентифікації зображень, і вони стали широко застосовуватися в галузі комп'ютерного зору. CNN зазвичай використовуються для обробки зображень, оскільки вони можуть ефективно виявляти різні ознаки зображення, такі як краї, форми та текстури.

Однією з головних переваг використання згорткових нейронних мереж для розпізнавання обличчя під довільним кутом зору є інваріантність до зсуву. Це означає, що CNN може розпізнавати об'єкти на зображенні незалежно від їхнього розташування.

CNN можуть вивчати різні рівні параметрів вхідного зображення. Верхні шари мережі вивчають складніші характеристики, такі як частини та форми об'єктів, а нижні шари – простіші елементи, наприклад межі та текстури. Ця ієрархічна модель дозволяє розпізнавати об'єкти на різних рівнях абстракції, що є особливо корисним для складних завдань, таких як виявлення об'єктів та сегментація.

Для розпізнавання обличчя під довільним кутом зору запропонованим методом, було розглянуто 3 згорткові мережі, а саме: «ResNet-50», «SeNet-50» та «VGG-16».

«ResNet-50» – модифікація «ResNet» з 48 згортковими шарами (convolution layers), 1 субдискретизуючим (підвибірковим) шаром (pooling layers) і 1 шаром середньої підвибірки. Кількість операцій з плаваючою комою становить $3,8 * 10^9$. Ця

глибока нейронна мережа складається з 50 шарів і використовує блоки зі зв'язками "residual", які дозволяють ефективно вирішувати проблему зниклих градієнтів під час навчання глибоких мереж. Щоб краще зрозуміти структуру «ResNet-50», наведемо детальний огляд її шарів:

- вхідний шар: зображення;
- початковий згортковий шар: перша обробка зображення;
- блок зі зв'язками "residual": два згорткових шари та один шар підвибірки;
- повторення блоку зі зв'язками "residual" ще 3 рази, утворюючи 4 таких блоки;
- max-pooling layers після кожного блоку для зменшення розміру зображення;
- останній блок містить повторний max-pooling layer;
- повнозв'язний шар (FC layer): останній шар для класифікації зображення на категорії.

Блоки зі зв'язками "residual" вирішують проблему зниклих градієнтів, що може виникнути під час тренування глибоких мереж, дозволяючи ефективно навчати модель на великій кількості шарів.

Замість того, щоб передавати вихідний сигнал з одного шару до наступного, блок зі зв'язками "residual" додає вихідний сигнал до виходу на певному етапі обробки внутрішнього блоку, що дозволяє "перенести" частину інформації на наступний шар.

На рисунку 2.1 зображена архітектура ResNet а також оцінка максимально досяжних математичних здібностей, що позначається як «floating point operations per second» (FLOPs) та показує скільки операцій з плаваючою комою мережа може обчислити в секунду.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Рисунок 2.1 – Архітектура ResNet [41]

ResNet-50 має оцінку 3.8×10^9 FLOPs, що є середнім результатом, порівняно із іншими моделями. Грубо кажучи, FLOPs можна використовувати як міру продуктивності комп'ютерних компонентів, таких як центральний процесор (CPU) або графічний процесор (GPU), при роботі із нейромережею.

Також для дослідження, було обрано мережу «Visual Geometry Group» (VGG). Вона була створена групою дослідників із університету Оксфорда, та описана в роботі "Very Deep Convolutional Networks for Large-Scale Image Recognition" [42]. Дана мережа активно використовується для розпізнавання об'єктів на зображеннях.

В залежності від кількості згорткових шарів мережі, було розроблено декілька версій мереж VGG («VGG-11», «VGG-13», «VGG-16», «VGG-19»). Мережі з більшою кількістю фільтрів демонстрували вищу точність на даних «ImageNet». Однак при додаванні більшої кількості шарів після певної межі відбувалося протилежне: точність падала, а помилка збільшувалась.

Архітектура мережі «VGG-16», яка зображена на рисунку 2.2, складається з кількох блоків, кожен з яких містить послідовність згорткових шарів та шарів

«pooling». У кожного блоку є фіксована кількість згорткових та «pooling» шарів, що є перевагою при регулюванні глибини мережі. Архітектура «VGG-16» характеризується значною кількістю параметрів, тому вона вимагає значних обчислювальних ресурсів під час тренування та використання.

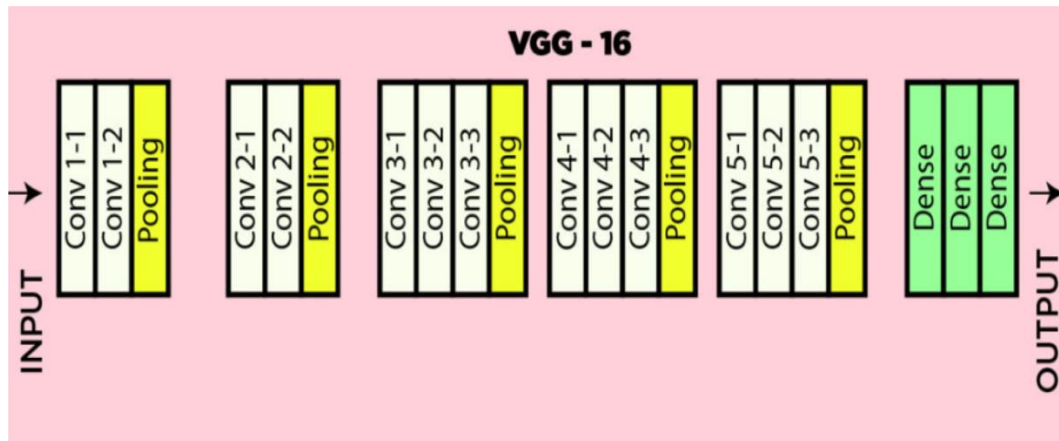


Рисунок 2.2 – Архітектура VGG-16 [43]

Мережа «VGG-16» складається з 5 блоків згортки:

- перший включає в себе 2 шари згортки з 64 фільтрами розміром $3 * 3$;
- другий складається з 2 шарів згортки з 128 фільтрами розміром $3 * 3$;
- третій має 3 шари згортки з 256 фільтрами розміром $3 * 3$;
- четвертий і п'ятий блоки мають по 3 шари згортки із 512 фільтрами розміром $3 * 3$.

Між блоками згортки використовуються шари агрегування, які базуються на «max pooling» для зменшення розмірності даних.

Також, до всіх згорткових шарів застосовується функція активації Rectified linear unit (ReLU). Після згорткових шарів слідує два «convolution» шари з 4096 нейронами на кожен шар. Останнім додається додатковий «fully connected» шар з функцією активації Softmax для класифікації.

Незважаючи на певні недоліки, ця архітектура є ефективною, швидко навчається та легко реалізується за допомогою технології TensorFlow. Графічне відображення архітектури «VGG-16» зображене на рисунку 2.3.

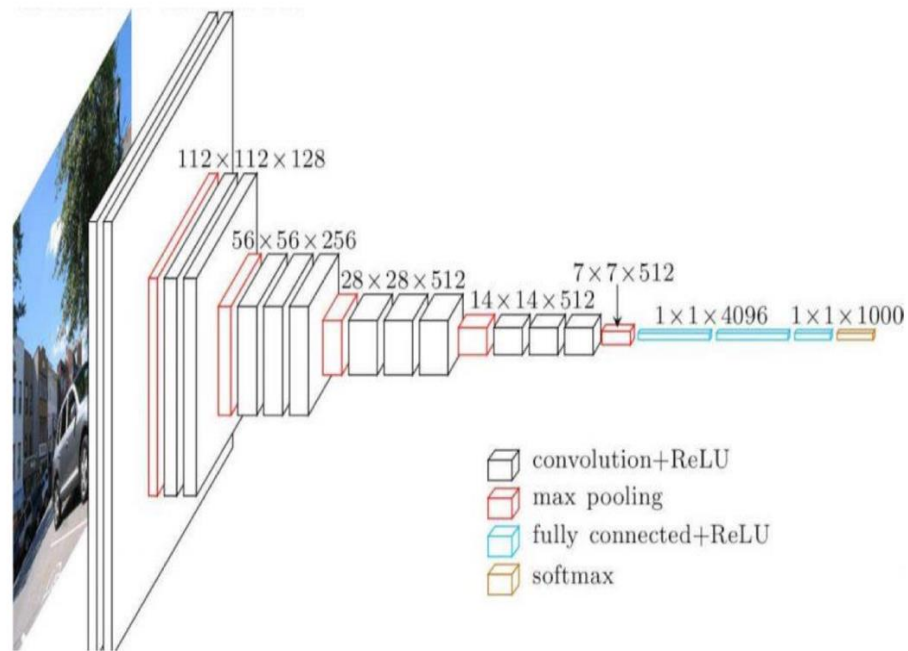


Рисунок 2.3 – Графічне зображення архітектури «VGG-16» [43]

Огляд архітектури VGG-16 показує, що глибина мережі має вплив на її продуктивність, однак при збільшенні глибини нейронних мереж може виникнути «gradient degradation».

«Gradient degradation» – це ситуація, коли градієнти (похідні) під час навчання глибоких нейромереж стають замалими чи занадто великими, що в подальшому призводить до проблем під час процесу оптимізації моделі, та в результаті призводить до невірної класифікації [44].

З цією метою в архітектуру був введений «Residual Connection Block». Його структура включає додавання виходу блоку до входу блоку, формуючи таким чином суму проміжних шарів у блоку та вхідних даних. Графічне подання цього блоку зображене на рисунку 2.4.

Останньою була розглянута мережа «Senet», яка також широко використовується у галузі комп'ютерного зору та роботи із зображеннями. Команда, яка працює над цією архітектурою, активно вивчає існуючі та створює нові сучасні методи глибокого навчання. Нами була обрана модель «SeNet-50», оскільки на нашу

думку, вона найбільше підходить під завдання розпізнавання обличчя під довільним кутом зору.

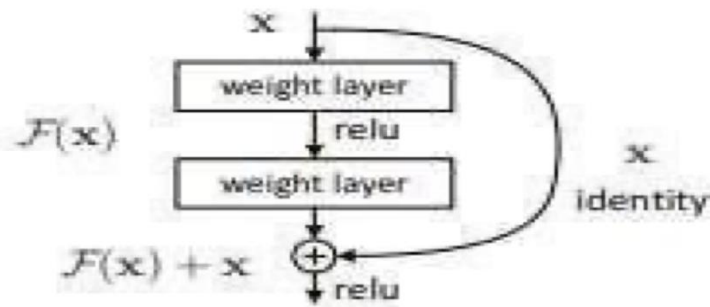


Рисунок 2.4 – Residual Connection Block [45]

«Senet-50» визначається своєю структурою, що містить послідовність згорткових та пулінгових шарів, розташованих у блоках. В кожному з цих блоків фіксована кількість згорткових та пулінгових шарів, надаючи можливість зручного контролю за глибиною мережі. Архітектура «Senet-50» вирізняється ефективністю та високою точністю, але водночас вимагає вагомих обчислювальних ресурсів для тренування та застосування.

Архітектура «SeNet-50» зображена на рисунку 2.5, та у загальному випадку складається із двох кроків.

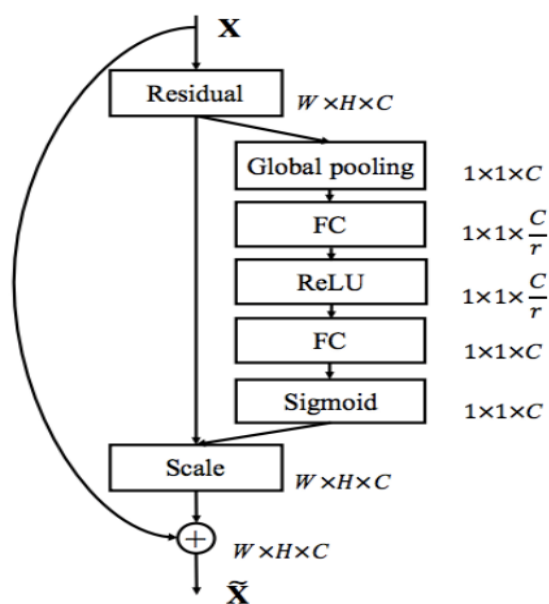


Рисунок 2.5 – Архітектура мережі SeNet-50 [51]

Першим етапом виступає стискання, завдяки якому, генерується статистика по «channel-wise»:

$$Se = F_{SeNet}(y_q) = \frac{1}{H*W} \sum_{i=1}^W y_q(I, J), \quad (2.1)$$

де F_{SeNet} – функція стискання;

y_q - це елемент карти ознак з просторовими розмірами $H*W$,

де y_q це q -й елемент Se , $i q = 1, 2, \dots, D$.

Другим етапом, є процес «excitation», який забезпечує та визначає залежності по каналах та значно мінімізує кількість параметрів за допомогою повністю пов'язаних шарів, сигмоїдальних та ReLU функцій. Нижче наведена формула даного етапу:

$$T = F_{excitation}(Se, W) = \sigma(G(Se, W)) = \sigma(W_2 \delta(W_1 Se)), \quad (2.2)$$

де $T = t_1, t_2, \dots, t_D$;

$F_{excitation}$ - функція «excitation» $t_q \in R^{H*W}$, $\delta(x) = \text{MAX}(x, 0)$ та належить функції ReLU;

G – належить глобальній функції та $\sigma(x) = \frac{1}{1+e^{-x}}$ є функцією сигмоїди $P = F_{scale}(Se_q, y_q) = Se_q \cdot y_q$.

Для покращення методу розпізнавання обличчя під довільним кутом зору в умовах недостатнього чи надмірного освітлення, ми використали 3 згорткові мережі, які підтримуються популярними фреймворками. Результати тестів визначають мережу із найбільшою ефективністю запропонованого методу.

Мережа «VGG» з архітектурою «VGG-16» використана у дослідженні, оскільки головною її особливістю є те, що вона задіює невеликі фільтри із розміром ядра $3 * 3$ пікселі та застосовує глибокі згорткові шари. Хоч дана архітектура і не є

новою, але її можна вважати фундаментом у розробці методів комп'ютерного зору, і вона не потребує значних обчислювальних ресурсів. Вона досить ефективна, легко піддається навчанню та підтримується багатьма популярними фреймворками, такими як : «PyTorch», «Keras» та «Tensorflow».

Наступною для дослідження обрана мережа «SeNet» із архітектурою «SeNet-50», тому що в ній доданий блок "squeeze-and-excitation" (стискання та розширення), який дозволяє мережі фокусуватись на найбільш важливих ознаках завдяки функції стискання, яка базується на «channel-wise», що в свою чергу, дозволяє вдосконалювати точність розпізнавання обличчя під довільним кутом зору. Вона також підтримується багатьма відомими фреймворками, але їй потребує значно більших обчислювальних ресурсів ніж «VGG-16», адже має 50 шарів згортки.

Останньою для покращення методу розпізнавання обличчя під довільним кутом зору в умовах недостатнього чи надмірного освітлення використана мережа «ResNet» з архітектурою «ResNet-50». Завдяки використанню 50 шарів та блоків зв'язками "residual", вона вирішує проблему зниклих градієнтів, що дозволяє їй аналізувати складні характеристики та залежності, що в умовах поставленої задачі є вагомим фактором. Дана мережа також легко піддається навчанню та підтримується вищевказаними фреймворками.

2.2 Підготовка вхідних даних

Для реалізації методу розпізнавання обличчя за зображенням під довільним кутом зору а також для покращення розпізнавання в умовах недостатньої чи надмірної освітленості, використано 3 набори даних, що є зображеннями обличчя різних людей, під різними кутами зору.

Першим набором даних, який ми в ході експерименту може використовуватись для навчання є – «VGGFace» [37], який містить у собі більше 2,6 мільйонів фото для 2,622 осіб, здебільшого відомих публічних людей (акторів, співаків, політиків тощо). Він містить зображення обличчя людей під різними кутами

зору. Зображення подані у різних позах та з різними емоціями на обличчя, але усі вони чіткі, та на них видно більшу частину обличчя. Приклад деяких фото із набору даних «VGGFace» зображено на рисунку 2.6.



Рисунок 2.6 – Обличчя із набору даних «VGGFace» [37]

Іншим набором даних, який використовується для донавчання, є «Face Recognition from Mugshots Database» (FRMDB) [25]. Даний датасет, є набором фото 39 людей, із яких – 17 жінок та 22 чоловіків. Середній вік людей, складає 24,6 років, причому наймолодшому 19 років, а найстаршому 52 (стандартне відхилення = 7,8). Для кожної людини було зроблено 28 кольорових фотографій, під різними кутами зору, де також присутнє фронтальне фото.

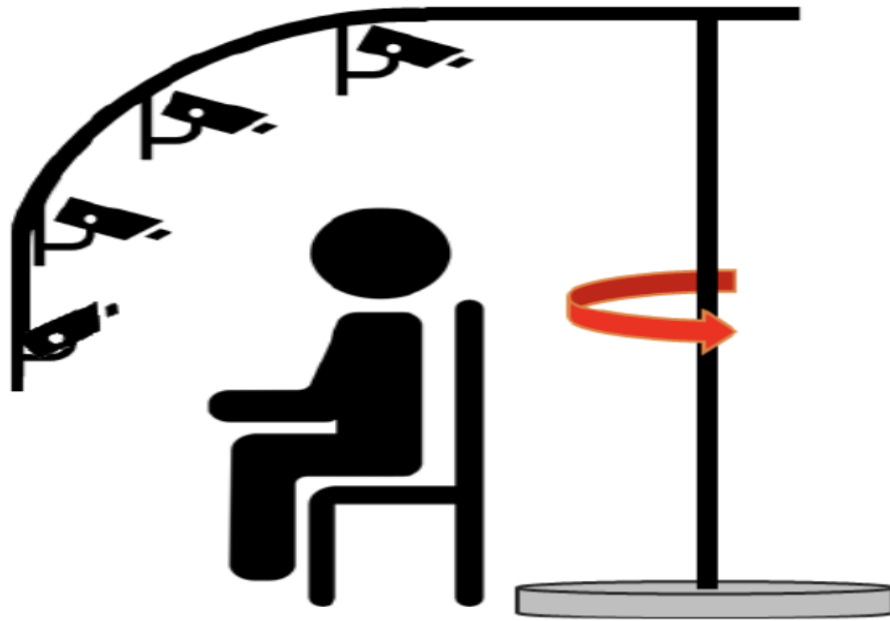


Рисунок 2.7 – Роботизований штатив із камерами [25]

Дані фото було зроблені за допомогою роботизованого штативу, який обертається навколо своєї осі із прикріпленими фотокамерами, як зображено на рисунку 2.7, що в результаті дає фото із 7 кутами огляду в горизонтальній площині та 4 у вертикальній площині (рис. 2.8).

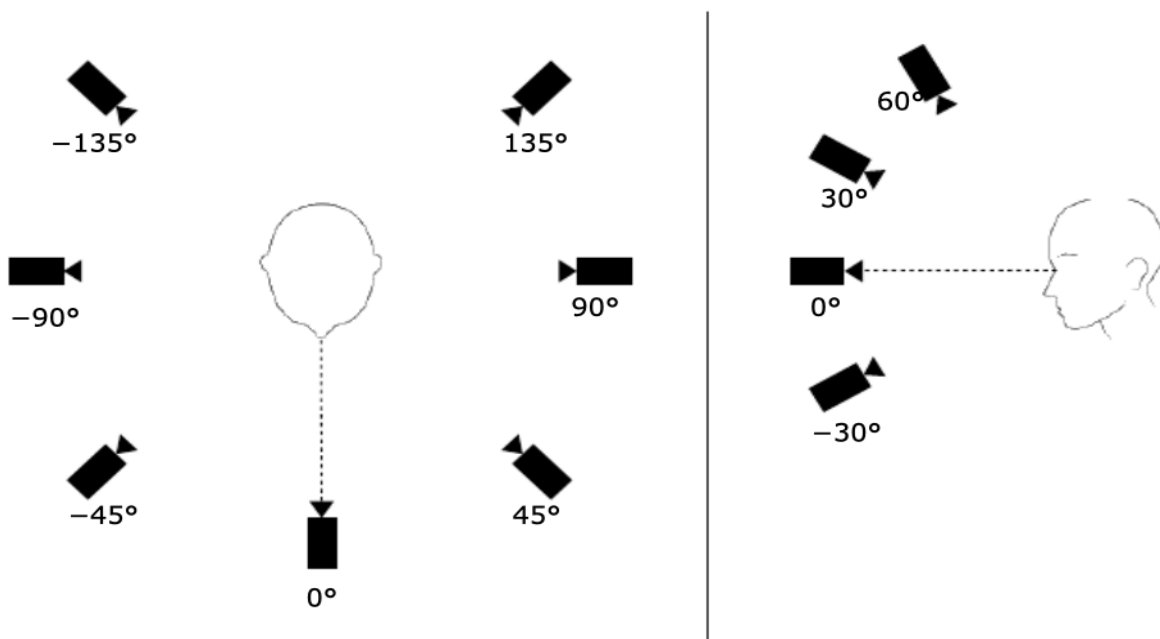


Рисунок 2.8 – Кути огляду вертикальній та горизонтальній площині [46]

Використовувалось студійне освітлення і в результаті, отримано 28 фото під різними кутами для кожної людини, де обличчя чи його частину добре видно.

На рисунку 2.9 показано приклад набору фото людини під номером 31 у цій базі з позначенням кутів огляду.

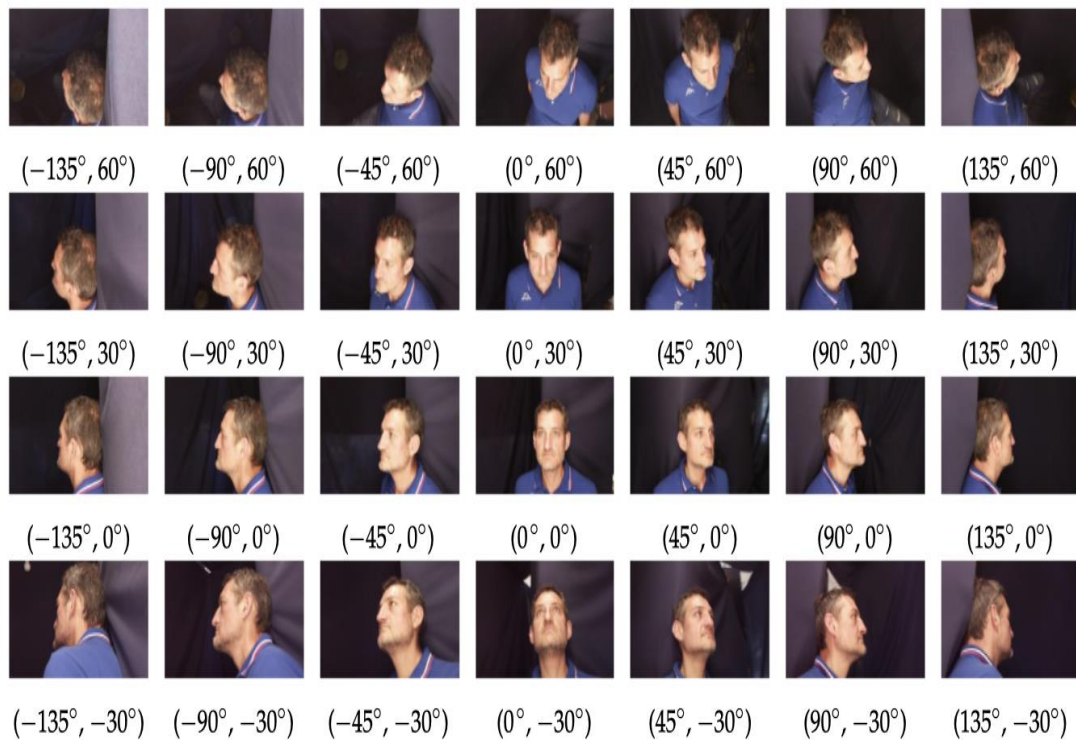


Рисунок 2.9 – Особа 31 із набору даних «FRMDB» [25]

Задля тестування роботи методи, використано датасет «Security Cam Face» (SCFace) [47], який включає в себе відеозаписи із камер спостереження, на яких присутні люди із набору «FRMDB». Дані відеозаписи у кодуванні H.264, зроблені у форматі (Matroska - .mkv) та зняті із частотою 60 кадрів на секунду. Для кожної людини наведені відеозаписи із 5 камер спостереження, які знаходяться у різних точках приміщення, відповідно людина зображена під різними кутами зору. Всі відеозаписи зроблені в одному приміщенні, при однаковому освітленні. Для формування навчальної вибірки, для запропонованого методу, з кожного відеозапису взято по довільному кадру.

Приклад отриманого нового набору даних (номер 31) зображено на рисунку 2.10.

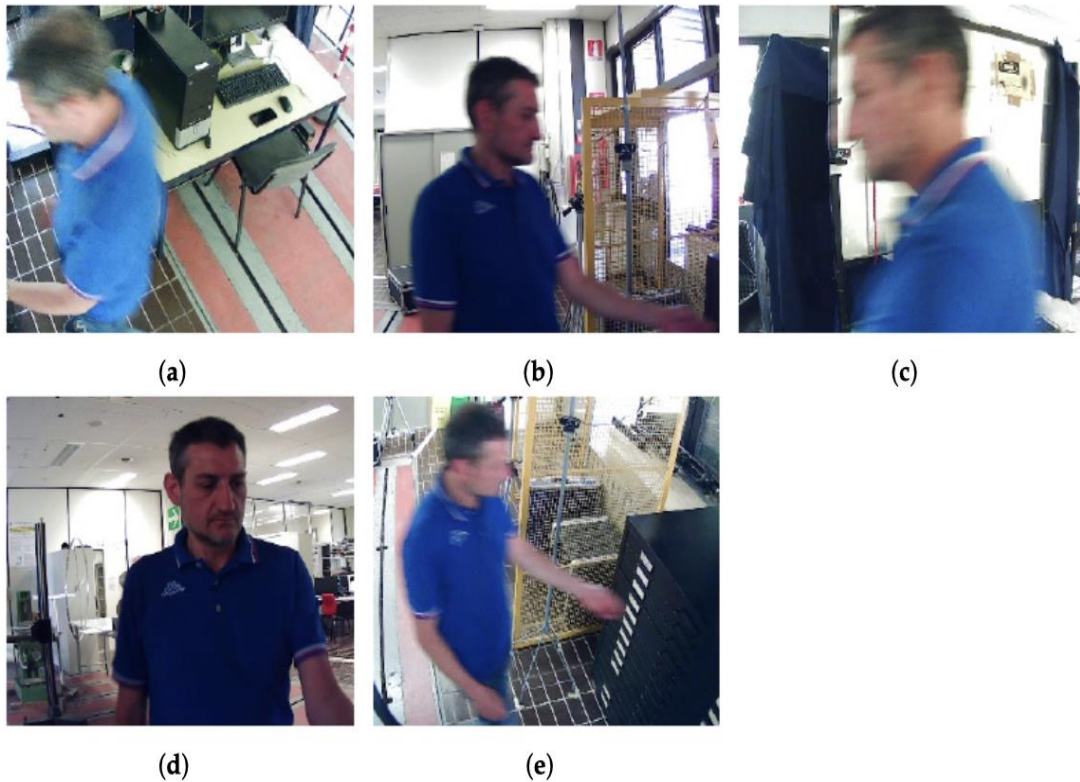


Рисунок 2.10 – Приклад зображення обличчя із сформованого датасету

Дані набори даних сформовані наведеним вище чином, тому що вони якомога краще підходять для задачі вдосконалення методу розпізнавання обличчя під довільним кутом зору, оскільки включають у себе фото обличчя із великою кількістю ракурсів. Далі, наведені набори даних доповнено модифікаціями існуючих, для моделювання зображень обличчя в умовах недостатньої та надмірної освітленості.

