

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

**ДОСЛІДЖЕННЯ ТА ПОРІВНЯННЯ МЕТОДІВ НОРМАЛІЗАЦІЇ**  
**КОЛЬОРУ ОБЛИЧЧЯ ВІДНОСНО ЕТАЛОННОГО ЗОБРАЖЕННЯ**  
(тема)

Виконав:  
здобувач 2 року навчання,  
групи ІНФМ-24-1

Ларін І. П.  
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика  
(повна назва освітньої програми)

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

Допускається до захисту

Завідувач кафедри інформатики \_\_\_\_\_  
(підпис)

Кобилін О. А.  
(прізвище, ініціали)

2025 р.

## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Ларіну Івану Павловичу  
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження та порівняння методів нормалізації кольору обличчя відносно еталонного зображення

затверджена наказом університету від 14 листопада 2025 року № 1045Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 24 листопада 2025 р.

3. Вихідні дані до роботи методи нормалізації кольору обличчя відносно еталонного зображення, літературні джерела щодо застосування методів нормалізації кольору, програмні засоби для реалізації вибраних методів нормалізації та десктоп-застосунку, зображення людей для тестування методів.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Аналіз сучасних методів нормалізації кольору обличчя відносно еталонного зображення.2. Аналіз літературних джерел щодо апробації результатів застосування існуючих методів нормалізації обличчя на зображеннях3. Формування покрокового алгоритму для кожного із вибраних методів нормалізації.4. Візуалізація сформованих покрокових алгоритмів.5. Розробка програмного застосунку, що надасть змогу нормалізувати колір обличчя відносно еталону кожним із вибраних методів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми нормалізації кольору обличчя відносно еталонного зображення, об'єкт та мета дослідження, постановка задачі, блок-схеми алгоритмів вибраних методів нормалізації, приклади еталонних зображень для класів сформованого набору даних, ілюстрація результатів нормалізації кожним із вибраних методів, висновки, перспективи та апробація роботи.

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

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

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	29.09.2025	
2	Аналіз завдання, підбір літератури	30.09.25-07.10.25	
3	Аналіз літератури з досліджуваної проблеми	08.10.25-14.10.25	
4	Дослідження особливостей методів нормалізації	15.10.25-20.10.25	
5	Розробка алгоритмів нормалізації	21.10.25-27.10.25	
6	Програмна реалізація	28.10.25-05.11.25	
7	Обґрунтування отриманих результатів	06.11.25-11.11.25	
8	Оформлення пояснювальної записки	12.11.25-14.11.25	
9	Перевірка на нормоконтроль	23.11.25-10.12.25	
10	Перевірка на плагіат	24.11.25-10.12.25	
11	Рецензування	25.11.25-10.12.25	
12	Підготовка презентації та доповіді	26.11.25-22.12.25	
13	Занесення роботи в електронний архів	27.11.25-22.12.25	
14	Попередній захист кваліфікаційної роботи	01.12.25-22.12.25	

Дата видачі завдання 29 вересня 2025 р.

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

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

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 107 с., 5 табл., 62 рис., 1 дод., 39 джерел.

АЛГОРИТМ, БАЛАНС БІЛОГО, БІБЛІОТЕКА OPENCV, БЛОК-СХЕМА, КОРЕКЦІЯ ОСВІТЛЕННЯ, НОРМАЛІЗАЦІЯ ОБЛИЧЧЯ, РОЗПІЗНАВАННЯ ОБЛИЧЧЯ, ЯСКРАВІСТЬ.

Об'єктом дослідження є методи нормалізації кольору обличчя на зображенні.

Метою дослідження є порівняння методів нормалізації кольору обличчя відносно еталонного зображення.

Для дослідження взято такі методи: Histogram matching, Color transfer, Color transfer на основі Face Parsing. Суть методів детально продемонстровано шляхом побудови блок-схем. Проаналізовано сучасне застосування методів нормалізації у застосунках та відповідних літературних джерелах.

Наукова новизна полягає у порівнянні методів, направлених на вирішення проблеми нормалізації кольору обличчя відносно еталонного зображення, на основі критеріальних описів.

Проведене дослідження ґрунтується на предметній області та архітектурі, що вивчені та представлені, як результат попередньої роботи. Використаний застосунок є одним із мікросервісів спроектованої системи.

Результати даного дослідження можуть бути використані у практичних задачах, пов'язаних з комп'ютерним зором та, зокрема, обробленням кольору обличчя людини на зображеннях або відеокадрах.