2.3 Навчання та донавчання моделі

Важливою частиною забезпечення ефективності роботи нейронної мережі є її навчання та донавчання. Навчання, обраних архітектур нейромереж, здійснювалось з нуля, за набором даних «VGGFace» використовуючи триплетну функцію витрат та стохастичний градієнтний спуск (SGD) для оптимізації.

Зображення проходить через набір згорткових шарів, де застосовуються фільтри з дуже малою областю чутливості: 3×3 (це найменше значення, щоб вловити поняття ліво / право, верх / низ, центр).

Всі приховані шари обладнані випрямленням нелінійності ReLU, що дозволяє діставати різні ознаки для вхідного зображення. Прикладом є рисунок 2.11.

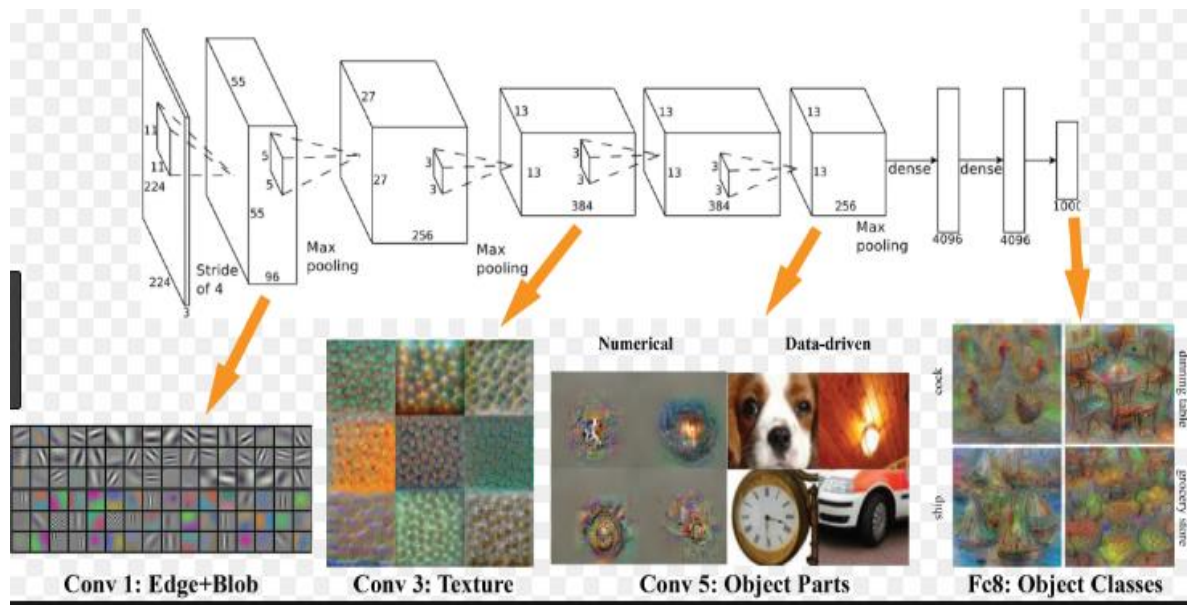


Рисунок 2.11 – ReLU на прихованих шарах мережі [48]

Функція триплетних витрат використовується у глибокому навчанні і її суть полягає в тому, що вхідні дані, які називають anchor (прив'язка) порівнюються із даними які є вірними (позитивними), та тими на яких об'єкт розпізнавання не присутній (негативні), в нашому випадку – обличчя людини. В результаті, відстань прив'язки від позитивних значень зменшується, та відповідно збільшується для негативних.

Тренування триплетних витрат спрямовано на порівняння векторів даних, що дозволяє визначати їх схожість між собою і виконується за допомогою відстані Евкліда, яка визначається за формулою (2.3).

$$E = \sqrt{\sum_{i=0}^n (p_i - q_i)^2}, \quad (2.3)$$

де n – кількість елементів вектора, p – значення 1-го вектору, а q – значення 2-го вектору.

На прикладі мережі, що навчена на розпізнавання зображень обличчя з використанням функції контрастних витрат, її результатом буде мала відстань у випадку якщо зображення обличчя «вже відоме» та велика відстань, якщо обличчя було ідентифіковано. Перевагою використання функції триплетних витрат, є те, що результатом є так званий рейтинг, зображень обличчя які найбільше схожі на вхідне, що дозволяє більш точно доналаштувати мережу для вірного розпізнавання. На рисунку 2.12 показана схема роботи функції триплетних витрат.

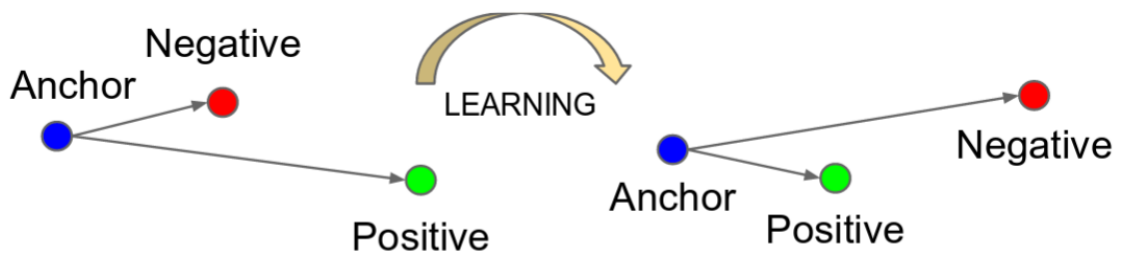


Рисунок 2.12 – Схема функції триплетних витрат [49]

Формула (2.4) функції триплетних витрат має вигляд:

$$\mathcal{L}(A, P, N) = \max(\|f(A) - f(P)\|_2 - \|f(A) - f(N)\|_2 + \alpha, 0), \quad (2.4)$$

де A – прив'язка, P – позитивні вхідні дані класу A , N – негативні вхідні дані класу A , α – відстань між позитивними та негативними парами, f – вбудовування.

Також у процесі навчання використаний стохастичний градієнтний спуск (SGD). Це такий процес оптимізації, який шукає оптимальне значення цільової функції. Цей метод розповсюджений у задачах машинного навчання, і використовується для зменшення функції витрат. Його основна мета – це визначення параметрів моделі, які забезпечують максимальну точність як для навчальних, так і

для тестових вхідних даних. У SGD, градієнт є вектором, що вказує на напрямлення у конкретній точці.

Класичний градієнтний спуск робить зайві розрахунки для великого набору даних, бо він знову і знову обчислює градієнти для схожих прикладів перед кожною зміною параметрів. Перевага стохастичного градієнтного спуску над класичним у тому, що він уникає надлишкових обчислень, роблячи одне оновлення на кожному кроці, тому він зазвичай набагато швидший, що стало причиною вибору його у нашому дослідженні.

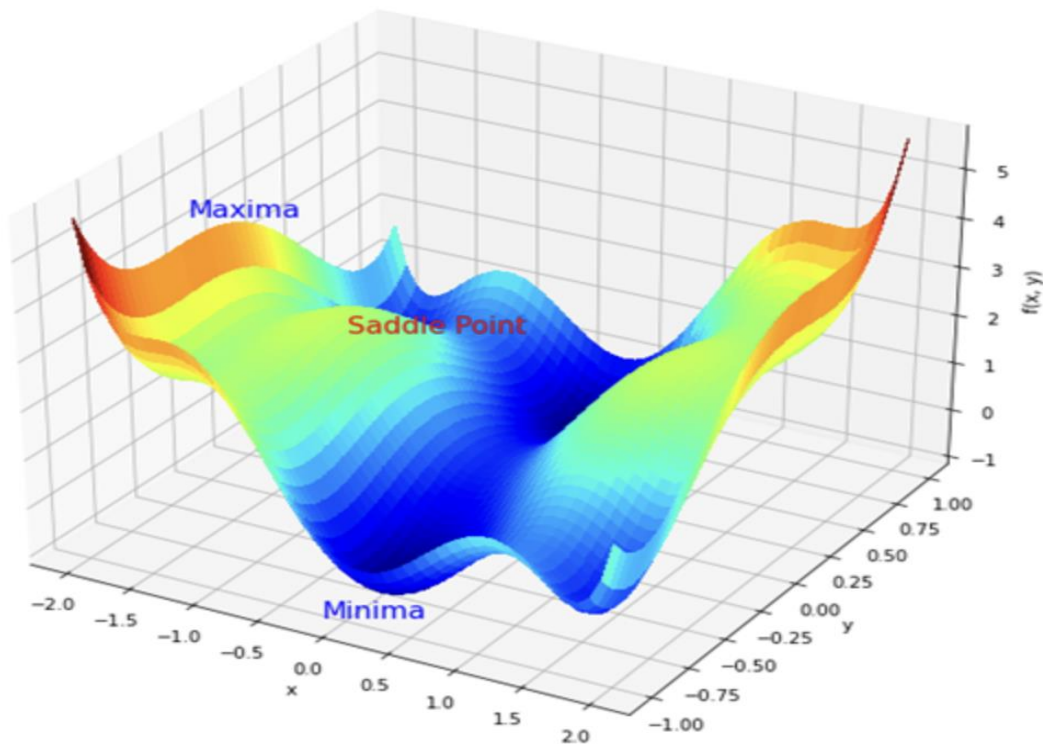


Рисунок 2.13 – Стохастичний градієнтний спуск [50]

Алгоритм стохастичного градієнтного спуску, переміщується в просторі параметрів, в напрямку найбільшого ухилу. Приклад SGD зображено на рисунку 2.13.

При використанні стохастичного градієнтного спуску, ключовим параметром є швидкість навчання. Якщо швидкість навчання занадто висока, алгоритм може ігнорувати «вузькі» улоговини, а при занадто низькій швидкості, може провалитись у один із локальних мінімумів і там залишитись.

Для навчання нейромереж використовують набори вхідних даних які підходять під завдання. Набори даних можна створювати самостійно, або ж, скористатись тими, що наявні у відкритому доступі. Коли мережа навчається, вона переглядає набір даних один раз, а при навчанні мережі з учителем – багато разів. Процес, коли мережа пройшла навчальний сет прийнято називати ітерацією. Ітерацію, можна представити як лічильник тренувальних сетів, що пройдені мережею. Одне повне опрацювання навчальної вибірки алгоритмом навчання називають епохою. Під час початкового налаштування нейронної мережі, значення епохи становить нуль, а максимальне значення епохи, може бути задане вручну. Якість натренованості визначається значенням епохи, що збільшується із кожним повним проходженням навчального набору даних, і чим епоха більше, тим кращим буде результат нейронної мережі.

Під час навчання з учителем, набір вхідних даних прийнято ділити на навчальну вибірку та тестову. Навчальні вхідні дані використовуються для навчання нейромережі, а тестові для оцінки ефективності роботи по завершенню навчання. Поділ набору вхідних даних на навчальні та тестові може бути довільний, але тестову вибірку неможна використовувати для навчання.

З кожною епохою навчання, помилка на тестових даних повинна зменшуватись, але також, збільшення епох може призвести до перенавчання мережі. Перенавчання, це той випадок, коли помилка на тестовій вибірці збільшується, а при цьому помилка на навчальній вибірці зменшується, та означає, що нейромережа вже не займається узагальненням, а запам'ятовує навчальну вибірку. Здебільшого, в таких випадках, процес навчання зупиняють.

Під час навчання з учителем набір вхідних даних зазвичай ділять на дві частини – навчальну вибірку і тестову вибірку. Принцип поділу може бути випадковим. Навчальні дані подаються до мережі для навчання, а тестові використовуються для оцінки якості роботи мережі після завершення навчання (тестові дані неможна застосовувати для навчання). Таким чином, якщо на тестових даних помилка зменшується, то мережа правильно виконує узагальнення. Якщо ж

помилка на навчальних даних продовжує зменшуватися, а помилка на тестових даних збільшується, це означає, що мережа вже не виконує узагальнення і просто «запам'ятовує» навчальні дані. Це явище називається перенавчанням мережі або оверфітінгом. У таких випадках навчання зазвичай припиняють.

Перенавчання може виникати, коли нейронна мережа адаптується до конкретних особливостей навчального датасету, а не до загальних закономірностей в даних. Для уникнення перенавчання, не потрібно довго тренувати мережу на тих самих чи дуже схожих даних. Також, перенавчання може бути спричинено некоректними підбором гіперпараметрів або не відповідною архітектурою.

Гіперпараметри – це такі параметри, значення яких встановлюється перед початком процесу навчання, як приклад, швидкість навчання, загальна кількість прихованих шарів та нейронів у них, тощо. В процесі навчання можуть виявитись інші проблеми, такі як: попадання мережі до локального мінімуму поверхні помилок або ж параліч. Нажаль, передбачити виникнення однієї чи іншої проблеми зазделегідь неможливо, відповідно важко надати рекомендації як їх уникнути.

У ході дослідження також може відбуватись процес донавчання. Під час нього, використовують ті самі підходи, що і під час основного навчання. Навчання здійснюється на загальному наборі даних, що дозволяє мережі виявити найбільш значущі та узагальнені параметри, а донавчання виконується значно меншими наборами, де присутні конкретні особливості, в залежності від завдання.

2.4 Оцінка якості навченої нейромережі

Для визначення якості роботи запропонованого методу використаємо наступні статистичні показники: точність (accuracy та precision), recall, f1 score та проведемо аналіз за ROC кривою.

Для визначення загальної точності скористаємось формулою:

$$Accuracy = \frac{TP+FN}{N} * 100, \quad (2.5)$$

де TP – кількість вірно класифікованих позитивних результатів, FN – вірно класифікованих негативних результатів, N – загальна кількість спроб.

Recall вимірює відсоток правильно визначених позитивних елементів відносно загальної кількості фактичних позитивних елементів. Високе значення recall свідчить про ефективність моделі в правильному визначенні позитивних прикладів.

Формула для обчислення recall (чутливість) виглядає наступним чином:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (2.6)$$

де – TP кількість правильно класифікованих позитивних результатів, FN – кількість правильно класифікованих негативних прикладів.

Precision вимірює відсоток правильно класифікованих позитивних прикладів відносно загальної кількості прикладів, які модель визначила як позитивні. Високе значення precision свідчить про те, що модель досить точно визначає позитивні приклади.

Формула для обчислення precision (точність) виглядає наступним чином:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (2.7)$$

де – TP кількість правильно класифікованих позитивних результатів, FP – кількість неправильно класифікованих позитивних прикладів.

F1-score використовується для об'єднання інформації про точність та чутливість моделі в один показник. F1-score найкраще працює в ситуаціях, коли обидві точність та чутливість важливі для оцінки ефективності моделі. Значення F1-score знаходиться у діапазоні від нуля до одиниці, де високе значення вказує на добре збалансований компроміс між точністю та чутливістю.

Формула для обчислення F1-score (F1-міра) виглядає як гармонічне середнє між precision та recall і має наступний вигляд:

$$\text{F1 score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (2.8)$$

«Receiver operating characteristic» (ROC) крива – це графік, який використовується для візуалізації ефективності бінарного класифікатора при різних значеннях порогу. ROC крива графічно зображує, як змінюються чутливість та специфічність класифікатора при зміні порогу прийняття рішення. Ідеальний класифікатор матиме ROC криву, яка проходить через верхній лівий кут графіка (0 False Positive Rate і 1 True Positive Rate), тоді як випадковий класифікатор буде мати ROC криву, що наближається до діагональної лінії.

На рисунку 2.14 можна побачити приклад графіку на якому зображена ROC крива.

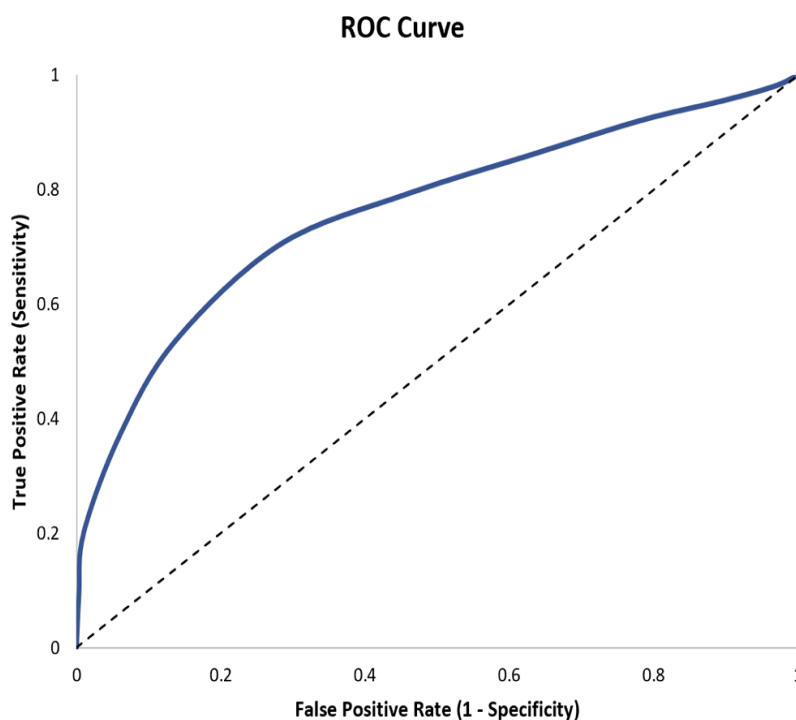


Рисунок 2.14 – ROC крива [51]

Аналіз за ROC кривою дозволяє оцінювати ефективність класифікатора, особливо в контексті задач класифікації. А площа під ROC кривою (AUC-ROC) надає одну цифрову метрику для порівняння різних класифікаторів: чим більше AUC-ROC, тим краще функціонує класифікатор.

На вісі X розташовані не правильно класифіковані позитивні результати, а на вісі Y – правильно класифіковані позитивні результати.

ROC крива допомагає визначити оптимальний поріг для класифікації, що дозволяє зберегти баланс між TPR і FPR в залежності від вимог задачі. Якщо крива підіймається вгору і вліво, класифікатор є ефективним.

Висновки до розділу 2

В другому розділі наведено основні кроки для вдосконалення методу виявлення обличчя за зображенням під довільним кутом зору. В якості нейромережевого засобу було розглянуто 3 згорткових нейронних мережі (CNN). Наведено опис а також архітектуру даних мереж.

Розглянуто 3 набори даних, які були обрані для навчання, донавчання та тестування методу.

Розглянуто методи навчання для нейронних мереж. Навчання було здійснення з використанням функції триплетних витрат та алгоритму стохастичного градієнтного спуску.

Розглянуто показники оцінки якості роботи методу та наведено їх опис та формули.

Розглянуто спосіб покращення виявлення обличчя за зображенням під довільним кутом зору зображень із недостатньою та надмірною освітленістю, шляхом донавчання мережі.

РОЗДІЛ 3

Проектування прототипної інформаційної системи для реалізації вдосконаленого методу розпізнавання обличчя за зображенням під довільним кутом зору

3.1 Вибір платформи, технологій та бібліотек

Реалізація методу розпізнавання обличчя за зображенням під довільним кутом зору потребує розробки декількох програм для підготовки вхідних даних та для розпізнавання обличчя. Існує велика кількість інтегрованих систем для розробки програмного забезпечення (IDE), які включають до свого функціоналу редактор коду, компілятор чи транслятор коду та інструменти для відлагодження. В якості середовища для програмування, розглянуто «Jupyter Notebook», яке має зручний інтерактивний інтерфейс для аналізу даних та має підтримку багатьох мов програмування. До переваг даного середовища можна віднести наведені нижче.

– *Інтерактивність та розділена структура.* Головний інтерфейс «Jupyter Notebook» розділений на комірки, кожна з яких може містити код, текстову інформацію, та результати виконання коду. Завдяки розділенню на комірки, код можливо виконувати поетапно, та спостерігати за його виконанням у реальному часі, що дає можливість легко тестувати та виправляти помилки. Набір комірок які готового рішення, прийнято називати блокнотом. «Jupyter Notebook» підтримує розмітку «Markdown», і це дозволяє формувати блокноти з документацію, яка легка для сприйняття, та відділити код від коментарів.

– *Підтримка різних мов.* Середовище «Jupyter Notebook» отримало свою назву від 3 мов які підтримуються даним середовищем, а саме : «Julia», «Python», «R», що в свою чергу дозволяє використовувати код на різних мовах в одному блокноті.

– *Взаємодія із системою.* Дане середовище надає можливість працювати із файловою системою, що дозволяє легко оперувати із файлами операційної системи для використання їх як вхідних даних, так і збереження результатів. Середовище

дозволяє використовувати «shell commands» прямо з блокнота, а також підтримує встановлення різноманітних додатків для розширення функціоналу.

– *Візуалізація.* Середовище «Jupyter Notebook» дозволяє візуалізувати результати прямо в своєму інтерфейсі та підтримує зображення, графіки, таблиці, HTML сторінки та PDF документи, тому вам не потрібно перевіряти результати які збережені у теках.

– *Сумісна робота.* Блокнот в «Jupyter Notebook» можна легко поширити для сумісної роботи, в свої структурі він являє собою текстовий документ, який зберігає усі заповненні комірки, а також конфігурацію системи на якій виконується. За наявності конфігурації, інша людина, яка буде працювати із блокнотом, зможе використати версію, яка наведена у блокноті, що дозволяє бути впевненим у ідентичності середовищ де буде виконуватись код. Також, існує багато онлайн інструментів, які дозволяють запускати виконання коду із блокнотів, та дозволяють одночасно працювати декільком людям над одним проектом. Нажаль, в таких системах підтримка файлової системи та «shell commands» обмежена, та в більшості обчислювальні ресурси які надаються для безкоштовного використання дещо обмежені.

Для реалізації методу розпізнавання обличчя розглянуто мову програмування «Python». Даний вибір був зумовлений тим, що мова «Python» широко використовується у задачах із використанням штучного інтелекту та комп'ютерного зору. Більшість наукових праць, розглянутих у розділі 1, проводили дослідження по розпізнаванню об'єктів на зображеннях, саме із використанням мови «Python». Ця мова підтримує величезну кількість бібліотек та фреймворків для роботи із нейронними мережами, що дозволяє легко мінімізувати об'єм коду та використовувати готові рішення для рутинних задач. До основних характеристик та переваг цієї мови можна віднести наведені нижче.

– *Потужність та простота.* «Python» – високорівнева мова програмування, яка дозволяє розробляти будь які додатки для різних цілей, від простих веб сайтів, до високонавантажених систем обробки даних. При тому, синтаксис мови дуже простий,

легко читається, не є типізованим, що дозволяє не думати про тип змінних які створювати, або передивлятися код, на наявність пропущених дужок.

– *Кросплатформенність*. Застосунки створенні за допомогою мови «Python», можуть працювати на різних операційних системах. Також, код не потребує компіляції, що сильно спрощує процес розробки та дозволяє використовувати багато зручних готових середовищ, таких до прикладу, як згадуване вище «Jupyter Notebook».

– *Широка розповсюдженість*. Оскільки мова «Python» є легкою в осяганні, вона досить швидко набрала своє популярності, як серед розробників програмного забезпечення, так і серед науковців, що сформувало велику та дружню спільноту навколо неї. У відкритому доступі легко можна знайти документацію, уроки, та прикладу коду. Кількість готових бібліотек та фреймворків для роботи із неймережами, робить її для нас очевидним вибором для розробки методу розпізнавання обличчя за зображенням під довільним кутом зору.

Реалізацію запропонованого методу можна пришвидшити, завдяки використанню готових бібліотек та фреймворків, наведених нижче.

– *Tensorflow*. Це відкрита та безкоштовна бібліотека, яка розроблена компанією «Google», та спочатку була використовувались лише в середині компанії. Після того, як кампанія «Google» зробила її публічною, вона стала досить популярною в колі науковців. Дана бібліотека добре підходить для роботи із машинним навчанням на проведення обчислень. Вона дозволяє реалізовувати різні типи глибоких нейронних мереж та моделей, та включає функції для роботи із ними.

– *OpenCV*. Це безкоштовна та відкрита бібліотека роботи із задачами комп'ютерного зору та містить велику кількість функцій для обробки зображень (фільтрація, виявлення країв, розпізнавання образів, тощо). Вона широко підтримується та вдосконалюється спільнотою.

– *Keras*. Найпопулярніша бібліотека з відкритим кодом для глибокого навчання, яка має велику спільноту користувачів та чітку документацію. «Keras» може використовуватись для різних завдань у галузі машинного навчання, включаючи класифікацію зображень. Дана бібліотека підтримує усі три архітектури

згорткових нейронних мереж які використані у дослідженні, та сильно спрощує процес розробки.

– *Scikit-learn*. Бібліотека для мови програмування Python, яка включає в себе великий набір інструментів для роботи у сфері машинного навчання та аналізу даних. Вона також включає в себе вбудовані набори даних, які можна використовувати для навчання та тестування моделей, також включає в себе інструменти оптимізації параметрів, включаючи пошук гіперпараметрів та оцінку їх впливу на ефективність моделі. Вона містить різні метрики для оцінки ефективності, а в ході дослідження для отримання таких метрик як : accuracy, recall, precision, f1-score, AUC.

– *NumPy*. Досить популярна бібліотека, що використовується для різних видів обчислень, та підтримує різні структури даних, а головне роботу із багатовимірними масивами. Її перевагою є швидкість обчислення великих наборів вхідних даних, та вона легко інтегрується для роботи із іншими бібліотеками для мови «Python», наприклад – «Matplotlib».

– *Matplotlib*. Потужна бібліотека, яка створена для візуалізації даних у вигляді графіків та діаграм різноманітних типів. Є можливість створення інтерактивних графіків, що дозволяє користувачеві взаємодіяти із ним, як до прикладу, виділяти точку на графіку та бачити її значення. Дана бібліотека надає багато можливостей із налаштування розмірів, кольорів, написів на кінцевих графіках та зберігати їх у різних форматах.