У результаті дослідження розроблено застосунок для нормалізації кольору обличчя відносно еталонного зображення трьома методами, а також здійснено розрахунок точності проведених операцій на основі відповідного критеріального опису.

## ABSTRACT

Explanatory note to the qualification work: 107 pages, 5 tables, 62 figures, 1 appendix, 39 sources.

ALGORITHM, BRIGHTNESS, FACE NORMALIZATION, FACE RECOGNITION, FLOWCHART, ILLUMINATION CORRECTION, OPENCV LIBRARY, WHITE BALANCE.

The object of the research is methods of normalizing facial color in images.

The purpose of the research is to compare methods of normalizing facial color relative to a reference image.

The following methods were used for the research: Histogram matching, Color transfer, and Color transfer based on Face Parsing. The essence of the methods is demonstrated in detail by constructing block diagrams. The current application of normalization methods in applications and relevant literature sources is analyzed.

The scientific novelty lies in the comparison of methods aimed at solving the problem of face color normalization relative to a reference image, based on criterion descriptions.

The research is based on the subject area and architecture studied and presented as a result of previous work. The application used is one of the microservices of the designed system.

The results of this research can be used in practical tasks related to computer vision and, in particular, the processing of human face color in images or video frames.

As a result of the research, an application was developed to normalize the color of faces relative to a reference image using three methods, and the accuracy of the operations performed was calculated based on the corresponding criterion description.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	8
Вступ.....	9
1 Аналіз існуючих вебзастосунків розпізнавання та ідентифікації користувачів інформаційних систем в Україні та за кордоном .....	10
1.1 Аналіз сучасних методів нормалізації облич на зображеннях та приклади їх практичного застосування .....	10
1.1.1 Аналіз методу Histogram matching або Histogram specification .....	11
1.1.2 Аналіз методу Color transfer (Reinhard та інші) .....	12
1.1.3 Аналіз методу Color transfer на основі Face Parsing .....	13
1.1.4 Аргументація вибору методів .....	15
1.1.5 Приклади практичного застосування розглянутих методів .....	16
1.2 Аналіз літературних джерел щодо апробації результатів застосування існуючих методів нормалізації облич на зображеннях.....	18
1.3 Постановка задачі дослідження.....	23
2 Особливості вибраних методів нормалізації обличчя на зображенні .....	25
2.1 Метод Histogram matching.....	25
2.1.1 Основна ідея методу Histogram matching .....	25
2.1.2 Алгоритм методу Histogram matching.....	26
2.2 Метод Color transfer .....	29
2.2.1 Основна ідея методу Color transfer.....	29
2.2.2 Алгоритм методу Color transfer .....	30
2.3 Метод Color transfer на основі Face Parsing.....	36
2.3.1 Основна ідея методу Color transfer на основі Face Parsing.....	36
2.3.2 Алгоритм методу Color transfer на основі Face Parsing.....	37

2.4	Формування методики для нормалізації обличчя на зображенні ....	40
2.5	Моделювання структури програмного застосунку для нормалізації обличчя на зображенні .....	42
3	Дослідження методів нормалізації обличчя на зображенні щодо вибраної предметної області.....	47
3.1	Вибір інструментальних засобів для реалізації вибраних методів.....	47
3.2	Етапи програмної реалізації вибраних методів нормалізації обличчя на зображенні .....	48
3.2.1	Формування набору даних для тестування .....	48
3.2.2	Програмна реалізація мікросервісної архітектури застосунку.....	50
3.2.3	Реалізація допоміжних функцій .....	53
3.2.4	Структура реалізації методу Histogram matching .....	54
3.2.5	Структура реалізації методу Color transfer .....	57
3.2.6	Структура реалізації методу Color transfer на основі Face Parsing .....	59
3.3	Застосування методів нормалізації обличчя на зображенні до вибраної предметної області.....	66
3.4	Порівняльний аналіз досліджених методів нормалізації обличчя на зображенні .....	89
3.4.1	Аналіз метричних оцінок точності нормалізації методами .....	89
3.4.2	Аналіз результатів порівняння нормалізованих зображень до еталону .....	92
3.5	Аналіз спотворень на нормалізованих зображеннях.....	97
3.6	Перспективи подальших досліджень.....	98
	Висновки .....	100
	Перелік джерел посилання .....	102
	Додаток А Алгоритм методу Color transfer на основі Face Parsing.....	107

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Альbedo – відбиткова здатність поверхні без тіней і текстур