– *TKinter*. Стандартна бібліотека «Python» для створення графічного інтерфейсу користувача (GUI), яка не є новою, та порівнянно із сучасними бібліотеками для роботи із графічними інтерфейсами є дуже простою у використанні. Вона включає всі необхідні функції які були потрібні нам для розробки прикладного додатку, а саме: робота із файловою системою, відображення зображень, створення простих компонентів для взаємодії із додатком (кнопки, поля для вводу, тощо).

3.2 Проектування прикладних компонентів застосування

Основним програмним засобом для реалізації вдосконалення запропонованого методу може слугувати блокнот у середовищі «Jupyter Notebook» з використанням мови програмування «Python». Код реалізації методу треба розділити на комірки та 2 етапи.

Перший етап слід виконувати у такій послідовності:

- 1) підключення усіх необхідних бібліотек;
- 2) налаштування та підготовка до навчання запропонованих архітектур глибоких нейронних мереж;
- 3) підключення та навчання глибоких нейронних мереж з нуля на початкових вхідних даних;

Другий етап складається із таких кроків:

- 1) донавчання моделей на підготовленому наборі даних;
- 2) розпізнавання обличчя за зображенням під довільним кутом зору із використанням тестового набору даних;
- 3) збереження та аналіз результатів;

Другий етап виокремлений, адже він є циклічним, та повинен повторюватись декілька разів для різних мереж та різних вхідних даних.

Для підготовки вхідних даних для донавчання, слід розробити два 2 додатки, що будуть працювати із зміною освітленості на зображеннях, які мають відповідати таким вимогам:

- наявність графічного інтерфейсу;
- простота у використанні;
- відображення змін до збереження результатів.

Перший додаток, використовується для перегляду зображення та зміни його налаштувань яскравості та контрастності. Він має графічний інтерфейс, а зміни відображаються у режимі реального часу. Користувач має змогу бачити на які

значення змінюються параметри яскравості та контрастності, а також має змогу зберегти змінене зображення. Діаграма активності додатку для зміни зображення представлена на рисунку 3.1.

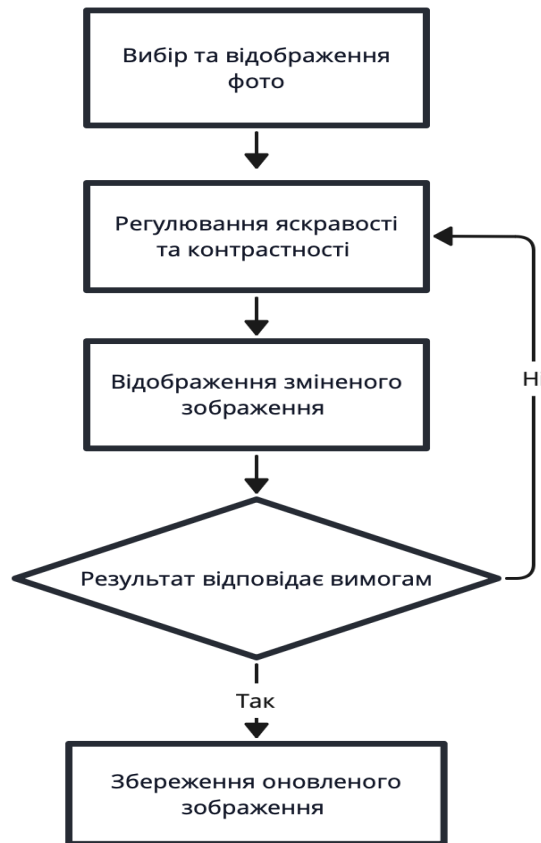


Рисунок 3.1 – Діаграма активності для першого додатку

Другим додатком, слугує програма із графічним інтерфейсом, яка дозволяє вказати шлях до теки із зображеннями, вказати значення параметрів яскравості та контрастності і застосувати їх, а також зберегти оновлені зображення. Діаграма активності представлена на рисунку 3.2

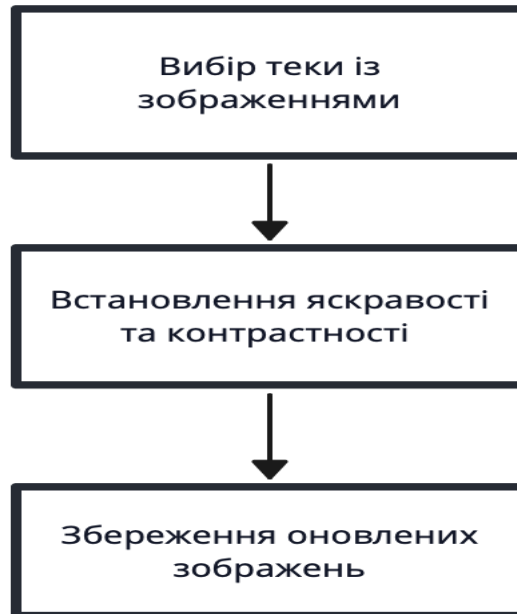


Рисунок 3.2 – Діаграма активності для другого додатку

Висновки до розділу 3

Третій розділ містить опис проекту реалізації вдосконаленого методу розпізнавання обличчя під довільним кутом зору. Наведено бібліотеки, фреймворки, які добре підходять для реалізації відповідного методу та супроводжуючих застосунків для підготовки вхідних даних.

Наведено опис та вимоги до застосувань необхідних для підготовки вхідних даних для навчання моделей та розпізнавання обличчя за зображенням. Представлено діаграми активностей для двох додатків, що будуть використовуватись для підготовки вхідних даних.

РОЗДІЛ 4

Дослідження методу розпізнавання обличчя за зображенням під довільним кутом зору

З метою дослідження вдосконаленого методу розпізнавання обличчя за довільним кутом зору, створене прототипне застосування для проведення експериментів та дослідження якості класифікаторів. Далі викладено отримані результати та проведено їх аналіз:

- опис прототипного застосування для дослідження запропонованого вдосконаленого методу розпізнавання обличчя за довільним кутом зору;
- підготовка вхідних даних за допомогою розробленого програмного продукту, задля зміни освітлення на зображенні;
- проведення експериментів, в ході яких було навчання та донавчання мережі, перевірки на тестових даних та аналіз отриманих результатів.

4.1 Прототипне застосування для дослідження запропонованого вдосконаленого методу розпізнавання обличчя за довільним кутом зору

Для проведення експериментів, використовувалось середовище «Jupyter Notebook» яке працює із мовою програмування Python. Дане середовище являє собою зручний веб інтерфейс, в якому розташовані комірки, в яких можна зручно розташовувати код, опис, здійснювати підключення різних бібліотек, а також одразу бачити результат виконання. Найважливішими для дослідження є бібліотеки для роботи із згортковими нейронними мережами, що широко представлені у відкритому доступі.

Середовище «Jupyter Notebook» зручне ще тим, що кожен метод можна використовувати у окремій комірці, що дозволяє за потреби швидко вносити зміни, або порівнювати декілька схожих функцій.

Дослідження у середовищі «Jupyter Notebook», можна розділити на декілька кроків, наведених нижче.

- Підключення усіх необхідних бібліотек для роботи із згортковими нейронними мережами, виконання обчислень, та виводу результатів у вигляді таблиць та графіків.

- Визначення усіх функцій, які будуть використані під час експерименту (завантаження вхідних даних, тренування моделей, обчислення, тощо).

- Навчання моделі з нуля на наборі даних який включає в себе велику кількість зображень облич («VGGFace”).

- Три ітерації донавчання та тестування на трьох згорткових нейромережах

- Отримання результатів експериментів та їх аналіз.

Для підготовки даних до навчання створено 2 десктопних застосування також на мові програмування «Python».

Перший додаток, головне вікно якого зображене на рисунку 3.1, використовується для відображення у режимі реального часу змінених параметрів яскравості та контрастності для обраного зображення.

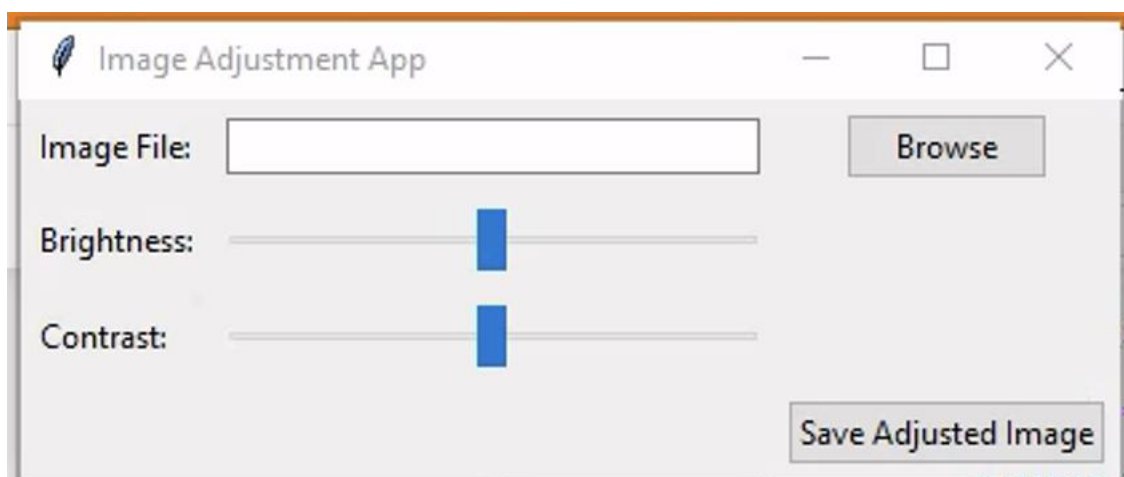


Рисунок 3.1 – Головний інтерфейс застосування для відображення змін в освітленості зображення

Даний додаток містить поле для вводу, в яке можна скопіювати шлях до зображення, або ж можна скористатись кнопкою «Browse» для вибору шляху за допомогою файлового менеджера (рис. 3.2).

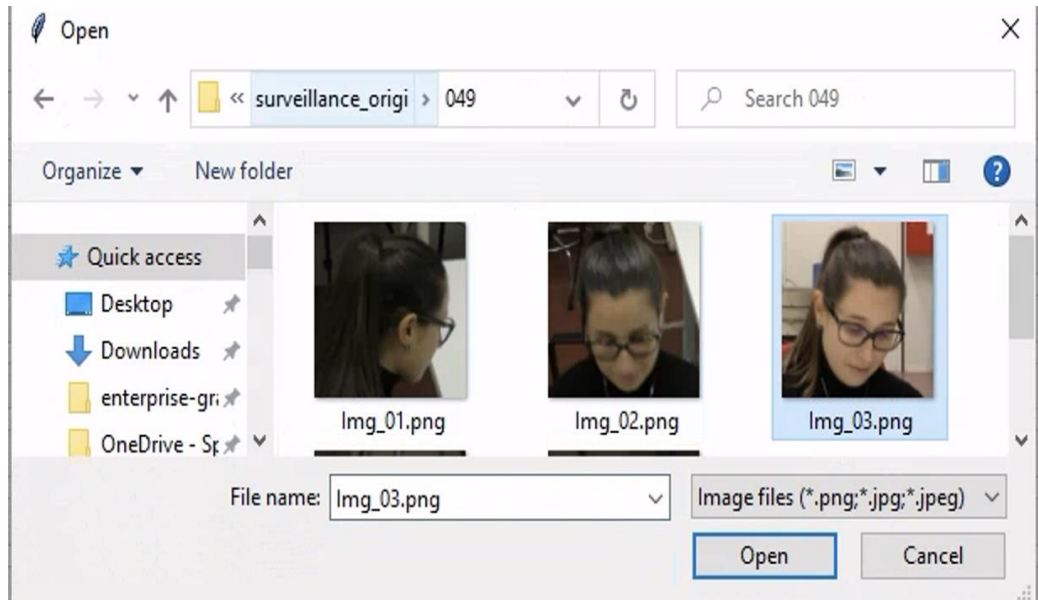


Рисунок 3.2 – Інтерфейс файлового менеджера для вибору зображення.

Після вибору зображення, та натискання кнопки «Open», у полі шлях до зображення відобразиться місце розташування обраного зображення, та ми зможемо побачити саме зображення (рис. 3.3).

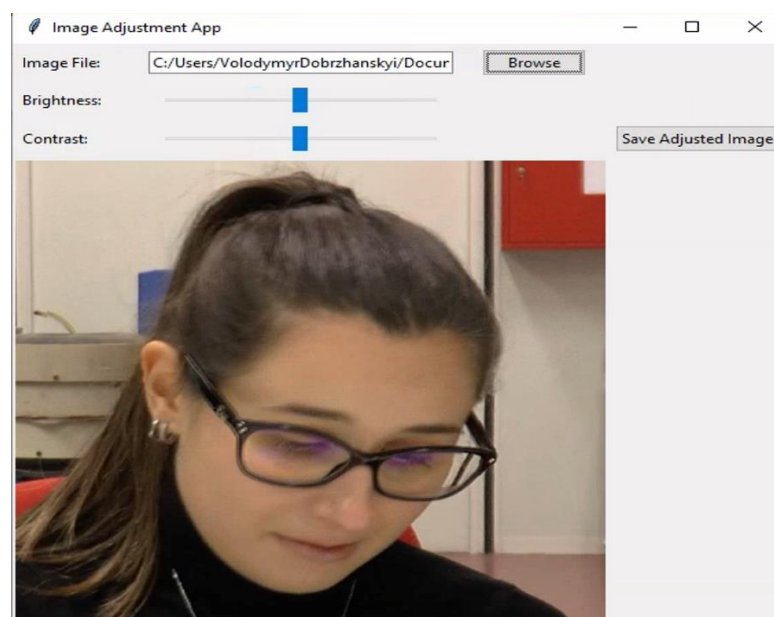


Рисунок 3.3 – Інтерфейс застосунку із відображенням фото.

Використовуючи слайдер яскравості «Brightness» та контрастності «Contrast», зображення змінюється у режимі реального часу, та справа під кнопкою «Browse» відображаються значення які були встановлені користувачем (рис. 3.4). Цей додаток використовувався для зміни освітлення лише для фронтальних зображень обличчя.

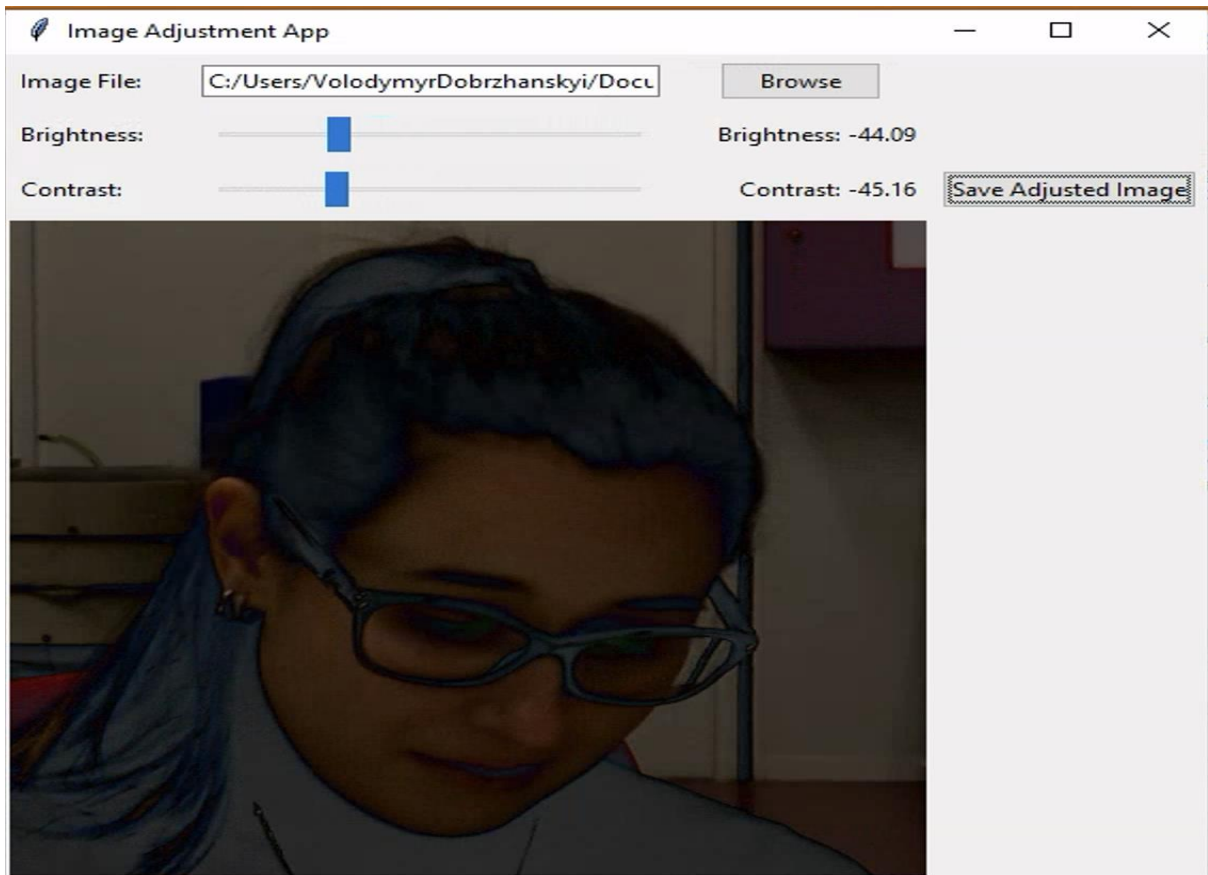


Рисунок 3.4 – Зміна освітленості на зображенні.

Для збереження отриманого файлу, використовується кнопка «Save adjusted Image». Новий файл зберігається в ту саму теку, де знаходився вхідний, і використовує його ім'я з додавання префіксу «adjusted_».

Другий додаток, графічний інтерфейс якого зображено на рисунку 3.5, має схожий функціонал, але вхідним параметром для нього слугує не зображення, а тека в якій зображення знаходяться.

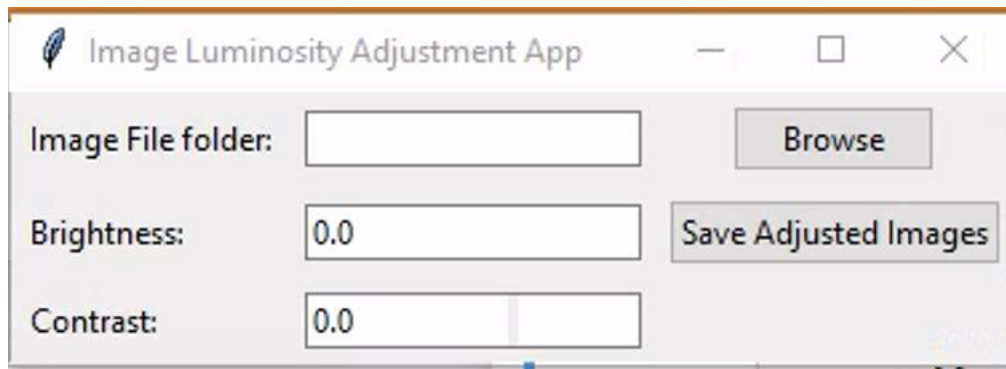


Рисунок 3.5 – Додаток для зміни освітлення усіх зображень у теці

Використовуючи поля для вводу яскравості «Brightness» та контрастності «Contrast», є можливість встановити як від’ємні так і позитивні значення, та після натискання на кнопку збереження «Save Adjusted Images», усі фото у теці обробляються і зберігаються в тій самій директорії із додавання до назви файлу префіксу «adjusted_» та значень які були встановлені.

По завершенню формування додаткових вхідних даних за допомогою розроблених додатків, відбувається перехід до процесу навчання та тестування трьох архітектур мереж у блокноті «Jupyter Notebook». В результаті обчислень та візуалізації результатів, отримано набір pdf файлів, які містять в собі показники ефективності та графіки ROC кривої та точності розпізнавання. Код блокноту та розроблених програм наведений у додатках.

4.2 Підготовка вхідних даних

Для проведення експериментів обрано три набори даних, для навчання та тестування, які були додатково модифіковані:

- набір даних «VGGFace» [37], для початкового навчання моделей з нуля;
- оригінальний датасет «FRMDB» [25], використовується для навчання;
- створений нами датасет, що отримав назву «FRMDB_Modified_Front» який включає у себе додаткові фото із надмірною та недостатньою яскравістю для фронтальних фото, використовується для донавчання;

– створений датасет «FRMDB_Modified_All_Angle», включає у себе додаткові фото із надмірною та недостатньою яскравістю для фронтальних фото, використовується для донавчання;

– датасет «SCFace» [47] в якому за допомогою розробленого програмного засобу змінили освітленість, використовується для тестування.

У таблиці 4.1 описані набори даних які використовувались у ході експериментів.

Таблиця 4.1 – Детальний опис використаних наборів даних

Datasets	Samples	Face Images	Not Face Images
		No. of Images	No. of Images
VGGFace	2.6 million	2.4 million	200 000
FRMDBoriginal	1442	1092	350
FRMDB front	100	78	22
FRMDB all angle	2384	2184	200
SCFaces original(video)	230	-	-
SCFaces updated	450	390	60

Оригінальний набір даних «FRMDB» [25] включає в себе 39 наборів по 28 знімків обличчя чоловіків та жінок, зроблених під різними кутами, та 350 фото на яких немає обличчя. Фото зроблені з 7 ракурсів у горизонтальній площині з кроком у 45 градусів та 4 ракурсів у вертикальній площині із кроком у 30 градусів. Усі фото мають розширення *.png та роздільну здатність 972 на 544 пікселі. Приклад набору даних зображено на рис. 4.2.



Рисунок 4.2 – Приклад зображень з набору даних «FRMDB»

Оригінальний набір даних «SCFace» [47] містить 230 відео, які взяті із камер відеоспостереження. На 195 із них присутні обличчя, під довільним кутом, та 35 де обличчя відсутнє. Якість відео дуже посередня, оскільки присутнє різне освітлення та роздільна здатність лише 352 на 288 пікселів, що є хорошим набором даних для тестування ефективності методу. На основі цього датасету, за допомогою розробленого застосунку кількість зразків збільшена до 450 зображень. Приклад набору даних зображено на рис. 4.3.



Рисунок 4.3 – Приклад зображень з набору даних «SCFace»

Для проведення експерименту, розроблено програмний продукт, детально описаний вище, та за його допомогою підготовлено додаткові вхідні дані для навчання мережі.

Спочатку, для кожної особи із набору даних «FRMDB» обрали фронтальне фото, та на його основі створили 2 додаткових зображення обличчя із надмірною та недостатньою освітленістю. Приклад набору даних зображено на рис. 4.4.

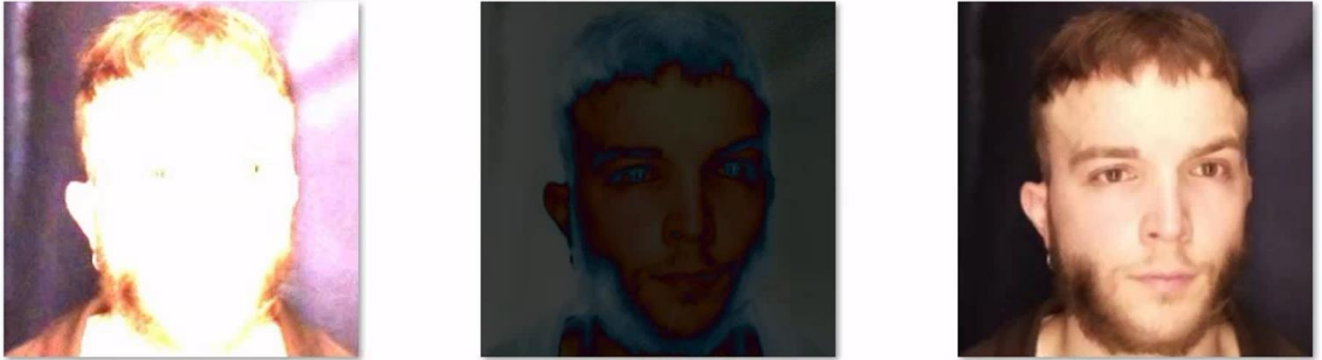


Рисунок 4.4 – Приклад створених додаткових зображень для фронтального фото із датасету «FRMDB»

Наступним кроком були взяті усі фото для кожної особи із набору даних «FRMDB» та аналогічним чином, створені зображення обличчя із надмірною та недостатньою освітленістю. Приклад набору даних зображено на рис. 4.5.



Рисунок 4.5 – Приклад створених додаткових зображень для усіх фото із датасету «FRMDB»

Для проведення тестів, із відеозаписів, які містились у наборі «SCFaces», обрано 5 кадрів із 5 відео для кожної людини, де вони розташовані у довільній позі та під різним ракурсом, та на основі них створено нові зображення. За допомогою програмного застосунку, ці зображені обрізали з метою відкидання лишніх об'єктів,

фокусуванні саме на обличчя та зменшені до розміру 972 на 544 пікселі, щоб вони відповідали розмірам зображень, що використовувались під час навчання. Приклад набору даних зображено на рис. 4.6.



Рисунок 4.6 – Приклад модифікації зображень з датасету «SCFaces»

Наступним етапом є використання розробленого програмного засобу для створення тестового набору даних «SCFaces updated», що включає в себе зображення обличчя під довільним кутом зору в умовах недостатньої чи надмірної освітленості.

Результатом стала вибірка 390 фото, яка доповнена зображеннями які не містили обличчя осіб із оригінального набору «FRMDB». Приклад набору даних зображено на рис. 4.7.

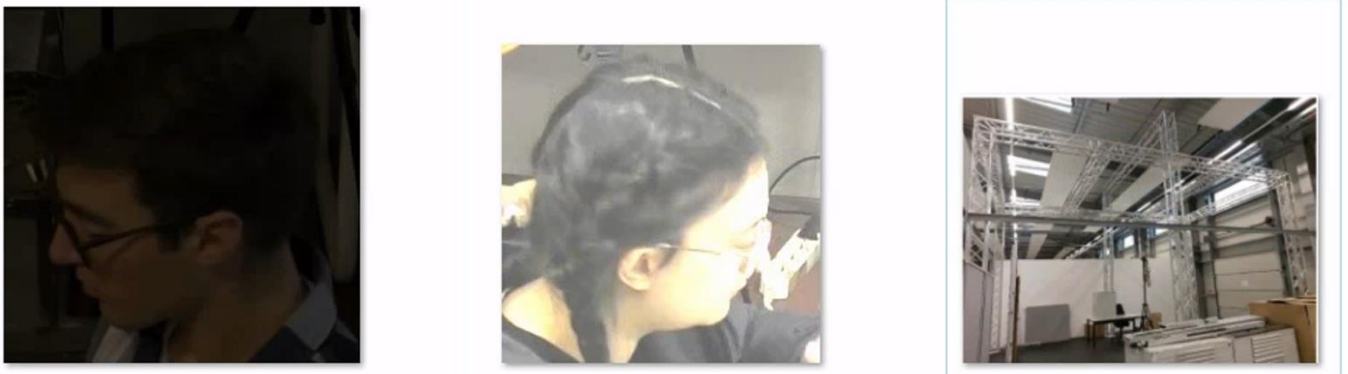


Рисунок 4.7 – Приклад зображень зі створеного для тестування набору даних

4.3 Результати експериментального дослідження вдосконаленого методу

Для оцінки якості роботи запропонованого вдосконаленого методу, проведено тести на кожній із трьох глибоких мереж: «VGG-16», «ResNet-50», «SeNet50».

Оцінка якості роботи вдосконаленого методу базується на визначенні загальної точності, параметрів recall, precision, f1-score та побудові ROC-кривої.

Перший тест являє собою спробу розпізнати обличчя за зображеннями з довільним кутом зору в умовах недостатньої чи надмірної освітленості, що створені за допомогою програмного застосування за відео із датасету «SCFaces». Попередньо, згорткові мережі навчалися з нуля на наборі даних «VGGFaces», та після донавчалась на датасеті «FRMDB original».

Результати першого тесту для кожної із використаних в ході експерименту мережі наведені нижче.

Згорткова мережа «VGG» з архітектурою «VGG-16» показала точність 59,2% (рис. 4.8), а у таблиці 4.2 наведено інші показники якості.

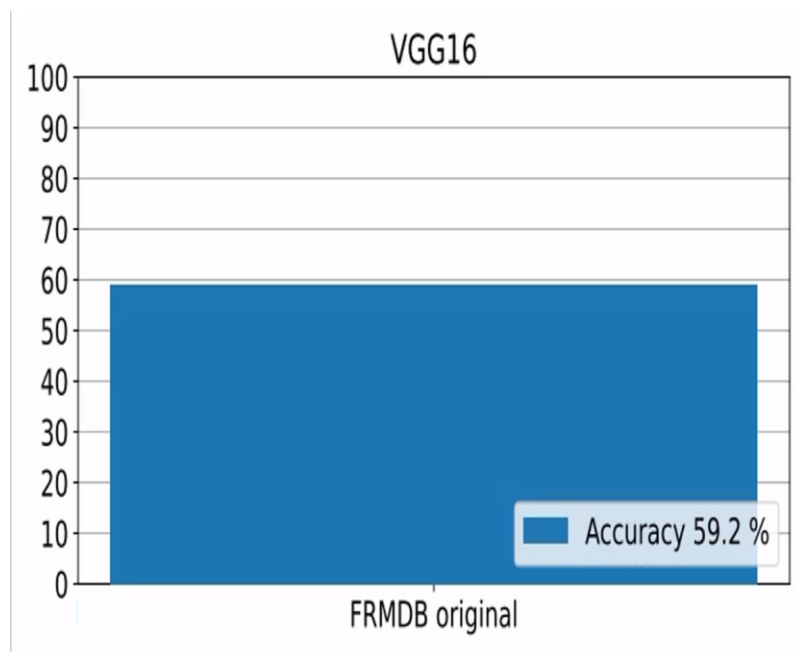


Рисунок 4.8 – Точність «VGG-16» для першого тесту

Таблиця 4.2 – Показники якості «VGG – 16» для першого тесту.

VGG-16	Test 1
Recall	0.65
Precision	0.72
F1-Score	0.69

Графік ROC кривої зображено на рисунку 4.9, площа під кривою (AUC) = 0.6443. Значення є більшим від 0.5, що вказує що класифікатор є певним чином навчений, але не достатньо для задачі, що розглядається. Це видно і за значеннями Recall, Precision та F1-Score.

VGG-16 FRMDb original (AUC=0.6443)

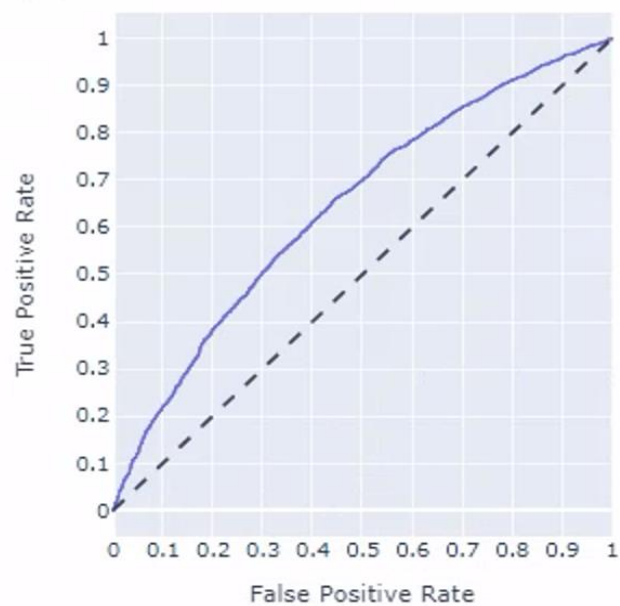


Рисунок 4.9 –ROC-крива для мережі «VGG -16» для першого тесту.

Згорткова мережа «SeNet» з архітектурою «SeNet-50» показала точність 61,2 % (рис 4.10), а у таблиці 4.3 наведено інші показники якості.

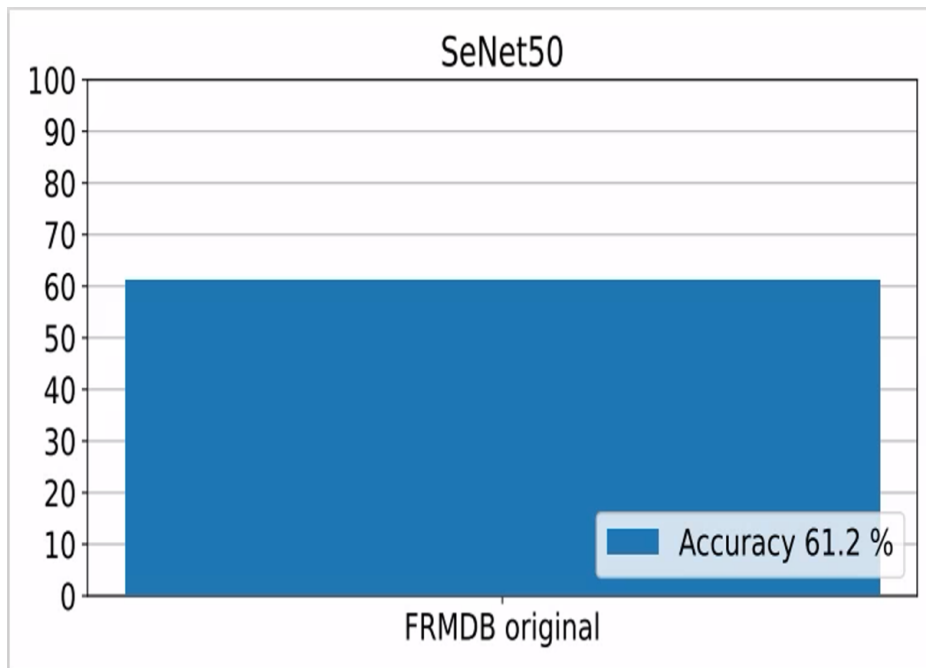


Рисунок 4.10 – Точність «SeNet-50» для першого тесту

Таблиця 4.3 – Показники якості «SeNet -50» для першого тесту.

Senet-50	Test 1
Recall	0.67
Precision	0.79
F1-Score	0.73

Графік ROC кривої зображено на рисунку 4.11. Площа під кривою (AUC) = 0.7004. Значення є більшим від 0.5, що вказує що класифікатор є певним чином навчений, крім того значення більше чим у попередньому експерименті, але не достатньо для задачі, що розглядається. Це видно і за значеннями Recall, Precision та F1-Score.

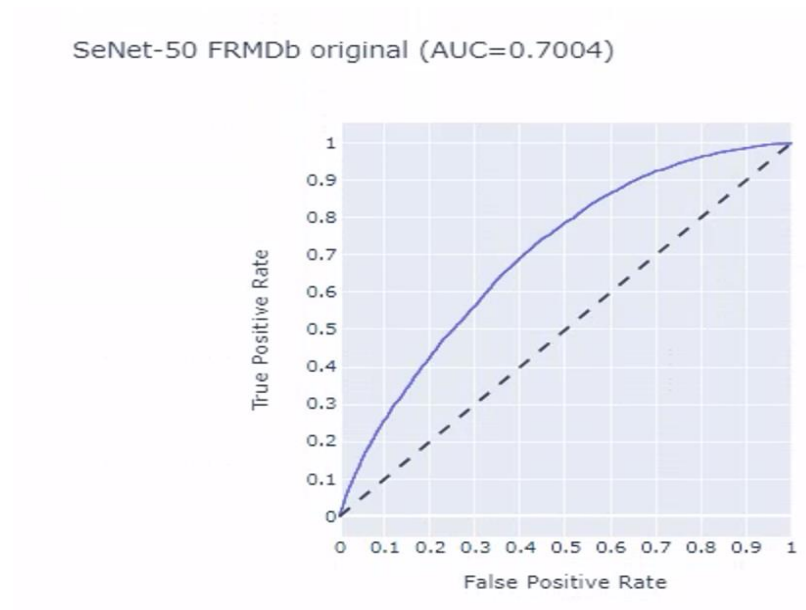


Рисунок 4.11 – Графік ROC кривою для мережі «SeNet-50» для першого тесту.

Згорткова мережа «ResNet» з архітектурою «ResNet-50» показала точність 68,7% (рис 4.12), а у таблиці 4.4 наведено інші показники якості.

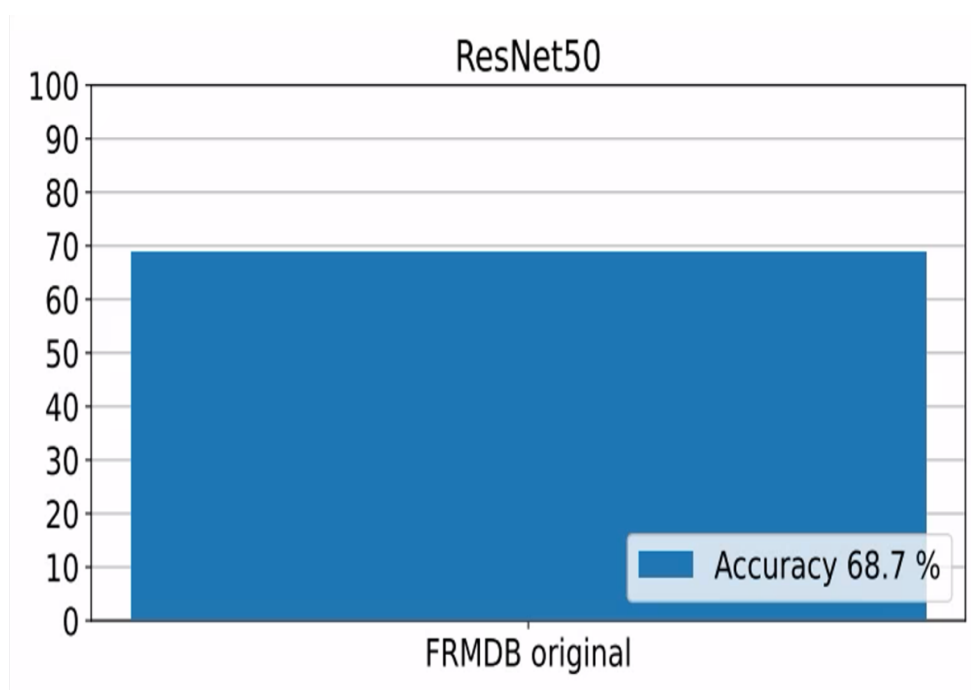


Рисунок 4.12 – Точність «ResNet-50» для першого тесту

Таблиця 4.4 – Показники якості «ResNet – 50» для першого тесту.

ResNet-50	Test 1
Recall	0.67
Precision	0.77
F1-Score	0.72

Графік ROC кривої зображено на рисунку 4.13. Площа під кривою (AUC) = 0.7515. Значення є більшим від 0.5, що вказує що класифікатор є певним чином навчений, крім того значення більше чим у попередніх експериментах, але не є ідеальним (близьким до 1) для задачі, що розглядається. Це видно і за значеннями Recall, Precision та F1-Score.

ResNet-50 FRMDB original (AUC=0.7515)

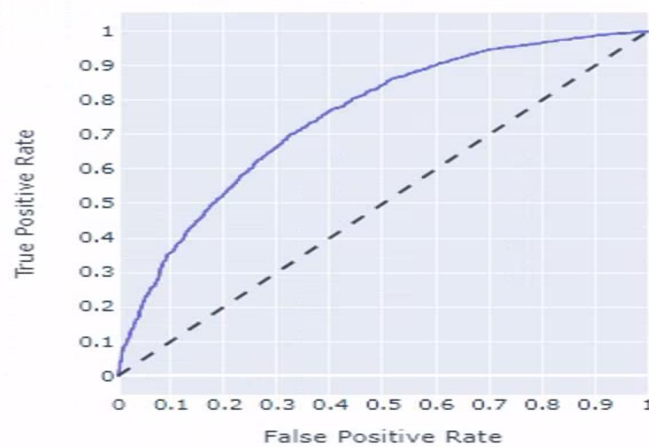


Рисунок 4.13 – Графік ROC кривою для мережі «ResNet-50» для першого тесту.

Порівнюючи результати точності (рис. 4.14) першого тесту та інших показників, що наведені у таблиці 4.5, видно, що мережа «ResNet» справилась із тестом найкраще, але на жаль, жодна мережа не показала достойних результатів для розпізнавання обличчя під довільним кутом зору, в умовах недостатньої чи надмірної освітленості. На нашу думку, це спричинено тим, що тестова вибірка, яка складається із фото із сильно зміненим освітленням, не дає отримати мережі достатню кількість

параметрів та характеристик для вірної класифікації зображень. Отримані результати можна описати, як кращі ніж довільне вгадування, але недостатні для сталого промислового використання.

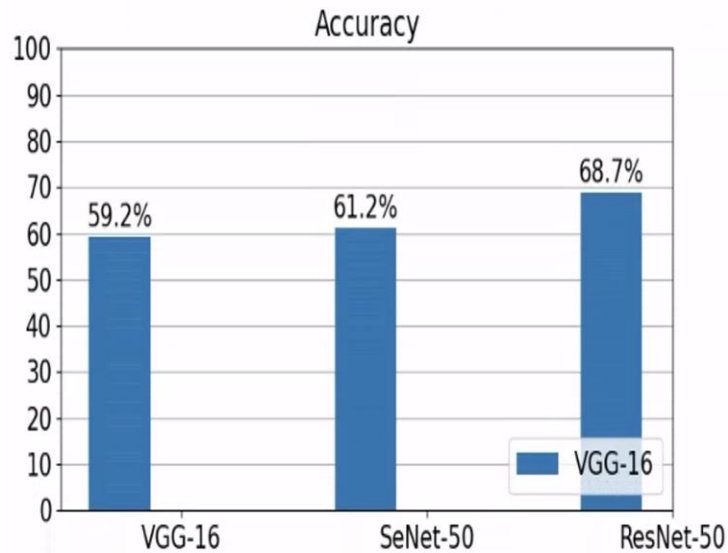


Рисунок 4.14 – Порівняння точності усіх мереж для першого тесту

Таблиця 4.5 – Порівняльна таблиця результатів першого тесту

Test 1	Precision	Recall	F1 - score	AUC
VGG-16	0,72	0,65	0,69	0,6443
SeNet-50	0,79	0,67	0,73	0,7004
ResNet-50	0,77	0,67	0,72	0,7515

Другий тест полягав у спробі розпізнати обличчя за зображеннями з довільним кутом зору в умовах недостатньої чи надмірної освітленості, із донавчанням мережі. Донавчання здійснювалось за допомогою набору даних, що створений, на основі фронтальних фото кожної із особи у початкову датасеті «FRMDB». За допомогою розробленого програмного забезпечення, фото оброблялись, з метою імітації перенасиченого та недостатнього освітлення.

Результати другого тесту для кожної із використаних в ході експерименту мережі представлені нижче.

Згорткова мережа «VGG» з архітектурою «VGG-16» показала точність 81,6% (рис 4.15), а у таблиці 4.6 наведено інші показники ефективності.

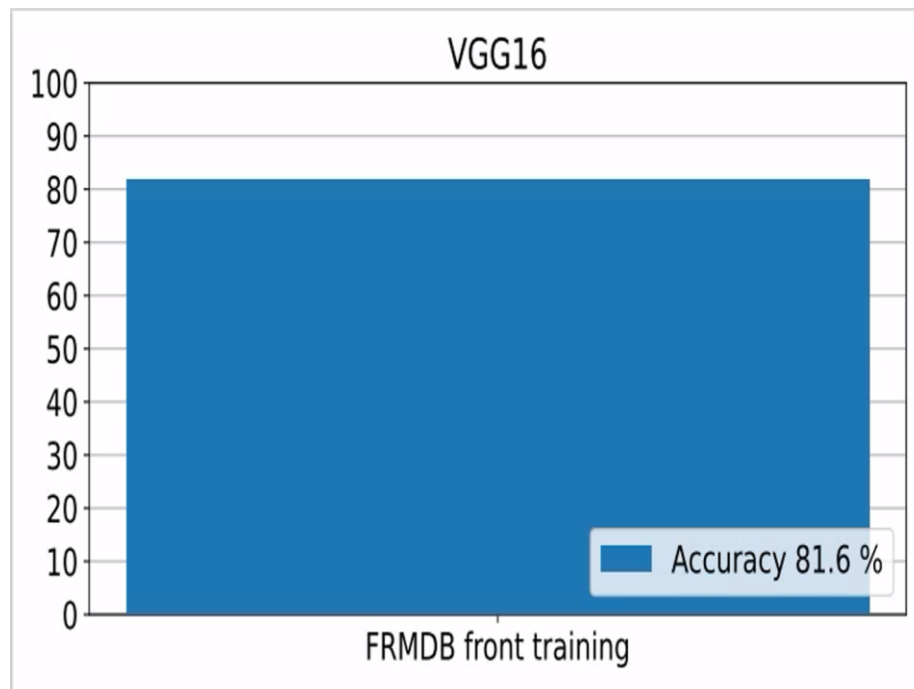


Рисунок 4.15 – Точність «VGG-16» для другого тесту

Таблиця 4.6 – Показники якості «VGG – 16» для другого тесту.

VGG-16	Test 2
Recall	0.78
Precision	0.75
F1-Score	0.76

Графік ROC кривої зображено на рисунку 4.16. Площа під кривою (AUC) = 0.7718. Значення є більшим від 0.5, що вказує що класифікатор є певним чином навчений, крім того значення більше чим у попередніх експериментах, але не є ідеальним (близьким до 1) для задачі, що розглядається. Це видно і за значеннями Recall, Precision та F1-Score.

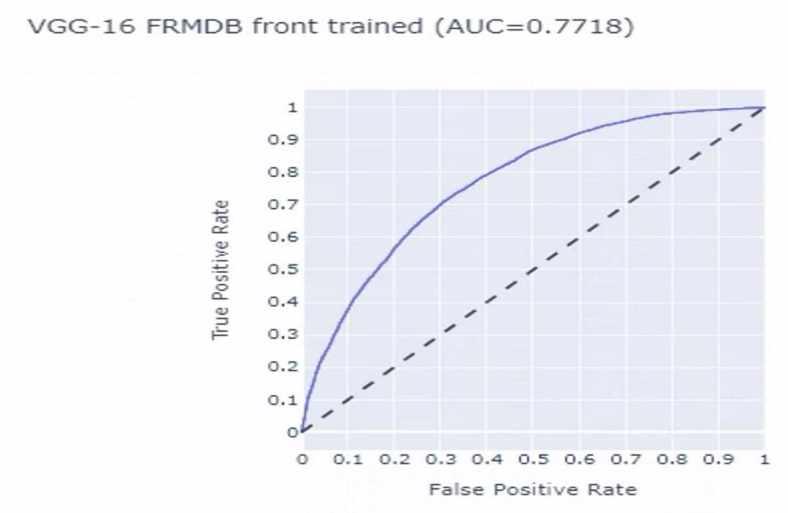


Рисунок 4.16 – Графік ROC кривою для мережі «VGG-16» для другого тесту.

Згорткова мережа «SeNet» з архітектурою «SeNet-50» показала точність 84,9% (рис 4.17), а у таблиця 4.7 наведено інші показники якості.

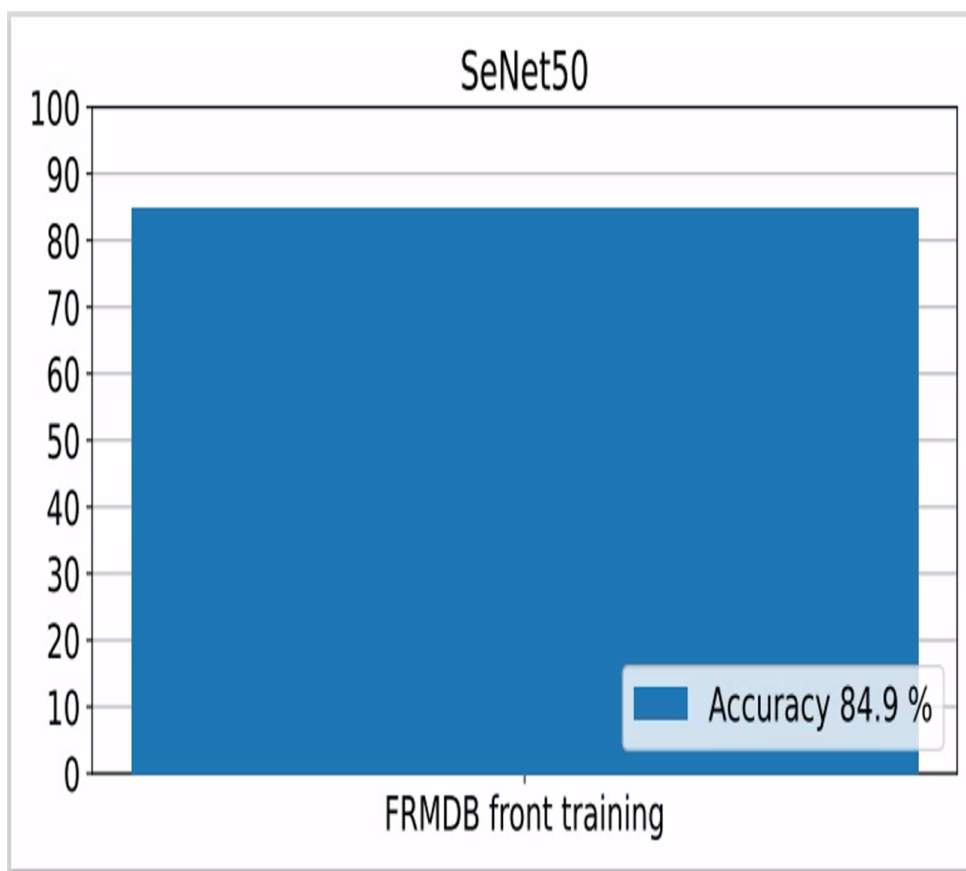


Рисунок 4.17 – Точність «SeNet-50» для другого тесту

Таблиця 4.7 – Показники якості «SeNet -50» для другого тесту.

Senet-50	Test 2
Recall	0.75
Precision	0.8
F1-Score	0.77

Графік ROC кривої представлений на рисунку 4.18. Площа під кривою (AUC) = 0.8230. Значення є вже значно більшим від 0.5, що вказує що класифікатор є навчений, значення більше чим у попередніх експериментах, але також не є ідеальним (близьким до 1) для задачі, що розглядається. Це видно і за значеннями Recall, Precision та F1-Score.

SeNet-50 FRMDB front trained (AUC=0.8230)

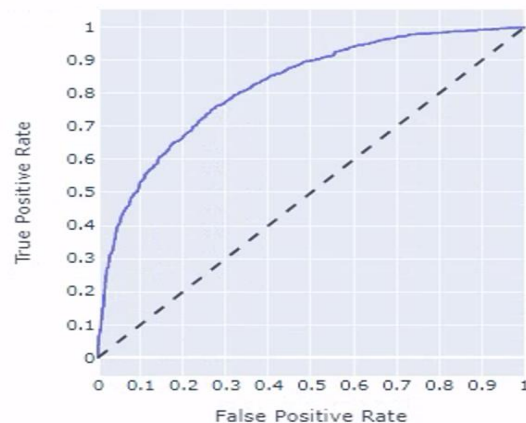


Рисунок 4.18 – Графік ROC кривої для мережі «SeNet-50» для другого тесту.

Згорткова мережа «ResNet» з архітектурою «ResNet-50» показала точність 86,1% (рис 4.19). У таблиці 4.8 наведені інші показники якості.

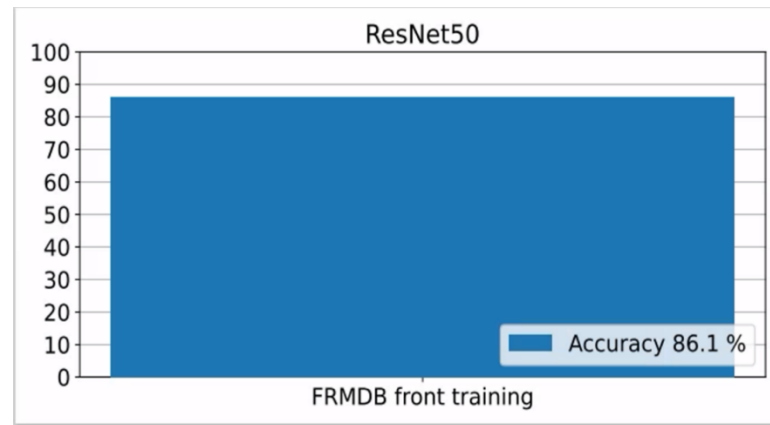


Рисунок 4.19 – Точність «ResNet-50» для другого тесту

Таблиця 4.8 – Показники якості «VGG – 16» для першого тесту.

ResNet - 50	Test 2
Recall	0.79
Precision	0.82
F1-Score	0.8

Графік ROC кривої зображено на рисунку 4.20. Площа під кривою (AUC) = 0.8479. Значення вже є ближчим до 1, що вказує на задовільну навченість класифікатора, значення більше чим у попередніх експериментах, але також не є ідеальним (близьким до 1) для задачі, що розглядається. Це видно і за значеннями Recall, Precision та F1-Score.

SeNet-50 FRMDB front trained (AUC=0.8479)

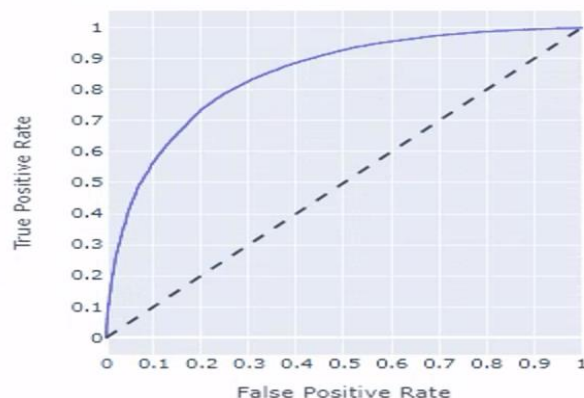


Рисунок 4.20 – Графік ROC кривою для мережі «ResNet-50».

Порівнюючи результати другого тесту, а саме точність (рис. 4.21), а також інші показники, які наведені у таблиці 4.9, можемо бачити, що мережа «ResNet-50», знову показала найкращі результати. В порівнянні із попереднім тестом, усі мережі значно покращили свої показники.

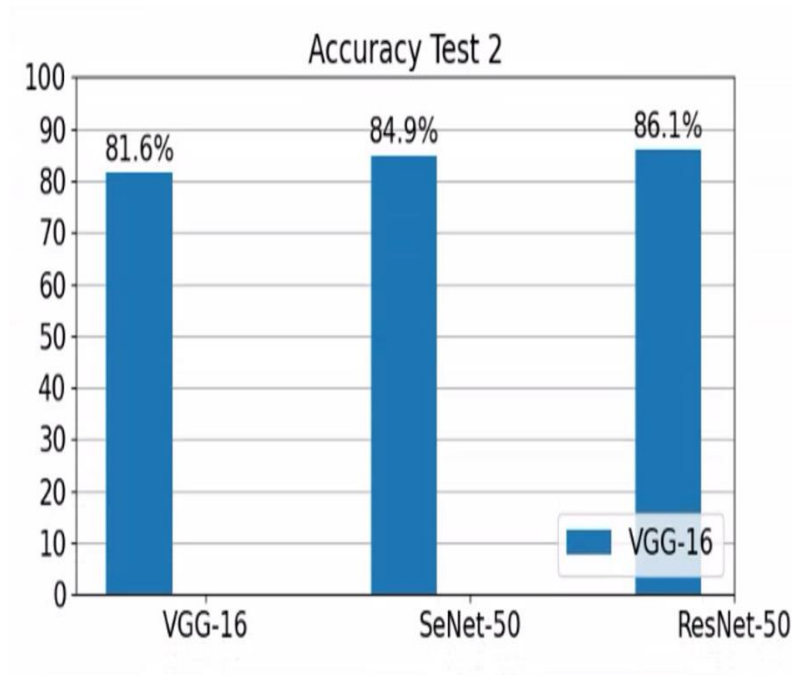


Рисунок 4.21 – Порівняння точності для другого тесту

Таблиця 4.9 – Порівняльна таблиця результатів другого тесту

Test 2	Precision	Recall	F1 - score	AUC
VGG-16	0,75	0,78	0,76	0,7718
SeNet-50	0,8	0,75	0,77	0,8230
ResNet-50	0,82	0,79	0,8	0,8479

Отже, показники покращились, за рахунок того, що вхідні дані, які використовувались для донавчання, а саме фронтальні фото із зміненим освітленням, дозволили неймережі ефективно опрацювати ключові елементи зображені на них. Тестовий набір даних, в більшості включав у себе зображення обличчя, де більшість його основних елементів, таких як ніс, були присутні.

Незважаючи на явні покращення у розпізнаванні обличчя під довільним кутом зору, жодна із мереж не є достатньо якісною.

Третій тест полягав у спробі розпізнати обличчя за зображеннями з довільним кутом зору в умовах недостатньої чи надмірної освітленості, із донавчанням мережі за допомогою набору даних, аналогічно як і у другому тесті, але не лише фронтального фото, а усіх наявних зображень обличчя у наборі даних «FRMDB».

Результати третього тесту для кожної із використаних в ході експерименту мережі наведені нижче.

Згортова мережа «VGG» з архітектурою «VGG-16» показала точність 93,1% (рис 4.22). У таблиці 4.10 наведено інші показники якості.

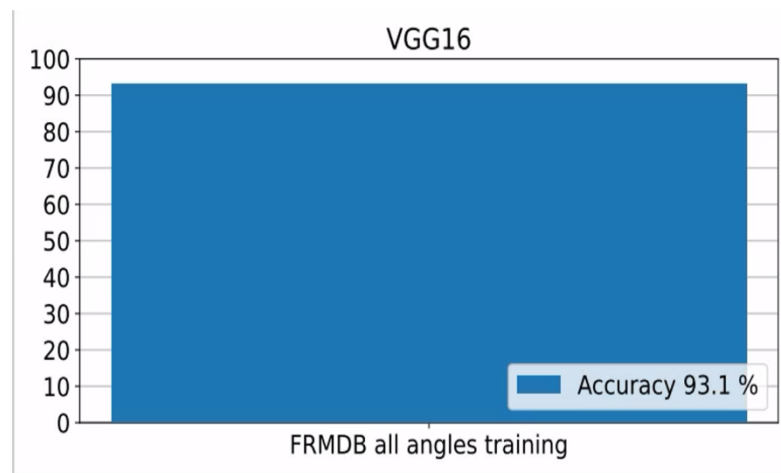


Рисунок 4.22 – Точність «VGG-16» для третього тесту

Таблиця 4.10 – Показники якості «VGG – 16» для другого тесту.

VGG - 16	Test 3
Recall	0.92
Precision	0.91
F1-Score	0.92

Графік ROC кривої зображено на рисунку 4.23. Площа під кривою (AUC) = 0.9173. Значення вже є значно ближчими до 1, що вказує на навченість класифікатора,

значення більше чим у попередніх експериментах, і вже вказують на достатню якість запропонованого методу. Це видно і за значеннями Recall, Precision та F1-Score.

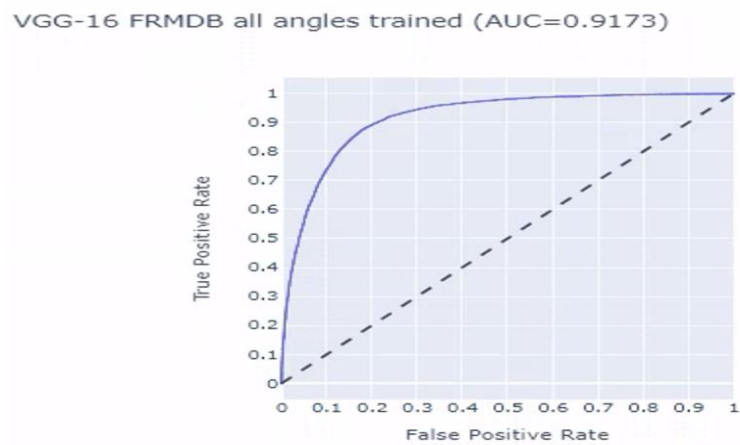


Рисунок 4.23 – Графік ROC кривою для мережі «VGG -16» для третього тесту.

Згорткова мережа «SeNet» з архітектурою «SeNet-50» показала точність 98,7% (рис 4.24), а у таблиці 4.11 наведені інші показники якості.

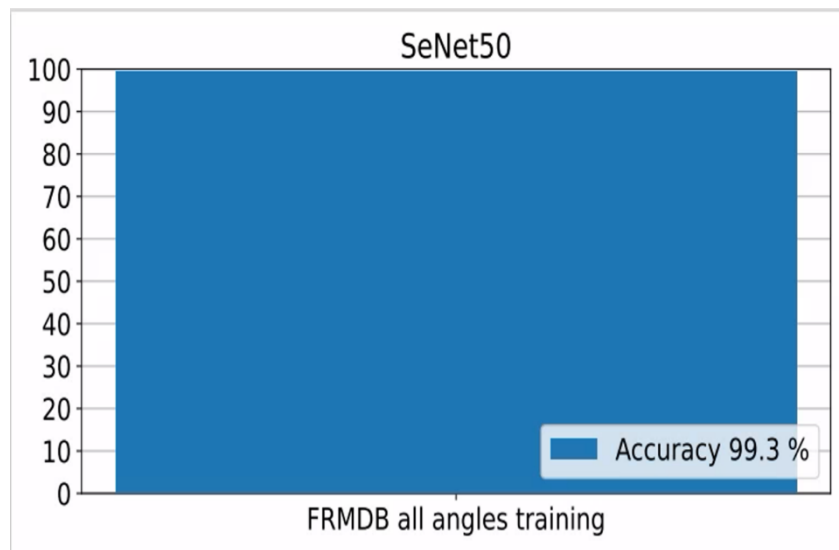


Рисунок 4.24 – Точність «SeNet-50» для третього тесту

Таблиця 4.11 – Показники якості «SeNet -50» для другого тесту.

Senet-50	Test 3
Recall	0.96
Precision	0.94
F1-Score	0.95

Графік ROC кривої зображено на рисунку 4.25. Площа під кривою (AUC) = 0.9710. Значення вже є значно ближчими до 1, що вказує на навченість класифікатора, значення більше чим у попередніх експериментах, і вже вказують на достатню якість запропонованого методу. Це видно і за значеннями Recall, Precision та F1-Score.

SeNet-50 FRMDB all angles trained (AUC=0.9710)

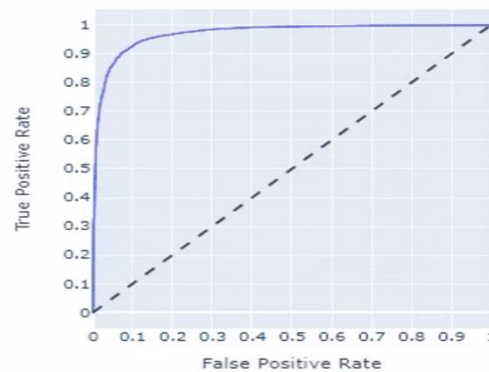


Рисунок 4.25 – Графік ROC кривої для мережі «SeNet-50» для третього тесту.

Згорткова мережа «ResNet» з архітектурою «ResNet-50» показала точність 86,1% (рис 4.26), а у таблиці 4.12 наведено інші показники якості.

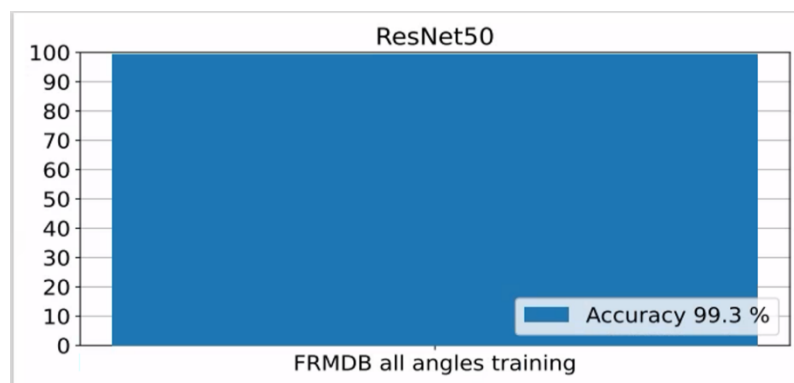


Рисунок 4.26 – Точність «ResNet-50» для третього тесту

Таблиця 4.12 – Показники якості «VGG-16» для третього тесту.

ResNet - 50	Test 3
Recall	0.97
Precision	0.98
F1-Score	0.98

Графік ROC кривої представлений на рисунку 4.27 та площа під кривою (AUC) = 0.9890. Значення вже є достатньо близькими до 1, що вказує на навченість класифікатора, значення більше чим у попередніх експериментах, і вже вказують на достатню якість запропонованого методу. Це видно і за значеннями Recall, Precision та F1-Score.

ResNet-50 FRMDB all angles trained (AUC=0.9890)

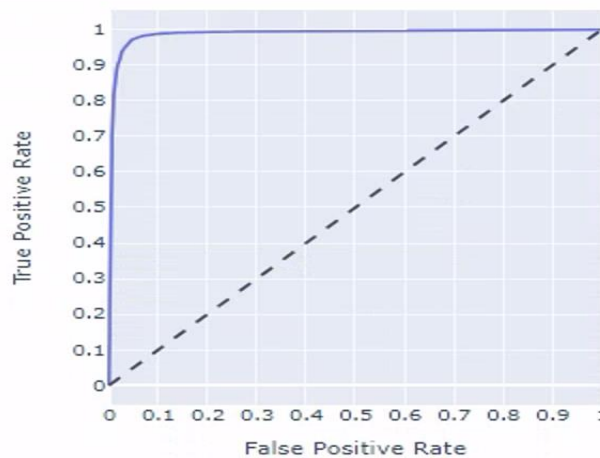


Рисунок 4.27 – Графік ROC кривою для мережі «ResNet-50».

Порівнюючи результати третього тесту, а саме точності (рис. 4.28), а також інших показників, які наведено у таблиці 4.13, можемо побачити, що усі мережі показують результати, по усім показникам більше ніж 0.9.

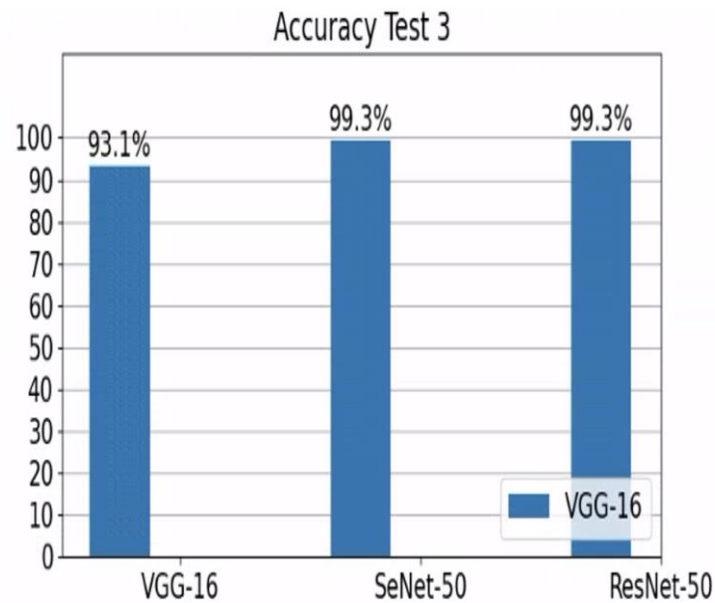


Рисунок 4.28 – Порівняння точності для третього тесту

Таблиця 4.13 – Порівняльна таблиця результатів другого тесту

Test 3	Precision	Recall	F1 - score	AUC
VGG-16	0,91	0,92	0,92	0,9173
SeNet-50	0,94	0,96	0,95	0,9710
ResNet-50	0,98	0,97	0,98	0,9890

Отже, показники покращились, за рахунок того, що вхідні дані, які використовувались для донавчання, а саме усі фото із зміненим освітленням, дозволили неймережі опрацювати набагато більше ключових елементів на фото, адже вхідні дані містили зображення, де обличчя було майже не видно.

Нажаль, мережа на основі архітектури «VGG-16», показала найгірший результат. Це зумовлено кількістю її шарів, що значно менша ніж у інших.

Мережі «SeNet-50» та «ResNet-50» показали однакову високу точність у розпізнаванні обличчя під довільним кутом зору в умовах недостатньої чи надмірної освітленості, що більше за 99%, проте «ResNet-50» має також вищі показники recall, precision та відповідно F1-score, що може свідчити про її кращу збалансованість.

Площа під ROC кривою (AUC), у мережі «ResNet-50» також більша, що показує більшу якість класифікатора.

За результатами трьох тестів, можна зробити висновок, що запропонований вдосконалений метод показав найкращу свою ефективність при використанні мереж «ResNet-50» та «SeNet-50». Цілком можливо, що збільшення набору даних для навчання, може ще більше покращувати якість, адже ця тенденція спостерігалась упродовж всього експерименту.

Висновки до розділу 4

У четвертому розділі роботи проведено дослідження методу розпізнавання обличчя за зображенням під довільним кутом зору в умовах надмірної чи недостатньої освітленості. Під час виконання зібрано чотири набори даних: «FRMD», «FRMD_Modified_Front», «FRMD_Modified_All_Angle», та «SecurityCamDB_Modified» які використовувалися для навчання та тестування запропонованого методу. Наведено характеристики (розмір, кількість тощо) зібраних датасетів.

Проведено три тести на кожній із використаних глибоких нейронних мереж з метою визначення загальної точності роботи, показників recall, precision, f1-score, побудови ROC кривої запропонованого методу на відповідних наборах даних.

На основі трьох згорткових нейронних мереж, продемонстрована ефективність роботи покращеного методу.

Зображено графіки точності розпізнавання, ROC кривої та наведено таблиці порівняння показників ефективності.

Загальні висновки

У кваліфікаційній роботі магістра вдосконалено метод розпізнавання обличчя під довільним кутом зору.

В процесі вдосконалення методу виконано наступні задачі:

- проаналізовано предметну область поставленого завдання та існуючі підходи та публікації з метою виявлення актуальності проблеми, яку необхідно вирішити;
- вдосконалено метод розпізнавання обличчя за зображенням під довільним кутом зору в умовах недостатньої чи надмірної освітленості;

Запропонований метод складається із 2 частин: підготовки вхідних даних розробленим програмним засобом та донавчання згорткової мережі отриманим датасетом.

Під час експерименту здійснене донавчання мережі зображеннями, на яких обличчя сильно засвічене, або ледве видно його контури. На нашу думку, процес донавчання можна покращити, якщо використовувати більшу кількість зображень із зміненими освітленням, де воно поступово збільшується від найтемнішого до найсвітлішого.

Розроблено десктоп додаток для підготовки вхідних даних, а для його реалізації обрано мову програмування «Python». Для оцінки якості роботи методу використовувалась інтерактивне оточення «Jupyter Notebook» та відповідні бібліотеки для роботи з неймережевими засобами. Його можна покращити, дозволивши користувачу вибирати не всі зображення із теки, а лише ті які потрібні. Також є можливість розширити функціонал, шляхом надання можливості створювати одразу декілька модифікованих зображень із різними параметрами освітлення, що дозволить зменшити час на підготовку вхідних даних, та тим саме прискорити процес розпізнавання обличчя за зображенням під довільним кутом зору.

Проведено валідацію розробленого методу на тестових наборах даних: де кожна із мереж показала суттєве збільшення показників recall, precision, f1-score, AUC, ассигасу. Узагальнюючим показником ефективності можна вважати точність розпізнавання, де для мережі «VGG-16» точність склала 93,1%, «SeNet-50» та «ResNet-50» показали точність у 99,3%.

Перелік посилань

1. Про функціонування системи фіксації адміністративних правопорушень у сфері забезпечення безпеки дорожнього руху в автоматичному режимі : Постанова Каб. Міністрів України від 10.11.2017 р. № 833 : станом на 19 квіт. 2022 р. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://zakon.rada.gov.ua/laws/show/833-2017>

2. Safe cities: Using smart tech for public security | A Better Connected World | Huawei | BBC Worldwide. BBC - Homepage. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://www.bbc.com/future/bespoke/specials/connected-world/government.html>

3. Штучна нейронна мережа – Вікіпедія. Вікіпедія. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: https://uk.wikipedia.org/wiki/Штучна_нейронна_мережа

4. Deep learning - Wikipedia. Wikipedia, the free encyclopedia. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: https://en.wikipedia.org/wiki/Deep_learning

5. Глибоке навчання і машинне навчання: в чому різниця?. Worldvision. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://worldvision.com.ua/articles/glubokoe-obuchenie-i-mashinnoe-obuchenie-v-chem-raznitsa>

6. Introduction to Convolutional Neural Networks (CNN). Analytics Vidhya. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>

7. Як працюють нейронні мережі? | APEPS department of Igor Sikorsky KPI. Інженерія програмного забезпечення та комп'ютерні науки в КПІ | Кафедра АПЕПС ТЕФ КПІ – програмна інженерія. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <http://apeps.kpi.ua/neural-networks/en>

8. Що таке нейронні мережі та як вони працюють? Класифікація штучних нейромереж - LivingFo. LivingFo. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://livingfo.com/shcho-take-nejronni-merezhi-ta-iaak-vony-pratsiuiut>

9. Хіт сезону – глибоке навчання та все, що про нього треба знати -. TechToday - Про технології в Україні та світі. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://techtoday.in.ua/news/hit-sezonu-gliboke-navchannya-ta-vse-shho-pro-nogo-treba-znati-67620.html>

10. Типи штучних нейронних мереж – Вікіпедія. Вікіпедія. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: https://uk.wikipedia.org/wiki/Типи_штучних_нейронних_мереж

11. Дишкант В. О. Оптимізація та побудова системи генерувань зображень на основі генеративно-змагальних мереж : магістерська робота. Київ, 2022. 94 с. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: https://ai.kpi.ua/ua/masters/thesis/28521smai-dyshkant_magistr.pdf

12. Kamali K. Galaxy Training: Deep Learning (Part 3) - Convolutional neural networks (CNN). Galaxy Training Network. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://training.galaxyproject.org/training-material/topics/statistics/tutorials/CNN/tutorial.html>

13. Мерзлікіна П. М. Застосування методів реідентифікації осіб для аналізу матеріалів систем відеоспостереження : магістерська робота : 6.170101. Київ, 2019. 105 с.

14. How to Use Deep Learning for Face Detection and Recognition Systems?. Hire the World's Most Deeply Vetted Developers & Teams | Turing. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://www.turing.com/kb/using-deep-learning-to-design-face-detection-and-recognition-systems>

15. Camacho C. Convolutional Neural Networks. Machine and deep learning educator. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: https://cezannec.github.io/Convolutional_Neural_Networks

16. Winarno E. Multi-View Faces Detection Using Viola-Jones Method / Journal of Physics: Conference Series. 2019. Т. 1114. С. 012068. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://doi.org/10.1088/1742-6596/1114/1/012068>
17. Johnson D. Back Propagation in Neural Network: Machine Learning Algorithm. Guru99. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://www.guru99.com/backpropogation-neural-network.html>
18. Wang M., Deng W. Deep face recognition: A survey. Neurocomputing. 2020. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://doi.org/10.1016/j.neucom.2020.10.081>
19. Multi-view face pose estimation based on supervised ISA learning / S. Z. Li та ін. Fifth IEEE International Conference on Automatic Face Gesture Recognition, м. Washington, DC, USA. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://doi.org/10.1109/afgr.2002.1004140>
20. You M., Akashi T. Multi-view face detection using frontal face detector. IEEE Transactions on Electrical and Electronic Engineering. 2018. Т. 13, № 7. С. 1011–1019. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://doi.org/10.1002/tee.22658>
21. Bah S. M., Ming F. An improved face recognition algorithm and its application in attendance management system. Array. 2020. Т. 5. С. 100014. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://doi.org/10.1016/j.array.2019.100014>.
22. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks / S. Ren та ін. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2019. Т. 39, № 6. С. 1137–1149. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://doi.org/10.1109/tpami.2016.2577031>
23. Research on Object Tracking Technology Based on Region Proposal Siamese Network / Z. Gu та ін. CIIS 2022: 2022 The 5th International Conference on Computational Intelligence and Intelligent Systems, м. Quzhou China. New York, NY, USA, 2022. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://doi.org/10.1145/3581792.3581793>

24. Bharat Singh, Larry S. Davis. SNIPER: Efficient Multi-Scale Training. NIPS 2018 conference, м. Монреаль, 2018 р. 2018. С. 378. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://arxiv.org/abs/1805.09300v3>.
25. FRMDB: Face Recognition Using Multiple Points of View / P. Contardo та ін. Sensors. 2023. Т. 23, № 4. С. 1939. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://doi.org/10.3390/s23041939>
26. Fine-Tuning – Dive into Deep Learning. Dive into Deep Learning – Dive into Deep Learning 1.0.3 documentation. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: http://d2l.ai/chapter_computer-vision/fine-tuning.html
27. Hybrid dilated faster RCNN for object detection / Н. Пан та ін. Journal of Intelligent & Fuzzy Systems. 2022. С. 1–11. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://doi.org/10.3233/jifs-212740>
28. Face–Iris Multimodal Biometric Identification System / В. Ammour та ін. Electronics. 2020. Т. 9, № 1. С. 85. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://doi.org/10.3390/electronics9010085>
29. Log Gabor filter - Wikipedia. Wikipedia, the free encyclopedia. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: https://en.wikipedia.org/wiki/Log_Gabor_filter.
30. Аналіз сингулярного спектру – Вікіпедія. Вікіпедія. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: https://uk.wikipedia.org/wiki/Аналіз_сингулярного_спектру.
31. Going deeper with two-stream ConvNets for action recognition in video surveillance / Y. Han та ін. Pattern Recognition Letters. 2018. Т. 107. С. 83–90. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://doi.org/10.1016/j.patrec.2017.08.015>
32. Deep Facial Attribute Analysis. Frontiers. [ЕЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://www.frontiersin.org/research-topics/46527/deep-facial-attribute-analysis>.

33. Tucker decomposition - Wikipedia. Wikipedia, the free encyclopedia. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: https://en.wikipedia.org/wiki/Tucker_decomposition.
34. Ghazi M. M., Ekenel H. K. A Comprehensive Analysis of Deep Learning Based Representation for Face Recognition. CVPR on Biometrics. 2020. Т. 2, № 1. С. 418. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://arxiv.org/abs/1606.02894>.
35. Papers with Code - Extended YouTube Faces (E-YTF) Dataset. The latest in Machine Learning | Papers With Code. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://paperswithcode.com/dataset/extended-youtube-faces-e-ytf>
36. Papers with Code - LFW Dataset. The latest in Machine Learning | Papers With Code. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://paperswithcode.com/dataset/lfw>
37. Papers with Code - VGG Face Dataset. The latest in Machine Learning | Papers With Code. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://paperswithcode.com/dataset/vgg-face-1>
38. A Light CNN for Deep Face Representation with Noisy Labels, TIFS 2018. GitHub. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://github.com/AlfredXiangWu/LightCNN>.
39. G. Hu. Attribute-Enhanced Face Recognition with Neural Tensor Fusion Networks IEEE International Conference on Computer Vision (ICCV). 2017 pp. 3764-3773. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ : https://openaccess.thecvf.com/content_ICCV_2017/papers/Hu_Attribute-Enhanced_Face_Recognition_ICCV_2017_paper.pdf
40. Recent Advances in Deep Learning Techniques for Face Recognition / М. Т. Н. Fuad та ін. IEEE Access. 2021. Т. 9. С. 99112–99142. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://doi.org/10.1109/access.2021.3096136>.
41. Residual Networks | Baeldung on Computer Science. Baeldung on Computer Science. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://www.baeldung.com/cs/residual-networks>.

42. Simonyan, K., & Zisserman, A.. Very deep convolutional networks for large-scale image recognition. 2018. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ : <https://ora.ox.ac.uk/objects/uuid:60713f18-a6d1-4d97-8f45-b60ad8aebbce>
43. Makarenko A. Boost Your Image Classification Model with pretrained VGG-16. Medium. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://medium.com/geekculture/boost-your-image-classification-model-with-pretrained-vgg-16-ec185f763104>.
44. Analysis of Gradient Degradation and Feature Map Quality in Deep All-Convolutional Neural Networks Compared to Deep Residual Networks. springerprofessional.de. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://www.springerprofessional.de/en/analysis-of-gradient-degradation-and-feature-map-quality-in-deep/15202130>.
45. Papers with Code - Residual Block Explained. The latest in Machine Learning | Papers With Code. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://paperswithcode.com/method/residual-block>.
46. Expanding Computer Vision Multi-View Stereo Capabilities: Automatic Generation of 3-dimensional Models via 360 Camera Footage Taiwan AILabs. Taiwan AI Labs. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://ailabs.tw/smart-city/expanding-computer-vision-multi-view-stereo-capabilities-automatic-generation-of-3-dimensional-models-via-360-camera-footage/>
47. SCface. SCface - Surveillance Cameras Face Database. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://www.scface.org>.
48. Relu. ResearchGate. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: https://www.researchgate.net/figure/Neural-network-architecture-Note-that-ReLU-activation-function-has-been-used-for-all_fig1_346475847
49. Triplet Loss with Keras and TensorFlow - PyImageSearch. PyImageSearch. [ЭЛЕКТРОННИЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://pyimagesearch.com/2023/03/06/triplet-loss-with-keras-and-tensorflow>

50. ML | Stochastic Gradient Descent (SGD) - GeeksforGeeks. GeeksforGeeks. [ЭЛЕКТРОННЫЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/>.

51. How to Interpret a ROC Curve (With Examples) - Statology. Statology. [ЭЛЕКТРОННЫЙ РЕСУРС]. – РЕЖИМ ДОСТУПУ: <https://www.statology.org/interpret-roc-curve/>.

ДОДАТКИ

Додаток А

Світлини наукових публікацій, виконаних при роботі над кваліфікаційною роботою магістра

(ксерокопії титульної сторінки, сторінки змісту та всіх сторінок із публікацією)

Перелік наукових публікацій:

Метод розпізнавання обличчя під довільним кутом зору / Добржанський В.О., Бармак О.В., Скрипник Т.К. // Науковий журнал «Вісник Хмельницького національного університету. Серія: Технічні науки» Хмельницький, 2023. №6

ISSN 2307-5732
DOI 10.31891/2307-5732

Науковий журнал



ВІСНИК

Хмельницького національного
університету

Технічні науки

Довідка: ВХНУ ТН 20/11/23

Видання: Вісник Хмельницького національного університету. Технічні науки

Категорія фаховості видання: фахове видання України, у якому можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора наук, кандидата наук та ступеня доктора філософії, категорії «Б» філософії, категорії «Б» (наказ МОН №1643 від 28.12.2019, наказ МОН №409 від 17.03.2020).

Напрямок – технічні науки за спеціальностями – 101, 121, 122, 123, 124, 125, 141, 151, 161, 172, 181, 182 (28.12.2019), спеціальності – 131, 132, 133 (17.03.2020)

Назва статті: МЕТОД РОЗПІЗНАВАННЯ ОБЛИЧЧЯ ПІД ДОВІЛЬНИМ КУТОМ ЗОРУ

Автори: Володимир Добржанський, Олександр Бармак, Тетяна Скрипник (Хмельницький національний університет»)

Номер, у який прийнято статтю: №6 до друку рекомендовано буде до 25 грудня 2023 року.

20.11.2023

Начальник відділу
інтелектуальної власності та трансферу технологій Ю.В.Кравчик



УДК 004.8
DOI:

Володимир ДОБРЖАНСЬКИЙ
Хмельницький національний університет
<https://orcid.org/0009-0007-0095-3433>
vdo brzhan@gmail.com
Олександр БАРМАК
Хмельницький національний університет
<https://orcid.org/0000-0003-0739-9678>
alexander.barmak@gmail.com
Тетяна СКРИПНИК
Хмельницький національний університет
<https://orcid.org/0000-0002-8531-5348>
tkskripnik1970@gmail.com

МЕТОД РОЗПІЗНАВАННЯ ОБЛИЧЧЯ ПІД ДОВІЛЬНИМ КУТОМ ЗОРУ

У статті висвітлюється проблема використання технології розпізнавання обличчя в індустріальних застосуваннях. Незважаючи на інтеграцію цієї технології, існують відкриті виклики, такі як верифікація та ідентифікація осіб з різних поз. Особливу проблему становить відсутність належних досліджень у сфері розпізнавання обличчя в відео, зокрема в системах відеоспостереження, де в якості посилання використовуються знімки, отримані зі спеціальних кутів зору (POV). Ці виклики підкреслені в контексті використання фотографій, знятих фронтально та з правого профілю, які традиційно збираються поліцією.

Для вирішення цих проблем і заповнення дослідницької ніші пропонується новий підхід у вигляді бази даних з розпізнавання обличчя за знімками (FRMDB). Ця база включає 28 знімків та 5 відео, отриманих з різних кутів зору для 39 унікальних об'єктів. Основною метою FRMDB є аналіз впливу використання знімків з різних кутів зору на точність розпізнавання обличчя на кадрах відеоспостереження.

Для перевірки ефективності FRMDB та порівняння з існуючими даними проведено тести точності, використовуючи дві глибокі нейронні мережі (CNN), а саме VGG16 і ResNet50. Для цього вони були передзавантажені на попередньо навчені набори даних VGGFace та VGGFace2 для вилучення ознак обличчя. Порівняльний аналіз результатів проведено із застосуванням даних з існуючого дослідження, а саме, Бази даних обличчя з камер відеоспостереження (SCFace).

Результати підкреслюють, що піднабір знімків, який включає фронтальні та праві профільні зображення, показує найнижчий рівень точності серед варіантів, що були протестовані. Це вказує на необхідність додаткового дослідження для визначення оптимальної кількості знімків для ефективного розпізнавання обличчя на кадрах відеоспостереження.

Ключові слова: розпізнавання, відеоспостереження, FRMDB, глибокі нейронні мережі, обличчя, точність.

Volodymyr DOBRZHANSKYI, Oleksandr BARMAK, Tetyana SKRYPNYK
Khmelnitskyi National University

FACIAL RECOGNITION METHOD UNDER ARBITRARY VIEW

The article addresses the challenge of utilizing facial recognition technology in industrial applications. Despite the integration of this technology, there are open issues such as verification and identification of individuals from different poses. The absence of proper research in facial recognition in videos, especially in surveillance systems using snapshots from various Points of View (POV), is a particular concern. These challenges are emphasized in the context of using photographs taken frontally and from the right profile, traditionally collected by the police.

To address these issues and fill the research gap, a new approach is proposed in the form of the Face Recognition from Mugshots Database (FRMDB). This database comprises 28 snapshots and 5 videos taken from different angles for

Набір даних Multi-PIE [4] містить 755,370 кольорових зображень (3072×2048 пікселів) 337 унікальних осіб, записаних в різних сесіях для включення змін пози, освітлення та виразів. Для кожної сесії було знято 13 зображень з кутом від -90° до $+90^\circ$ з кроком 15° на горизонтальній площині за допомогою різних камер на рівні голови. Додатково було знято два зображення під кутом $\pm 30^\circ$ на горизонтальній площині та вище рівня голови. Набір даних доступний за запитом.

База даних ідентифікації фото Національного інституту стандартів та технологій (MID) [5] включає 3228 відтінків сірого зображень (різних розмірів) 1573 осіб. У 1333 суб'єктів є як фронтальні, так і профільні фото, у 131 суб'єкта є два або більше фронтальних зображення, а у 89 суб'єктів є два або більше профільних зображень. База даних доступна за запитом.

Незважаючи на те, що бази даних FERET, Yale, MPI, KFDDB, CAS-PEAL, Multi-PIE та MID включають фото з різних точок зору, вони не містять жодного кадру або відео суб'єктів з відеокamer для аналізу впливу використання різних підмножин зображень з різних кутів на результативність розпізнавання обличчя в сценаріях відеоспостереження.

Метою роботи є: покращення існуючого підходу у розпізнаванні обличчя за зображенням під довільним кутом, навчаючи модель знімками із бази FRMDB, а саме кадрами взятих з камер відеонагляду.

Виклад основного матеріалу

З урахуванням потреби в базах даних обличчя для оцінки можливості розпізнавання обличчя на зображеннях з різних точок зору, ми пропонуємо новий набір даних, який включає 28 різних фото та 5 відео з камер відеоспостереження для кожного об'єкта. Такий набір даних буде корисним для оцінки впливу використання додаткових поз, крім фронтального та правого профілю, які зазвичай доступні в базах даних правоохоронних органів. Для встановлення початкового показнику для запропонованого набору даних ми тестували дві різні глибокі нейронні мережі, а саме VGG16 і ResNet50, натреновані на розпізнавання обличчя на великих базах даних обличчя VGGFace [8] та VGGFace2.

База даних з розпізнавання обличчя на зображенні (FRMDB) включає 39 унікальних осіб, з яких 17 жінок і 22 чоловіки. Середній вік суб'єктів складає 24,6 років, наймолодша особа має 19 років, а найстарша - 52 роки (стандартне відхилення 7,8). Для кожного об'єкта набір даних включає:

- Загальну кількість 28 фото, тобто 28 кольорових зображень, знятих з різних точок зору з суб'єктом, який позує під час зйомки.

- Загальну кількість 5 відео з камер відеоспостереження, знятих з 5 точок зору. Крім того, доступне відео-мозаїка, яка включає всі 5 відео одночасно.

На рисунку 1 представлено 28 фото суб'єкта "031" у базі даних (кожен ідентифікатор - це трицифровий код для збереження анонімності). Кожне фото - це зображення JPEG розміром 972×544 пікселі, зроблені із 7 кутів на горизонтальній площині та 4 кутів на вертикальній площині.

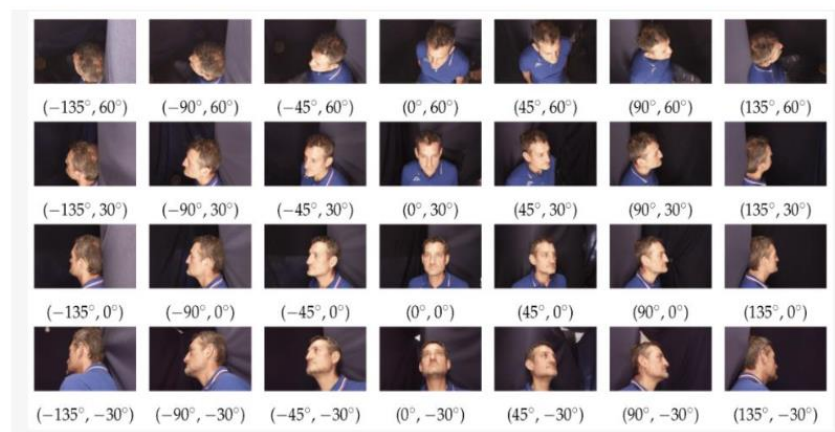


Рисунок 1 – Суб'єкт 031 із бази даних

Зокрема, на горизонтальній площині знімки були зроблені від -135° до $+135^\circ$ з кроком 45° (при цьому 0° - це ззаду об'єкта). На вертикальній площині знімки були зроблені від -30° до $+60^\circ$ (при цьому 0° - це камера на рівні очей суб'єкта) з кроком 30° . У цьому відношенні на рисунку 2 показано різні точки зору на горизонтальній та вертикальній площинах, використані для отримання зображень.

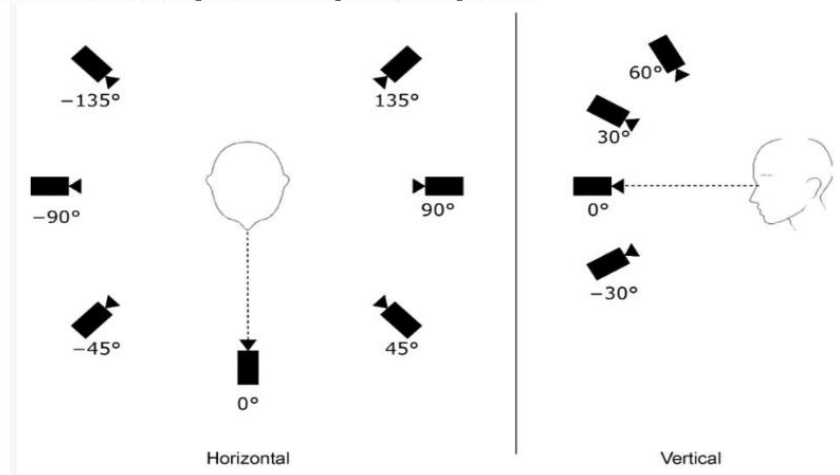


Рисунок 2 – Різні кути огляду на горизонтальній та вертикальній площині

Рисунок 3 містить кадри для кожного з 5 відеозаписів камер відеонагляду, які належать суб'єкту «031» у базі даних. Відео кодується за допомогою кодека H.264 (контейнерний формат Matroska — mkv) і записуються зі швидкістю 60 кадрів на секунду. Розмір кадру — 352×288 пікселів (розмір мозаїки разом із усіма 5 роликками — 1280×720 пікселів). Середня тривалість роликів становить 18,5 с (мінімальна 15 с, максимальна 29 с, стандартне відхилення 2,9 с). Щоб записати відео з камер спостереження, кожного випробуваного попросили підійти до комода, відкрити ящик, дістати документ, підписати документ і повернутися до початкової точки. Під час виконання таких завдань 5 камер, розміщених у 5 різних точках, фіксували об'єкт. Одночасно було знято по 5 відеороликів кожного предмета. Незважаючи на те, що всі відеозаписи були зняті в одній кімнаті з постійним освітленням, фон залежить від точки зору кожної камери, як показано на рисунку 3.

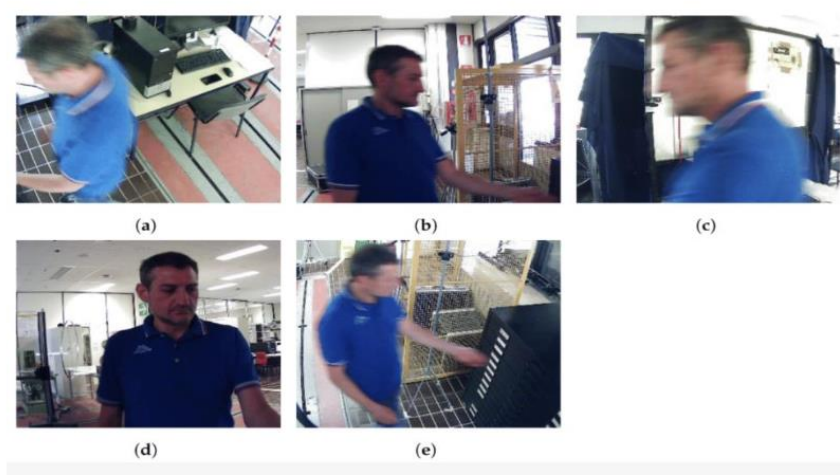


Рисунок 3 – кадри з камер відеонагляду

Для експериментів, представлених у цій статті, ми вручну обрали один кадр для кожного відео та

обрізували обличчя, щоб перевірити ефективність розпізнавання таких кадрів за допомогою різних наборів фотографій. На додаток до описаних фотографій і відео з камер спостереження, FRMDB містить додаткове фронтальне зображення (1920 × 1080 пікселів, JPEG) для кожного об'єкта, зроблене з різним освітленням та камерою, розташованою перед об'єктом.

Використовуючи запропонований набір даних та базу даних SCFace, ми тестували здатність розпізнавання двох різних CNN, а саме VGG16 та ResNet50, з використанням різних датасетів. CNN використовують вектор ознак для кожного обличчя, і ці вектори порівнюються за допомогою відстані чи міри подібності, таких як евклідова відстань чи косинусна подібність. В обох мережах, на вхід ми отримуємо зображення обличчя розміром 224 × 224, а вектор ознак обчислюється через Global Average Pooling на виході останнього згорткового блоку мережі. VGG16 має вектор ознак розміром 512 елементів, ResNet50 - 2048 елементів.

Навчання мереж виконано так, як описано в [9] для VGG16 і в [11] для ResNet50. VGG16 тренувалась з нуля на VGGFace з використанням triplet loss та SGD[10], з партіями розміром 64 та початковою швидкістю навчання 0,01, зменшеною тричі на порядок 10 при зупинці збільшення точності на валідаційному наборі. ResNet50 тренувалась з нуля на VGGFace2 з використанням softmax loss та SGD, з партіями розміром 256 та початковою швидкістю 0,1, зменшеною двічі на порядок 10 при зупинці зменшення помилки. Замість повторного тренування ми використали оригінальні ваги (доступні за посиланнями на VGG16 і ResNet50) за допомогою Keras conversion[11, 12] з оригінальних Caffe моделей.

Ми тестували здатність розпізнавання VGG16 та ResNet50, навчених на VGGFace і VGGFace2, на зображеннях бази даних SCFace і запропонованого набору даних. Ми визначили різні набори фото як посилання для розпізнавання обличчя на знімках з камер відеоспостереження. База даних SCFace містить 9 фото для 130 унікальних суб'єктів, а також 21 додаткове зображення обличчя для кожного унікального суб'єкта. Ми використовуємо її для розуміння, як використання зображень з різних кутів може вплинути на розпізнавання обличчя. Для цього 9 фото для кожного унікального суб'єкта систематично отримуються з різних кутів на горизонтальній площині, від -90° до 90°, з кроком 22,5°.

Таблиця 1 містить різні набори фото. Кожен набір описується кутами (h , v), де h - кут на горизонтальній площині, а v - кут на вертикальній площині. Для SCFace v завжди 0°.

Таблиця 1 – Опис наборів фото

Назва	Фото (SCFace)	Фото (FRMDB)
“Test F”	(0°, 0°)	(0°, 0°)
“Test F-L1-R1”	(0°, 0°), (-22.5°, 0°), (22.5°, 0°)	(0°, 0°), (-45°, 0°), (45°, 0°)
“Test 1”	(0°, 0°), (90°, 0°)	(0°, 0°), (90°, 0°)
“Test 2”	(0°, 0°), (90°, 0°), (-90°, 0°)	(0°, 0°), (90°, 0°), (-90°, 0°)
“Test 3”	(0°, 0°), (90°, 0°), (-90°, 0°), (77.5°, 0°), (-77.5°, 0°)	(0°, 0°), (90°, 0°), (-90°, 0°), (45°, 0°), (45°, 0°)
“Test 4”	(0°, 0°), (90°, 0°), (-90°, 0°), (77.5°, 0°), (-77.5°, 0°), (45°, 0°), (-45°, 0°)	(0°, 0°), (135°, 0°), (-135°, 0°), (90°, 0°), (-90°, 0°), (45°, 0°), (45°, 0°)
“Test 5”	(0°, 0°), (90°, 0°), (-90°, 0°), (77.5°, 0°), (-77.5°, 0°), (45°, 0°), (-45°, 0°), (-22.5°, 0°), (22.5°, 0°)	(0°, 0°), (135°, 0°), (-135°, 0°), (90°, 0°), (-90°, 0°), (45°, 0°), (45°, 0°), (0°, 30°), (135°, 30°), (-135°, 30°), (90°, 30°), (-90°, 30°), (45°, 30°), (45°, 30°)
“Test 6”		(0°, 0°), (135°, 0°), (-135°, 0°), (90°, 0°), (-90°, 0°), (45°, 0°), (45°, 0°), (0°, 30°), (135°, 30°), (-135°, 30°), (90°, 30°), (-90°, 30°), (45°, 30°), (45°, 30°), (0°, 0°)

Назва	Фото (SCFace)	Фото (FRMDB)
		60°), (135°, 60°), (-135°, 60°), (90°, 60°), (-90°, 60°), (45°, 60°), (45°, 60°), (0°, -30°), (135°, -30°), (-135°, -30°)

"Test F": Фронтальне зображення (0°, 0°) для обох баз даних.

"Test F-L1-R1": Фронтальне зображення, лівий кут (для SCFace: -22.5°, 0°; для FRMDB: -45°, 0°) та правий кут (SCFace: 22.5°, 0°; FRMDB: 45°, 0°).

"Test 1": Фронтальне та правий профіль (90°, 0°) для обох баз даних.

"Test 2": Фронтальне, правий профіль та лівий профіль (-90°, 0°).

"Test 3": "Test 2" зображення на один крок ближче до фронтального.

"Test 4": "Test 3" зображення при (45°, 0°) і (-45°, 0°) для SCFace, та (135°, 0°) і (-135°, 0°) для FRMDB.

"Test 5": Всі 9 фото для SCFace та "Test 4" плюс всі фото з кутом 30° на вертикальній площині для FRMDB.

"Test 6": Всі 28 фото для FRMDB.

Щодо зображень обличчя для SCFace використовуємо знімки з кольорових камер на відстані 1 м від суб'єкта, виключаючи трьох суб'єктів з прихованим обличчям. Для FRMDB використовуємо вручну вирізані обличчя з відео. Загалом є 210 зображень для розпізнавання в обох базах даних.

Аналіз ефективності запропонованого методу

Результати на базі SCFace, отримані за допомогою ResNet50 та VGG16, зображені на рисунку 5, який включає точність розпізнавання на зображеннях п'яти відеокамер, знятих на відстані 1 м від кожного унікального об'єкта. VGG16 показав гіршу точність, ніж ResNet50 в кожному рейтингу, незалежно від того, чи розглядали топові знімки або топові ідентифікатори. З ResNet50 правильний об'єкт належить до топ-10 найближчих ідентифікаторів або знімків на 99% зображень від відеокамер безпеки для всіх доступних підмножин відомих знімків (рисунок 5h). Зокрема, при розгляді об'єктів у топ-3 ідентифікаторах розпізнавання обличчя стає ефективним (точність становить понад 98% в усіх тестах, крім "Test 1" та "Test 2") при використанні ResNet50 (рисунок 5d). У топ-3 ідентифікаторах точність досягає мінімуму 97% в підмножині, що складається лише зі спереду і праворуч профільованих знімків, тобто "Test 1".

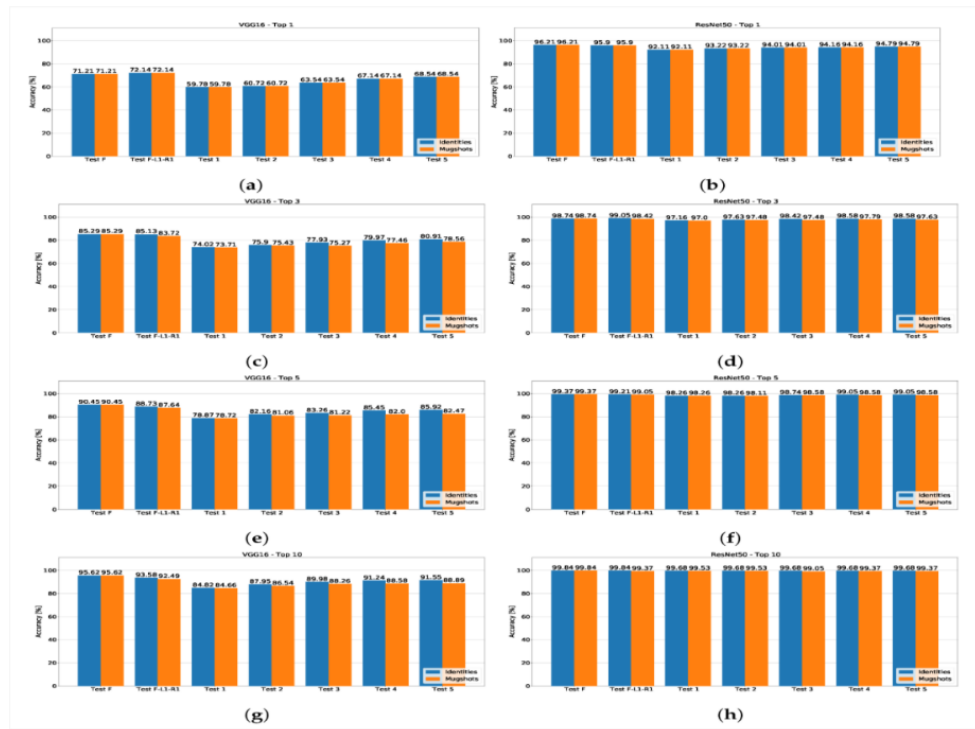


Рисунок 4 – Вимірювання точності SCFace для ResNet50 і VGG16

Отже, отримана точність вказує, що використання більшої кількості зображень з різних точок зору ймовірно не є чинником поліпшення результатів в розпізнаванні обличчя.

У порівнянні з VGG16, ResNet50 з використанням бази даних FRMDB набирає кращі бали в усіх рейтингах "Топ" навіть на запропонованому наборі даних, проте точність значно нижча, ніж на SCFace для обох CNN. Цей результат вказує на те, що FRMDB містить відмінні особливості. Як показано на рисунку 3, кадри відеокамер мають різні перспективи (замість того, щоб містити тільки фронтальні обличчя, як у SCFace), що є головною особливістю. Крім того, відео мають низьку роздільність (352×288 пікселів), емулюючи камери низької якості громадського безпеки, які можуть включати дуже мале зображення (наприклад, 85×85 пікселів). Таким чином, запропонований набір даних здається кращим відображенням ситуацій, які виникають в реальному житті при розпізнаванні обличчя на відеокамерах.

На відміну від бази даних SCFace, підмножина, що складається лише зі спереду профільованого знімка ("Test F"), ніколи не досягає найкращої точності з ResNet50. Замість цього підмножина, що складається зі спереду профільованого знімка, знімка при $(-45^\circ, 0^\circ)$ і знімка при $(45^\circ, 0^\circ)$, тобто "Test F-L1-R1", досягає найкращих результатів у всіх рейтингах "Топ" (рисунок 5b, d, f, h). Наприклад, правильна особа знаходиться в топ-10 (рисунок 5h) найближчих ідентифікаторів для 74,87% кадрів відеокамер безпеки, використовуючи зображення з підмножини "Test F-L1-R1" як посилання. Такий відсоток зменшується до 71,28%, використовуючи лише зображення спереду як посилання. Навіть з VGG16 (рисунок 5a, c, e, g) не спостерігається чіткого переважання підмножини, що складається лише зі спереду профільованого знімка, на відміну від бази даних SCFace. Наприклад, з підмножиною "TestF-L1-R1" правильний об'єкт знаходиться в топ-3 (рисунок 5c) ідентифікаторів та знімків для 44,62% зображень відеокамер безпеки, тоді як зі знімком лише спереду цей відсоток зменшується до 43,08%.

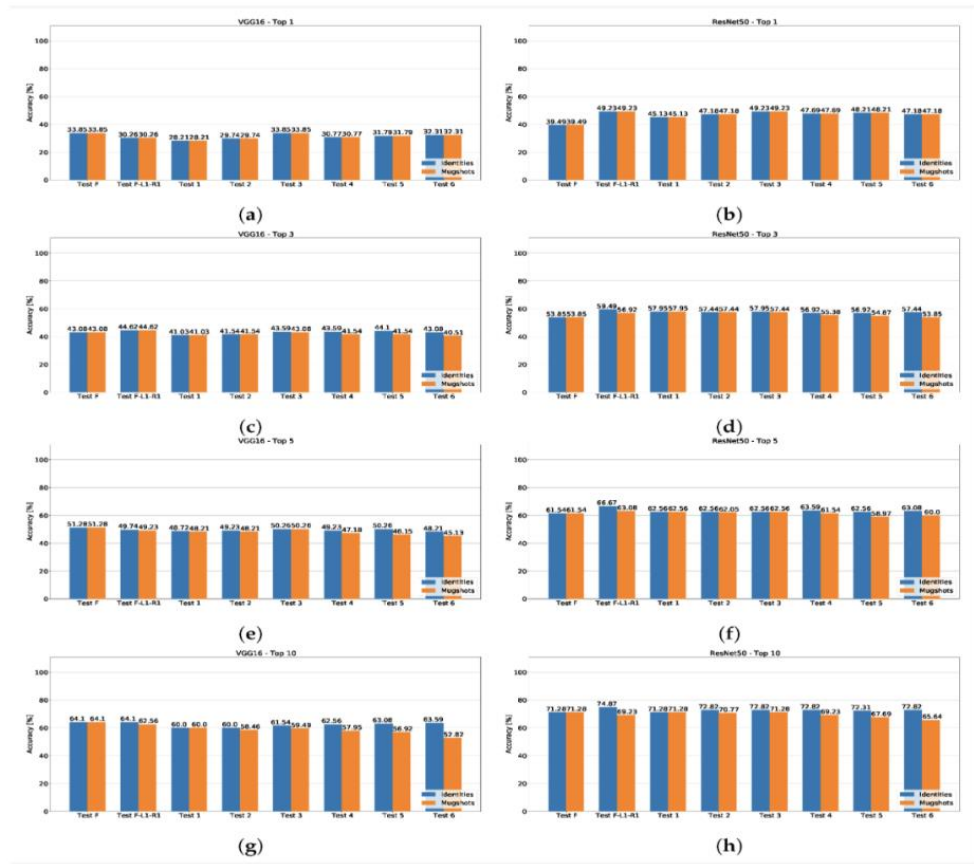


Рисунок 4 – Вимірювання точності FRMDB для ResNet50 і VGG16

ВИСНОВКИ

Загалом результати, отримані використовуючи вибірки фото з більшою кількістю зображень кращі, ніж використання лише фронтального зображення чи фронтального зображення та правого профілю. Такі результати і нижча точність порівняно з базою даних SCFace підтверджують запропонований набір даних, як відповідний для вивчення ефекту використання зображень з різних точок зору для розпізнавання обличчя.

Ми представили FRMDB, тобто набір даних, який включає 28 фотографій мушотів та 5 відео з відеореєстратора безпеки 39 унікальних об'єктів. Запропонований набір даних призначений для оцінки технік розпізнавання обличчя для ідентифікації суб'єктів на відео за допомогою доступних фото.

На запропонованому наборі даних, а також на базі SCFace, ми випробували дві вже встановлені згортовки нейронні мережі, ResNet50 і VGG16, попередньо натреновані на наборах даних VGGFace та VGGFace2 для вилучення обличчя. Такі експерименти дозволяють зробити наступні висновки:

Запропонований набір даних підходить для оцінки технік розпізнавання обличчя для ідентифікації суб'єктів на фото, враховуючи різні точки зору. Нижча точність порівняно з базою даних SCFace підкреслила особливість набору даних. Крім того, вибірка фото, складена лише з фронтального обличчя, не показала тих самих переваг, що й на SCFace, оскільки FRMDB включає відео з різних кутів огляду.

З обома наборами даних традиційні фотографії з відеонагляду, тобто фронтальне зображення та правий профіль, виявилися менш ефективними, ніж інші вибірки фото. Зокрема, на запропонованому FRMDB вибірка, складена з фронтального зображення та зображень при $\pm 45^\circ$ на горизонтальній площині, досягла найкращої точності в більшості тестів.

Для отримання результатів щодо ідеальної кількості фото потрібні додаткові дослідження. Для отримання більш загальних результатів необхідно випробувати більше технік, включаючи ті, які використовуються для розпізнавання обличчя при зміні пози, а також оцінки позиції, з метою вибору фото з позою, яка найбільше відповідає кадрам камери спостереження.

Майбутні роботи з запропонованим набором даних вирішать описані обмеження, додавши більше суб'єктів, відео вищого розширення, щоб мати більше варіацій і, отже, створити базу даних, яка ще більше відображає відеоспостереження в реальному житті.

Література

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* 2015, *521*, 436–444. [Електронний адрес]: <https://www.nature.com/articles/nature14539>
2. Raaijmakers, S. Artificial Intelligence for Law Enforcement: Challenges and Opportunities. *IEEE Secur. Priv.* 2019, *17*, 74–77. [Електронний адрес]: <https://ieeexplore.ieee.org/abstract/document/8821442/>
3. Guo, G.; Zhang, N. A survey on deep learning based face recognition. *Comput. Vis. Image Underst.* 2019, *189*, 102805. [Електронний адрес]: <https://www.sciencedirect.com/science/article/abs/pii/S1077314219301183>
4. Wang, M.; Deng, W. Deep face recognition: A survey. *Neurocomputing* 2021, *429*, 215–244. [Електронний адрес]: <https://www.sciencedirect.com/science/article/abs/pii/S0925231220316945>
5. Vrskova, R.; Hudec, R.; Kamencay, P.; Sykora, P. A New Approach for Abnormal Human Activities Recognition Based on ConvLSTM Architecture. *Sensors* 2022, *22*, 2946. [Електронний адрес]: <https://www.mdpi.com/1424-8220/22/8/2946>
6. Bhatti, M.T.; Khan, M.G.; Aslam, M.; Fiaz, M.J. Weapon Detection in Real-Time CCTV Videos Using Deep Learning. *IEEE Access* 2021, *9*, 34366–34382. [Електронний адрес]: <https://ieeexplore.ieee.org/abstract/document/9353483>
7. Berardini, D.; Galdelli, A.; Mancini, A.; Zingaretti, P. Benchmarking of Dual-Step Neural Networks for Detection of Dangerous Weapons on Edge Devices. In Proceedings of the 2022 18th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), Taipei, Taiwan, 29–31 August 2022; pp. 1–6. [Електронний адрес]: <https://ieeexplore.ieee.org/abstract/document/10004469>
8. Rossi, L.; Paolanti, M.; Pierdicca, R.; Frontoni, E. Human trajectory prediction and generation using LSTM models and GANs. *Pattern Recognit.* 2021, *120*, 108136. [Електронний адрес]: <https://www.sciencedirect.com/science/article/abs/pii/S003132032100323X>
9. Gomez-Barrero, M.; Drozdowski, P.; Rathgeb, C.; Patino, J.; Todisco, M.; Nautsch, A.; Damer, N.; Priesnitz, J.; Evans, N.; Busch, C. Biometrics in the Era of COVID-19: Challenges and Opportunities. *IEEE Trans. Technol. Soc.* 2022, *3*, 307–322. [Електронний адрес]: <https://ieeexplore.ieee.org/abstract/document/9873965/>
10. Guo, G.; Zhang, N. A survey on deep learning based face recognition. *Comput. Vis. Image Underst.* 2019, *189*, 102805. [Електронний адрес]: <https://www.sciencedirect.com/science/article/abs/pii/S1077314219301183>
11. Ammour, B.; Boubchir, L.; Bouden, T.; Ramdani, M. Face-Iris Multimodal Biometric Identification System. *Electronics* 2020, *9*, 85. [Електронний адрес]: <https://www.mdpi.com/2079-9292/9/1/85>
12. Forti, M. AI-driven migration management procedures: Fundamental rights issues and regulatory answers. *Biolaw J.* 2021, *2021*, 433–451. [Електронний адрес]: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3877083

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* 2015, *521*, 436–444. [Електронний адрес]: <https://www.nature.com/articles/nature14539>
2. Raaijmakers, S. Artificial Intelligence for Law Enforcement: Challenges and Opportunities. *IEEE Secur. Priv.* 2019, *17*, 74–77. [Електронний адрес]: <https://ieeexplore.ieee.org/abstract/document/8821442/>
3. Guo, G.; Zhang, N. A survey on deep learning based face recognition. *Comput. Vis. Image Underst.* 2019, *189*, 102805. [Електронний адрес]: <https://www.sciencedirect.com/science/article/abs/pii/S1077314219301183>
4. Wang, M.; Deng, W. Deep face recognition: A survey. *Neurocomputing* 2021, *429*, 215–244. [Електронний адрес]: <https://www.sciencedirect.com/science/article/abs/pii/S0925231220316945>
5. Vrskova, R.; Hudec, R.; Kamencay, P.; Sykora, P. A New Approach for Abnormal Human Activities Recognition Based on ConvLSTM Architecture. *Sensors* 2022, *22*, 2946. [Електронний адрес]: <https://www.mdpi.com/1424-8220/22/8/2946>
6. Bhatti, M.T.; Khan, M.G.; Aslam, M.; Fiaz, M.J. Weapon Detection in Real-Time CCTV Videos Using Deep Learning. *IEEE Access* 2021, *9*, 34366–34382. [Електронний адрес]: <https://ieeexplore.ieee.org/abstract/document/9353483>
7. Berardini, D.; Galdelli, A.; Mancini, A.; Zingaretti, P. Benchmarking of Dual-Step Neural Networks for Detection of Dangerous Weapons on Edge Devices. In Proceedings of the 2022 18th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), Taipei, Taiwan, 29–31 August 2022; pp. 1–6. [Електронний адрес]: <https://ieeexplore.ieee.org/abstract/document/10004469>

8. Rossi, L.; Paolanti, M.; Pierdicca, R.; Frontoni, E. Human trajectory prediction and generation using LSTM models and GANs. *Pattern Recognit.* 2021, *120*, 108136. [Электронный адрес]: <https://www.sciencedirect.com/science/article/abs/pii/S003132032100323X>
9. Gomez-Barrero, M.; Drozdowski, P.; Rathgeb, C.; Patino, J.; Todisco, M.; Nautsch, A.; Damer, N.; Priesnitz, J.; Evans, N.; Busch, C. Biometrics in the Era of COVID-19: Challenges and Opportunities. *IEEE Trans. Technol. Soc.* 2022, *3*, 307–322. [Электронный адрес]: <https://ieeexplore.ieee.org/abstract/document/9873965/>
10. Guo, G.; Zhang, N. A survey on deep learning based face recognition. *Comput. Vis. Image Underst.* 2019, *189*, 102805. [Электронный адрес]: <https://www.sciencedirect.com/science/article/abs/pii/S1077314219301183>
11. Ammour, B.; Boubchir, L.; Bouden, T.; Ramdani, M. Face–Iris Multimodal Biometric Identification System. *Electronics* 2020, *9*, 85. [Электронный адрес]: <https://www.mdpi.com/2079-9292/9/1/85>
12. Forti, M. AI-driven migration management procedures: Fundamental rights issues and regulatory answers. *Biolaw J.* 2021, *2021*, 433–451. [Электронный адрес]: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3877083

Додаток Б

Презентаційний матеріал

Кваліфікаційна робота магістра

МЕТОД РОЗПІЗНАВАННЯ ОБЛИЧЧЯ ЗА ЗОБРАЖЕННЯМ ПІД ДОВІЛЬНИМ КУТОМ ЗОРУ

Виконав : студент 2 курсу ОР «Магістр», група КНм-22-1, **Добржанський В. О.**
Керівник : Зав. кафедри КН, д.т.н., професор, **Бармак О.В.**

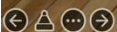


Актуальність теми

Розпізнавання обличчя на зображенні є важливою частиною сучасного життя, особливо у сфері безпеки, як загальної (камери спостереження) так і персональної (особисті файли на пристрої).

Більшість сучасних смартфонів використовують фронтальну камеру для ідентифікації особи, але не завжди цей процес є успішним, через такі фактори як : якість отриманого зображення, недостатня попередня навченість моделі, кут огляду, рівень освітлення.

Деякі підприємства встановлюють автоматичні системи спостереження на своїх об'єктах, які сигналізують, якщо вони зуміли зафіксувати особу, яка не має певного рівня доступу до приміщення, але на жаль через ті самі фактори, дані системи не можуть працювати на 100 % ефективно.



Мета роботи

Метою кваліфікаційної роботи магістра є вдосконалення методу розпізнавання обличчя за зображенням під довільним кутом зору.

Вдосконалений метод повинен ефективно розпізнавати обличчя за зображенням в умовах недостатньої чи надмірної освітленості.



Завдання

Для досягнення поставленої мети визначено наступні задачі:

- ❖ Проаналізувати існуючі способи розпізнавання обличчя на зображенні під довільним кутом зору;
- ❖ Здійснити аналіз згорткових нейронних мереж та їх архітектур для вибору найбільш відповідних для поставленого завдання;
- ❖ Вдосконалити метод розпізнавання обличчя на зображенні під довільним кутом зору та покращити його для умов недостатньої або надмірної освітленості;
- ❖ Підготувати набори даних для донавчання згорткової нейронної мережі;
- ❖ Провести валідацію та аналіз отриманих експериментальних результатів запропонованого методу розпізнавання обличчя на зображенні за стандартними показниками якості.



Об'єкт та предмет дослідження

Об'єкт дослідження – процес розпізнавання обличчя за зображенням під довільним кутом зору.

Предмет дослідження – моделі, методи та засоби для обробки зображень, розпізнавання та класифікації зображень, оцінювання якості ідентифікаторів та класифікаторів.



Наукова новизна

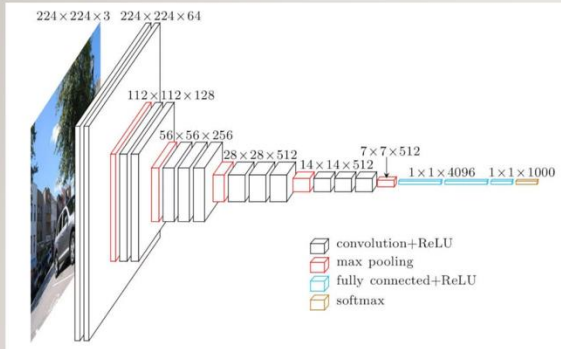
В результаті проведеної роботи було отримано наступний результат:

Вдосконалено метод розпізнавання обличчя на зображенні під довільним кутом зору в умовах недостатнього чи надмірного освітлення, шляхом додаткового навчання початкової моделі, розробленим набором даних, що підвищило статистичні показники якості для усіх розглянутих архітектур моделей глибокого навчання.

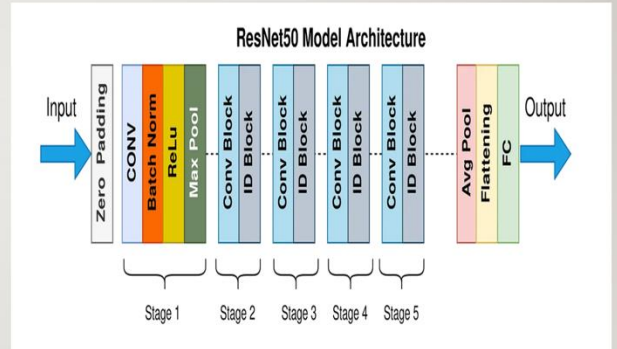
Найкращий результат точності у **99,3%** було отримано з використанням архітектур **«SeNet-50»** та **«ResNet-50»**.



Архітектури згорткових нейронних мереж

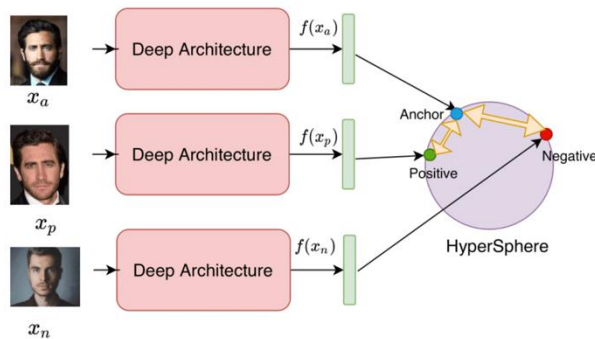


VGG-16

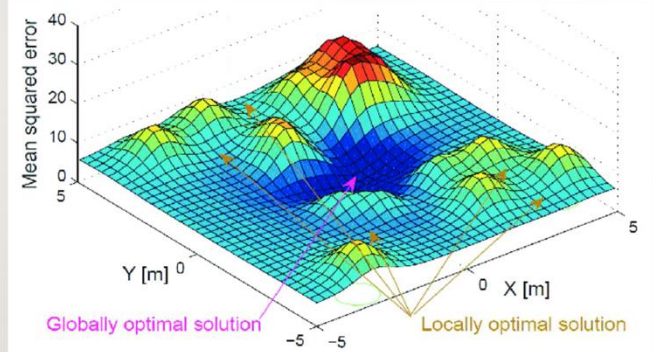


ResNet-50

Навчання згорткової нейронної мережі

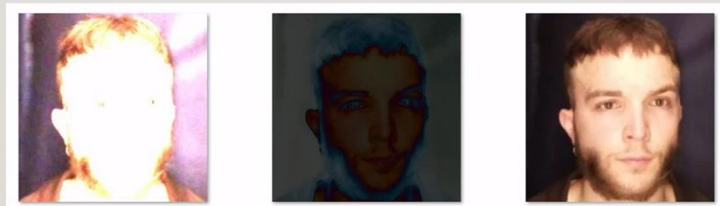
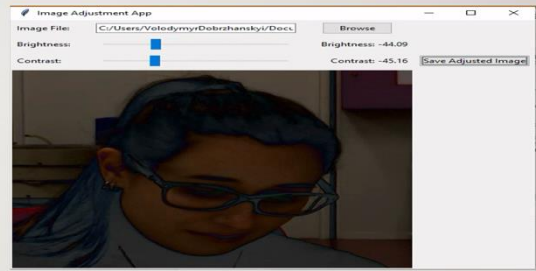
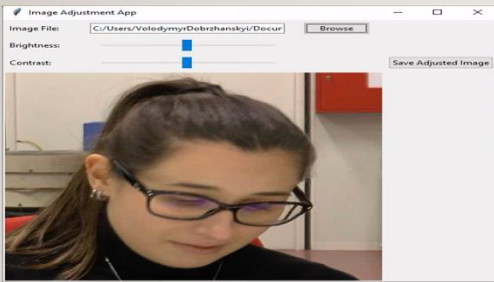


Функція триплетних витрат

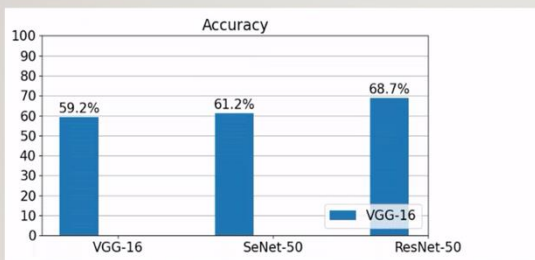
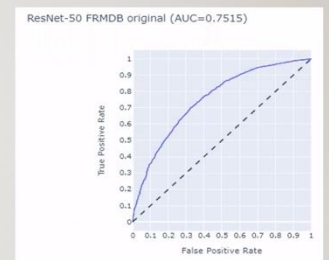
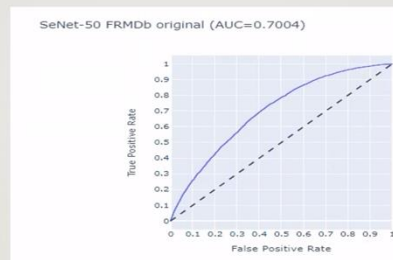
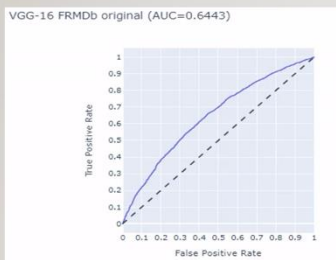


Стохастичний градієнтний спуск

Підготовка вхідних даних



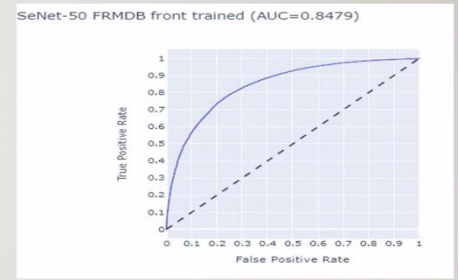
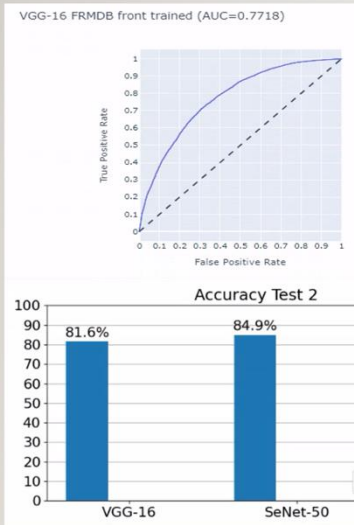
Результати роботи методу



Test 1	Precision	Recall	F1 - score	AUC
VGG-16	0,72	0,65	0,69	0,6443
SeNet-50	0,79	0,67	0,73	0,7004
ResNet-50	0,77	0,67	0,72	0,7515

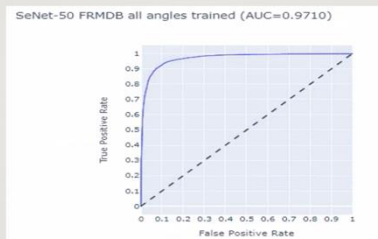
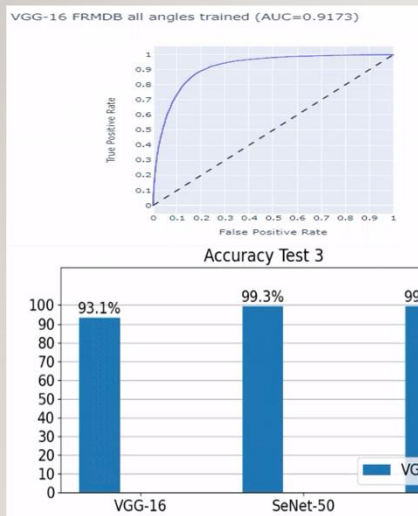


Результати роботи покращеного методу



Test 2	Precision	Recall	F1 - score	AUC
VGG-16	0,75	0,78	0,76	0,7718
SeNet-50	0,8	0,75	0,77	0,8230
ResNet-50	0,82	0,79	0,8	0,8479

Результати роботи покращеного методу



Test 3	Precision	Recall	F1 - score	AUC
VGG-16	0,91	0,92	0,92	0,9173
SeNet-50	0,94	0,96	0,95	0,9710
ResNet-50	0,98	0,97	0,98	0,9890

Дякую за увагу



Додаток В

Програмні коди

```
#Image Adjustment App
import os
import cv2
import tkinter as tk
from tkinter import ttk, filedialog
from PIL import Image, ImageTk

class ImageAdjustmentApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Image Adjustment App")

        self.image_path_var = tk.StringVar()
        self.brightness_var = tk.DoubleVar()
        self.contrast_var = tk.DoubleVar()

        self.create_widgets()

    def create_widgets(self):
        # Image File Input
        file_label = ttk.Label(self.root, text="Image File:")
        file_label.grid(row=0, column=0, padx=5, pady=5, sticky="w")

        file_entry = tk.Entry(self.root, textvariable=self.image_path_var)
        file_entry.grid(row=0, column=1, padx=5, pady=5, sticky="we")

        browse_button = tk.Button(self.root, text="Browse",
            command=self.browse_file)
        browse_button.grid(row=0, column=2, padx=5, pady=5)

        brightness_label = ttk.Label(self.root, text="Brightness:")
        brightness_label.grid(row=1, column=0, padx=5, pady=5,
            sticky="w")

        self.brightness_value_label = ttk.Label(self.root, text="")
        self.brightness_value_label.grid(row=1, column=2, padx=5, pady=5,
            sticky="e")

        brightness_slider = tk.Scale(self.root, from_=-100, to=100,
            variable=self.brightness_var, orient="horizontal", length=200,
            command=self.update_adjustment)
        brightness_slider.grid(row=1, column=1, padx=5, pady=5)

        # Contrast Slider
        contrast_label = ttk.Label(self.root, text="Contrast:")
        contrast_label.grid(row=2, column=0, padx=5, pady=5, sticky="w")

        self.contrast_value_label = ttk.Label(self.root, text="")
        self.contrast_value_label.grid(row=2, column=2, padx=5, pady=5,
            sticky="e")

        contrast_slider = tk.Scale(self.root, from_=-100, to=100,
            variable=self.contrast_var, orient="horizontal", length=200,
            command=self.update_adjustment)
        contrast_slider.grid(row=2, column=1, padx=5, pady=5)

        # Save Button
        save_button = tk.Button(self.root, text="Save Adjusted Image",
            command=self.save_adjusted_image)
        save_button.grid(row=2, column=3, padx=5, pady=5)

        # Image Preview
        self.image_preview_label = ttk.Label(self.root)
        self.image_preview_label.grid(row=3, column=0, columnspan=3)

    def browse_file(self):
        file_selected = filedialog.askopenfilename(filetypes=(("Image files",
            "*.png;*.jpg;*.jpeg")))
        self.image_path_var.set(file_selected)

        # Display the selected image in the preview
        self.display_image_preview()

    def update_adjustment(self, _):
        # Display the updated image in the preview when the slider is moved
        self.display_image_preview()

    def display_image_preview(self):
        image_path = self.image_path_var.get()

        if image_path:
            brightness = self.brightness_var.get()
            contrast = self.contrast_var.get()

            # Read the image
            original_image = cv2.imread(image_path)

            # Adjust brightness and contrast
            adjusted_image = cv2.convertScaleAbs(original_image, alpha=1
                + brightness/100, beta=contrast)

            # Convert the OpenCV image to Tkinter format
            adjusted_image = cv2.cvtColor(adjusted_image,
                cv2.COLOR_BGR2RGB)
            image_pil = Image.fromarray(adjusted_image)
            image_tk = ImageTk.PhotoImage(image=image_pil)

            # Update the label with the new image
            self.image_preview_label.configure(image=image_tk)
            self.image_preview_label.image = image_tk

    def save_adjusted_image(self):
        image_path = self.image_path_var.get()

        if image_path:
            brightness = self.brightness_var.get()
            contrast = self.contrast_var.get()
            brightness_value = self.brightness_var.get()
            contrast_value = self.contrast_var.get()
            self.brightness_value_label.config(text=f"Brightness:
                {brightness_value:.2f}")
            self.contrast_value_label.config(text=f"Contrast:
                {contrast_value:.2f}")

            # Read the image
            original_image = cv2.imread(image_path)

            # Adjust brightness and contrast
            adjusted_image = cv2.convertScaleAbs(original_image, alpha=1
                + brightness/100, beta=contrast)

            # Get the file name and extension
            file_name, file_extension =
                os.path.splitext(os.path.basename(image_path))

            # Save the adjusted image with a new name
            adjusted_image_path = os.path.join(os.path.dirname(image_path),
                f"adjusted_{file_name}{file_extension}")
            cv2.imwrite(adjusted_image_path, adjusted_image)

if __name__ == "__main__":
    root = tk.Tk()
    app = ImageAdjustmentApp(root)
    root.mainloop()

##Image Luminosity Adjustment App
import os
import cv2
import tkinter as tk
from tkinter import filedialog
from tkinter import ttk
from PIL import Image, ImageTk
```

```

class ImageAdjustmentApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Image Luminosity Adjustment App")

        self.image_path_var = tk.StringVar()
        self.luminosity_var = tk.DoubleVar()

        self.create_widgets()

    def create_widgets(self):
        # Image File Input
        file_label = tk.Label(self.root, text="Image File folder:")
        file_label.grid(row=0, column=0, padx=5, pady=5, sticky="w")

        file_entry = ttk.Entry(self.root, textvariable=self.image_path_var)
        file_entry.grid(row=0, column=1, padx=5, pady=5, sticky="we")

        browse_button = ttk.Button(self.root, text="Browse",
        command=self.browse_file)
        browse_button.grid(row=0, column=2, padx=5, pady=5)

        # Luminosity Input
        luminosity_label = tk.Label(self.root, text="Brightness:")
        luminosity_label.grid(row=1, column=0, padx=5, pady=5,
        sticky="w")

        luminosity_entry = ttk.Entry(self.root,
        textvariable=self.luminosity_var)
        luminosity_entry.grid(row=1, column=1, padx=5, pady=5)

        br_label = tk.Label(self.root, text="Contrast:")
        br_label.grid(row=2, column=0, padx=5, pady=5, sticky="w")

        br_entry = ttk.Entry(self.root, textvariable=self.luminosity_var)
        br_entry.grid(row=2, column=1, padx=5, pady=5)

        # Save Button
        save_button = ttk.Button(self.root, text="Save Adjusted Images",
        command=self.save_adjusted_image)
        save_button.grid(row=1, column=2, padx=5, pady=5)

        # Image Preview
        self.image_preview_label = tk.Label(self.root)
        self.image_preview_label.grid(row=2, column=0, columns=3)

        # Light Display
        # light_label = tk.Label(self.root, text="Light:")
        # light_label.grid(row=0, column=3, padx=5, pady=5, sticky="e")

        # light_display_label = tk.Label(self.root,
        textvariable=self.luminosity_var)
        # light_display_label.grid(row=1, column=3, padx=5, pady=5,
        sticky="e")

        def browse_file(self):
            file_selected = filedialog.askopenfilename(filetypes=[("Image files",
            "*.png;*.jpg;*.jpeg")])
            self.image_path_var.set(file_selected)

            # Display the selected image in the preview
            self.display_image_preview()

        def update_adjustment(self, _):
            # Display the updated image in the preview when the entry is
            modified
            self.display_image_preview()

        def display_image_preview(self):
            image_path = self.image_path_var.get()

            if image_path:
                try:
                    luminosity = float(self.luminosity_var.get())
                except ValueError:
                    # Handle non-numeric input

```

```

                    tk.messagebox.showerror("Error", "Please enter a valid numeric
                    value for Luminosity.")
                    return

                # Read the image
                original_image = cv2.imread(image_path)

                # Adjust luminosity
                adjusted_image = cv2.convertScaleAbs(original_image, alpha=1
                + luminosity/100, beta=0)

                # Convert the OpenCV image to Tkinter format
                adjusted_image = cv2.cvtColor(adjusted_image,
                cv2.COLOR_BGR2RGB)
                image_pil = Image.fromarray(adjusted_image)
                image_tk = ImageTk.PhotoImage(image=image_pil)

                # Update the label with the new image
                self.image_preview_label.configure(image=image_tk)
                self.image_preview_label.image = image_tk

                # Display the light value as a percentage
                # light_percentage = int((1 + luminosity/100) * 100)
                # self.luminosity_var.set(f"{light_percentage}")

            def save_adjusted_image(self):
                image_path = self.image_path_var.get()

                if image_path:
                    try:
                        luminosity = float(self.luminosity_var.get())
                    except ValueError:
                        # Handle non-numeric input
                        tk.messagebox.showerror("Error", "Please enter a valid numeric
                        value for Luminosity.")
                        return

                    # Read the image
                    original_image = cv2.imread(image_path)

                    # Adjust luminosity
                    adjusted_image = cv2.convertScaleAbs(original_image, alpha=1
                    + luminosity/100, beta=0)

                    # Get the file name and extension
                    file_name, file_extension =
                    os.path.splitext(os.path.basename(image_path))

                    # Save the adjusted image with a new name
                    adjusted_image_path = os.path.join(os.path.dirname(image_path),
                    f"adjusted_{file_name}{file_extension}")
                    cv2.imwrite(adjusted_image_path, adjusted_image)

            if __name__ == "__main__":
                root = tk.Tk()
                app = ImageAdjustmentApp(root)
                root.mainloop()

##required libraries
#pip install opencv-python
##pip install git+https://github.com/rcmalli/keras-vggface.git
#pip install tensorflowpip install keras_applications --no-deps
#pip install scikit-learn
#pip install keras_applications==1.0.4 --no-deps
#pip install keras_preprocessing==1.0.2 --no-deps
#pip install keras
#pip install git+https://github.com/rcmalli/keras-vggface.git
#pip install keras-preprocessing
#pip install --upgrade keras==2.12.0
#pip install --upgrade tensorflow==2.11
#pip install --upgrade keras-applications==1.0.8
#pip install matplotlib
import cv2
import os
import numpy as np

```

```

def load_dataset(mainDir, resizeDim=(224, 224), savePath=None):
    X = list()
    y = list()
    z = list()

    for subject in os.listdir(mainDir):
        path = os.path.join(mainDir, subject)

        # Skip non-directory items (e.g., '.DS_Store')
        if not os.path.isdir(path):
            continue

    for filename in os.listdir(path):
        filepath = os.path.join(path, filename)

        face = cv2.imread(filepath)
        face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
        face = cv2.resize(face, resizeDim)

        label = subject

        filename = filename.split('.')[0]
        pose = filename.split('_')[1]

        # print('Extracting...', filepath, label)

        z.append(pose)
        y.append(label)
        X.append(face)

    X = np.asarray(X)
    y = np.asarray(y)
    z = np.asarray(z)

    if savePath:
        np.savez_compressed(savePath, arr_0=X, arr_1=y, arr_2=z)
        return X, y, z
load_dataset('C:/Users/VD/Documents/facerec/data/content/datarepo/frm
db/cropped/mugshots', (224, 224),
'C:/Users/VD/Documents/facerec/data/content/photosignalling.npz')
load_dataset('C:/Users/VD/Documents/facerec/data/content/datarepo/frm
db/cropped/surveillance', (224, 224),
'C:/Users/VD/Documents/facerec/data/content/security.npz')
filename =
"C:/Users/VD/AppData/Local/Programs/Python/Python37/Lib/site-
packages/keras_vggface/models.py"
text = open(filename).read()
open(filename, "w+").write(text.replace('keras.engine.topology',
'tensorflow.keras.utils'))
import tensorflow as tf
from tensorflow.keras import __version__
import sklearn

print('Using Keras version:', __version__)

device_name = tf.test.gpu_device_name()
if device_name == "":
    device_name = "None"
print('Using TensorFlow version:', tf.__version__, 'GPU:', device_name)

print('The scikit-learn version is {}'.format(sklearn.__version__))
import time
import tensorflow as tf

from keras_vggface.vggface import VGGFace
from keras_vggface.utils import preprocess_input

def get_embedding(model, sample, preprocessing_v):
    """ Extracts a face embedding from a face image
    """
    sample = sample.astype('float32')
    samples = np.expand_dims(sample, axis=0)
    samples = preprocess_input(samples, version=preprocessing_v)
    yhat = model.predict(samples)
    return yhat[0]

```

```

def compute_embeddings(modelName, dataPath, savePath=None,
verbose=False):
    """ Computes the face embeddings for a dataset.
    If True, prints the model summary and execution time.
    """
    data = np.load(dataPath)
    X, y, z = data['arr_0'], data['arr_1'], data['arr_2']
    model = VGGFace(model=modelName, include_top=False,
input_shape=(224, 224, 3), pooling='avg')
    if verbose:
        model.summary()
        embeddings = list()
        t0 = time.time()

        preprocess_version = 2
        if modelName == 'vgg16':
            preprocess_version = 1

    for sample in X:
        embedding = get_embedding(model, sample, preprocess_version)
        if embeddings is np.ndarray:
            print('numpy array')
            embeddings.append(embedding)
        embeddings = np.asarray(embeddings)

    t1 = time.time()
    if verbose:
        print("Time:", t1-t0, 's, for', X.shape[0], 'images.')
    if savePath:
        np.savez_compressed(savePath, arr_0=embeddings, arr_1=y, arr_2=z)
        #compute
        compute_embeddings('vgg16',
'C:/Users/VD/Documents/facerec/data/content/photosignalling.npz',
'C:/Users/VD/Documents/facerec/data/content/embeddings-vgg16-
photosignalling.npz', True)
        compute_embeddings('resnet50',
'C:/Users/VD/Documents/facerec/data/content/photosignalling.npz',
'C:/Users/VD/Documents/facerec/data/content/embeddings-resnet-
photosignalling.npz', True)
        compute_embeddings('senet50',
'C:/Users/VD/Documents/facerec/data/content/photosignalling.npz',
'C:/Users/VolodymyrDobrzanskiy/Documents/facerec/data/content/emb
eddings-senet50-photosignalling.npz', True)

    ##
    from PIL import Image
    import numpy as np
    from sklearn.preprocessing import Normalizer
    from sklearn.preprocessing import LabelEncoder
    from scipy.spatial import distance as dist
    import copy
    import matplotlib.pyplot as plt

def load_dataset_dictionaries(photosignallingPath, securityPath,
normalize=True, verbose=False):
    """ Puts the embeddings in the datasets at photosignallingPath (the
mugshots)
    """
    photosignallingData = np.load(photosignallingPath)
    securityData = np.load(securityPath)
    photoEmb, photoLabel, photoCam = photosignallingData['arr_0'],
photosignallingData['arr_1'], photosignallingData['arr_2']
    secEmb, secLabel, secCam = securityData['arr_0'], securityData['arr_1'],
securityData['arr_2']
    in_encoder = Normalizer(norm='l2')
    if normalize:
        photoEmb = in_encoder.transform(photoEmb)
        secEmb = in_encoder.transform(secEmb)

    if verbose:
        print("Photosignalling shapes: ", photoEmb.shape, photoLabel.shape,
photoCam.shape)
        print("Security shapes: ", secEmb.shape, secLabel.shape, secCam.shape)

    photoLabEmbCam = zip(photoLabel, photoEmb, photoCam)

```

```

photoLabEmbCam = sorted(photoLabEmbCam, key=lambda item:
(item[0], item[2]))
photoLabel = [label for label, emb, cam in photoLabEmbCam]
photoEmb = [emb for label, emb, cam in photoLabEmbCam]
photoCam = [cam for label, emb, cam in photoLabEmbCam]

secLabEmbCam = zip(secLabel, secEmb, secCam)
secLabEmbCam = sorted(secLabEmbCam, key=lambda item: (item[0],
item[2]))
secLabel = [label for label, emb, cam in secLabEmbCam]
secEmb = [emb for label, emb, cam in secLabEmbCam]
secCam = [cam for label, emb, cam in secLabEmbCam]

photoDict = { }
for label, emb, cam in photoLabEmbCam:
    if label not in photoDict:
        photoDict[label] = { }
    photoDict[label][cam] = emb

secDict = { }
for label, emb, cam in secLabEmbCam:
    if label not in secDict:
        secDict[label] = { }
    secDict[label][cam] = emb

return photoDict, secDict

def distanceRank(embeddingCam, mugshotsDict):
    """ Computes the Euclidean distance between the embedding
    "embeddingCam" and
    all the embeddings in the dictionary mugshotsDict.

    Parameters
    -----
    embeddingCam : ndarray of float32
        Embedding of a face in a security cam

    mugshotsDict : dictionary
        Contains the mugshot embeddings {identity: {pose:
        embedding }},
        where an embedding is a ndarray of float32

    Returns
    -----
    distance_list : list
        Sorted list of tuple (distance, identity), where the distance
        is the Euclidean distance between the embeddingCam and an
        embedding of such identity.

    """
    distance_list = list()
    for identity in mugshotsDict:
        for pose in mugshotsDict[identity]:
            distance = dist.euclidean(embeddingCam,
mugshotsDict[identity][pose])
            dist_label = (distance, identity)
            distance_list.append(dist_label)
    return sorted(distance_list)

def filterMugshotDictByPoseList(mugshotsDict, poseList):
    """ Filters the mugshot dictionary according to the list of poses to be
    considered.

    """
    labemb = { }
    labemb = copy.deepcopy(mugshotsDict)
    identity_deleted = list()
    for identity in mugshotsDict:
        allPoseDetected = True
        for pose in poseList:
            if pose not in mugshotsDict[identity]:
                allPoseDetected = False
        if allPoseDetected:
            for pose in mugshotsDict[identity]:
                if pose not in poseList:
                    labemb[identity].pop(pose)
        else:
            del labemb[identity]

```

```

        identity_deleted.append(identity)
        print('identity ', identity, ' removed because some of the pose are not
available')
        return labemb, identity_deleted

def filterSecurityCamsByCamList(securityDict, camList,
identity_deleted):
    """ Filters the security cams dictionary according to the list of cams to
    be
    considered and the deleted identities (in case some identities were
    removed
    from the mugshots dictionary)

    """
    filteredMug, identity_deleted =
filterMugshotDictByPoseList(labemb_mug_dict, poseList)
    filteredCam = filterSecurityCamsByCamList(labemb_cam_dict,
camList, identity_deleted)
    return filteredMug, filteredCam

def test(mugshotDict, securityDict, poseList, camList):
    """ Computes the ranking of the nearest identities and mugshots
    comparing the
    security cams embeddings with those of the mugshots.

    """
    filteredMug, filteredCam = filterDicts(mugshotDict, securityDict,
poseList, camList)

    result = list()
    labelResult = list()
    for identity in filteredCam:
        for cam in filteredCam[identity]:
            sample = filteredCam[identity][cam]
            result.append(distanceRank(sample, filteredMug))
            labelResult.append(identity)

    listRankImg = list() # nearest mugshots ranking
    listRankId = list() # nearest identities ranking
    for i in range(len(result)):
        i_rank = result[i]
        i_label = labelResult[i]
        betterId = set()
        for j in range(len(i_rank)):
            id = i_rank[j][1]
            if id == i_label:
                listRankId.append(len(betterId))
                listRankImg.append(j)
                break;
            else:
                betterId.add(id)
    return listRankId, listRankImg, result, labelResult

def TopN(listRank, N):
    """ Computes the percentage of images for which the correct result is in
    the
    first N nearest results (identities or mugshots)

    """
    count = 0
    for item in listRank:
        if item < N:
            count = count+1
    countPercentage = count/len(listRank)
    return countPercentage

def plot_histograms(id_bars, img_bars, decimals, labels, title, filename):
    """ Plots the experiment results.

    """
    if decimals > 0:
        id_bars = [round(val, decimals) for val in id_bars]
        img_bars = [round(val, decimals) for val in img_bars]
    fig_size = [22,10]
    title_size = 24
    data_size = 26

```

```

x_pos = np.arange(len(labels))
width = 0.35

fig, ax = plt.subplots(figsize=fig_size)
plt.ylim(0, 111)
ax.bar(x_pos - width/2, id_bars, width, label='Recognitions', zorder=3)
#ax.bar(x_pos + width/2, img_bars, width, label='Mugshots', zorder=3)

#ax.set_ylabel('%', fontsize=title_size + 2, rotation='horizontal')
#ax.set_title('Accuracy on Binary Dataset', fontsize=title_size)
ax.set_xticks(x_pos)
ax.set_xticklabels(labels, fontsize=data_size - 2)
ax.set_yticks([0, 10, 20,30, 40,50, 60,70, 80,90, 100])
ax.set_yticklabels([0, 10, 20,30, 40,50, 60,70, 80,90, 1000],
fontsize=data_size-2)
ax.legend(loc='lower right', fontsize=data_size - 2)
ax.grid(b=True, axis='y', zorder=0)
plt.ylabel('Accuracy [%]', fontsize=data_size - 2)
plt.title(title, fontsize=data_size)

for i in range(len(labels)):
    plt.text(i-0.2, id_bars[i] + 1, str(id_bars[i]+'%'), ha = 'center',
fontsize=data_size - 2, zorder=3)
    #plt.text(i+0.2, img_bars[i] + 1, str(img_bars[i]), ha = 'center',
fontsize=data_size - 2, zorder=3)

fig.tight_layout()

plt.savefig(filename, transparent=True)

plt.show()

def run_test(mugshotsPath, securityPath, normalize, tops, poseList,
testLabel, verbose):

```

```

""" Runs a test on the FRMDB dataset.

"""
mughostDict, camDict = load_dataset_dictionaries(mugshotsPath,
securityPath, normalize)
perc_top_id = {}
perc_top_imgs = {}
for N in tops:
    camList = ['01', '02', '03', '04', '05']
    if verbose:
        print(testLabel, '- Top', N, 'Orientations:', poseList)
    listRankId, listRankImg, result, labelResult = test(mughostDict,
camDict, poseList, camList)
    if verbose:
        print('rank ID:', listRankId)
        print('rank img:', listRankImg)
        print('rank ID avg:', np.average(listRankId))
        print('rank img avg:', np.average(listRankImg))
        #print('result', result)
        #print('labelResult', labelResult)

    perc_top_id[N] = TopN(listRankId, N)*100
    perc_top_imgs[N] = TopN(listRankImg, N)*100

    if verbose:
        print('ids in Top', N, perc_top_id[N])
        print('imgs in Top', N, perc_top_imgs[N])
        print('\n')

return perc_top_id, perc_top_imgs

```



ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

гр. КНМ-22-1 *Добржанського Володимира Олеговича* за темою: Метод розпізнавання обличчя за зображенням під довільним кутом зору

1. Актуальність теми

В сучасному світі, системи розпізнавання обличчя на основі штучних нейронних мереж досить широко поширені, та присутні навіть у телефонах. Штучні нейронні мережі мають змогу дуже швидко обробляти зображення та розпізнавати обличчя, проте вони показують 100% точність лише в ідеальних умовах. Тема роботи є актуальною, оскільки вона описує підхід, згідно яким зменшує вплив таких факторів як кут зору та рівень освітлення на ефективність розпізнавання обличчя за зображенням.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Тема кваліфікаційної роботи студента повністю відповідає предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до кваліфікаційної роботи магістра. Робота вирішує науково-прикладну задачу у сфері комп'ютерних наук (а саме застосування інтелектуальних методів обробки інформації (моделей глибокого навчання) до розпізнавання зображень обличчя людей під різними кутами зору, та при різній освітленості. У роботі проведено дослідження яке розвиває існуючі знання та процедури.

3. Професійні та особистісні якості магістранта

Під час виконання кваліфікаційної роботи магістра, студент *Добржанський В.О.*, проявив себе дисциплінованим студентом, вчасно виконуючи поставлені перед ним завдання та показав високий рівень знань та професійних навичок за спеціальністю «Комп'ютерні науки».

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Робота виконана самостійна, академічного плагіату не виявлено. До всіх запозичень у наведено відповідні посилання на джерела.

5. Наукова новизна та оригінальність запропонованих підходів

В ході дослідження було отримано такі результати: Вдосконалено метод розпізнавання обличчя на зображенні під довільним кутом зору в умовах недостатнього чи надмірного освітлення, шляхом додаткового навчання початкової моделі, розробленим

набором даних, що підвищило статистичні показники якості для усіх розглянутих архітектур моделей глибокого навчання. Найкращий результат точності у 99,3% було отримано з використанням архітектур «SeNet-50» та «ResNet-50».

Отримані результати оприлюднені у науковому журналі «Вісник Хмельницького національного університету. Серія: Технічні науки», Хмельницький, 2023, №6. Назва статті: Метод розпізнавання обличчя під довільним кутом зору / Добржанський В.О., Бармак О.В., Скрипник Т.К.

6. Ступінь оволодіння методами дослідження

У процесі роботи над покращенням методу розпізнавання обличчя за зображенням під довільним кутом зору, студент Добржанський В.О. показав високий рівень володіння методами, що були використані у дослідженні.

7. Повнота та якість розкриття теми роботи

Тема роботи повністю розкрита, актуальність предметної області та відомі дослідження щодо обраної тематики достатньо проаналізовані. Достовірність наукових положень підтверджуються отриманими результатами.

8. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу

Матеріал кваліфікаційної роботи Добржанського В.О. подано логічно, послідовно та аргументовано. Мова і стиль викладення роботи відповідають стандартам, що забезпечує доступність сприйняття матеріалу та відповідає сучасним вимогам до кваліфікаційних робіт.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Може мати практичне застосування в системах безпеки, які базуються на ідентифікації людини за зображенням обличчя, а також у системах відеоспостереження, що забезпечить ефективне розпізнавання обличчя навіть в умовах недостатньої видимості, що зумовлено недостатнім чи надмірним освітленням.

10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота

З огляду на рівень виконання та забезпечення усіх необхідних вимог, вважаю, що кваліфікаційна робота Добржанського Володимира Олеговича може бути допущена до захисту та заслуговує на оцінку «Відмінно».

Науковий керівник  д.т.н., проф. Олександр Бармак



ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

гр. КНМ-22-1 Добржанського Володимира Олеговича за темою: Метод розпізнавання обличчя за зображенням під довільним кутом зору

1. Актуальність обраної теми

Розпізнавання обличчя за зображенням під довільним кутом зору є актуальною та перспективною темою, адже головною сферою застосування є сфера безпеки, в тому числі і персональних даних. Хоча системи розпізнавання існують досить давно, але всі вони схильні до похибки, а максимальна ефективність їх роботи досягається тоді, коли обличчя на зображенні розташоване фронтально. Перевагою удосконаленого методу є те, що він дозволяє зменшити похибку при розпізнаванні обличчя за зображенням, в умовах коли обличчя знаходиться під довільним кутом зору та в умовах недостатнього чи надмірного освітлення.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Тема кваліфікаційної роботи студента повністю відповідає предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до кваліфікаційної роботи магістра. Робота вирішує науково-прикладну задачу з сфери комп'ютерного зору, яка є складовою предметної області комп'ютерних наук (а саме застосування інтелектуальних методів обробки інформації (моделей глибокого навчання) до розпізнавання зображень обличчя людей під різними кутами зору, та при різній освітленості. У роботі присутнє дослідження яке розвиває існуючі підходи та отримані нові результати які за певними показниками – вищі, чим у аналогічних підходах.

3. Повнота розкриття мети та завдань дослідження

В результаті виконання роботи поставлена мета та завдання повністю розкриті автором.

4. Наявність наукової новизни

В кваліфікаційній роботі представлена новизна, відповідна спеціальності 122 «Комп'ютерні науки» в межах обраної області дослідження. Продемонстровані результати мають наукове та інноваційне значення, та оприлюднені у науковому журналі «Вісник Хмельницького національного університету».

5. Зміст кожного розділу роботи

Кваліфікаційна робота складається із чотирьох розділів. У першому розділі проведено дослідження сучасних методів та підходів до розпізнавання обличчя за зображенням під

довільним кутом зору. Другий метод присвячений розробці покращеного методу розпізнавання обличчя та навчання нейромережі. У третьому розділі наведено та описано прототип програмної реалізації методу розпізнавання обличчя під довільним кутом зору. У четвертому розділі виконано дослідження з використанням розробленого програмного засобу та наведено результати ефективності роботи покращеного методу розпізнавання обличчя за зображенням під довільним кутом зору.

6. Ступінь розкриття теми роботи

Тема кваліфікаційної роботи повною мірою розкрита та обґрунтована. Проведено детальний аналіз актуальності, предметної області та новітніх досліджень у межах теми розпізнавання обличчя за зображенням під довільним кутом зору. Завдання дослідження були виконані та проведено аналіз результатів прикладного застосування покращеного методу.

7. Якість оформлення кваліфікаційної роботи

Оформлення роботи відповідає нормам та вимогам, що ставляться до оформлення кваліфікаційних робіт.

8. Недоліки кваліфікаційної роботи

Доцільно було б використати набір даних із більшою кількістю зображень для тестування запропонованого методу, та об'єднати в один додатки для підготовки вхідних наборів даних.

9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота.

Робота виконана на високому рівні та відповідає усім необхідним вимогам.

Кваліфікаційна робота може бути допущена до захисту та заслуговує на оцінку «Відмінно».

Опонент

Яшине О. М. доктор фізико-математичних наук

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА ДО ЗАХИСТУ ЗА РЕЗУЛЬТАТАМИ АНАЛІЗУ ЗВІТУ ПОДІБНОСТІ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод розпізнавання обличчя за зображенням під довільним кутом зору

Автор: В.О.Добржанський

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: д.т.н., професор, зав. кафедри КН О.В.Бармак

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) за програмою Anti-Plagiarism виявлені 6% запозичень вказують на документ автора роботи (ЗВІТ з науково-дослідної практики).
- 2) За програмою UNICHECK виявлені 9.5% є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 6% і 9.5% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КН





О. В. Бармак

Р. В. Багрій

О. В. Бармак

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 6.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 13%**

ID: 122559 Назва: КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА на тему Метод розпізнавання обличчя за зображенням під довільним кутом зору Додано в БД: 2023-12-11 Автора: В.О.Добржанський Керівники: О.В. Бармак Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	81020	1186	6004 (7%)	95 (8%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Ім'я користувача:
Кафедра КН

ID перевірки:
1015996161

Дата перевірки:
12.12.2023 09:49:36 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
12.12.2023 09:54:50 EET

ID користувача:
100005671

Назва документа: КНм-22-1 Добржанський_02

Кількість сторінок: 82 Кількість слів: 13364 Кількість символів: 98425 Розмір файлу: 3.74 MB ID файлу: 1015679072

9.53% Схожість

Найбільша схожість: 1.68% з джерелом з Бібліотеки (ID файлу: 1015647281)

8.81% Джерела з Інтернету 932 Сторінка 84

2.27% Джерела з Бібліотеки 67 Сторінка 91

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

80.2% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

Немає вилучених Інтернет-джерел

80.2% Вилученого тексту з Бібліотеки 1 Сторінка 91

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 2