AdaIN – Adaptive Instance Normalization (метод перенесення стилю через адаптивну нормалізацію)

AR face dataset (AR) – набір даних з фотографіями облич з варіаціями освітлення, емоцій, аксесуарів

Face ID – цифровий ідентифікатор обличчя

GAN – Generative Adversarial Network (генеративно-змагальна нейронна мережа)

Histogram specification – метод нормалізації освітлення обличчя

*HSV* – колірний простір (*H* – колірний тон, *S* – насиченість, *V* – колір)

*III2I3* – колірний простір (*I1* – яскравість, *I2/I3* – колір)

*LSLM* – колірний простір, оснований на фізіології зору (*L* – світлість, *S* – сигнал синього, *LM* – змішані сигнали)

OpenCV – Open Computer Vision Library (бібліотека для роботи з обличчям)

*RGB* – базовий колірний простір (*R* – червоний, *G* – зелений, *B* – синій)

*XGB* – варіант перестановки каналів у *RGB*

*XYZ* – абсолютний колірний простір, близький до сприйняття людини

*YCbCr* – колірний простір для цифрового відео (*Y* – яскравість, *Cb/Cr* – колір)

*YIQ* – колірний простір (*Y* – яскравість, *I/Q* – колір)

*YUV* – колірний простір (*Y* – яскравість, *U/V* – колір)

*ZRG* – варіант перестановки каналів у *RGB*

## ВСТУП

Коректне розпізнавання обличчя завжди було проблемою. Суттю даної проблеми було не тільки вірно ототожнити основні риси людини, з пам'яті або із документів, з тими, що надано для перевірки. Найпоширенішими проблемами, наприклад, є погане освітлення, через яке людина не може чітко розгледіти інше обличчя, фізичні зміни обличчя, положення, з якого одна людина спостерігає за іншою. Через це люди завжди намагались створити умови для чіткої ідентифікації людини: створювали місця з гарним освітленням або ставили вартів, у яких була гарна пам'ять на обличчя.

Із стрімким розвитком застосунків та програм, які оточують нас на кожному кроці, стало очевидним, що з'являться інструментальні засоби, що будуть ідентифікувати обличчя користувача і надавати йому доступ до певних можливостей.

Згадані вище проблеми нікуди не ділися, а тільки зросли у масштабі, враховуючи вимоги сьогодення. Наразі на правильне розпізнавання обличчя критично може вплинути, наприклад, кут нахилу обличчя, наявність окулярів, освітлення у кімнаті, відтінок шкіри. Деякі застосунки Face ID від Apple можуть розпізнавати користувача навіть за поганих умов, але методи, якими вони керуються, є комерційною таємницею.

Саме тому, на даний момент, дослідження методів, які б допомогли привести зображення до певного визначеного еталону, за яким повинне здійснюватися порівняння, є актуальним, а тема кваліфікаційної роботи «Дослідження та порівняння методів нормалізації кольору обличчя відносно еталонного зображення» є затребуваною.

Програма, яка реалізує, один із таких методів, могла б стати гарним доповненням для застосунки, що використовує розпізнавання обличчя для більш ефективної ідентифікації користувача.

# 1 АНАЛІЗ ІСНУЮЧИХ ВЕБЗАСТОСУНКІВ РОЗПІЗНАВАННЯ ТА ІДЕНТИФІКАЦІЇ КОРИСТУВАЧІВ ІНФОРМАЦІЙНИХ СИСТЕМ В УКРАЇНІ ТА ЗА КОРДОНОМ

1.1 Аналіз сучасних методів нормалізації облич на зображеннях та приклади їх практичного застосування

Проблематика розпізнавання обличчя сягає початку часів існування людства. Відколи були створені перші незалежні поселення, постала проблема розпізнати, хто наближається – друг чи ворог. Завжди існували фактори, які критично можуть впливати на розпізнання обличчя, наприклад, освітлення та його якість, наявність додаткових елементів одягу, відстань від спостерігача до ідентифікованого об'єкта.

З появою електронних застосунків, вплив на результат розпізнавання став ще критичніший. Людина може дуже швидко провести повторний аналіз після зміни факторів, які їй заважають. Наприклад, якщо при порівнянні документів з обличчям особи перевіряючий не може розібрати дані, то він ввімкне ліхтар. Повторний запуск автоматизованого процесу ідентифікації у застосунку може бути дорогавартісним або мати період очікування між спробами, що може бути критичним у деяких ситуаціях. Через цю причину при порівнянні еталонного та поточного зображення, зокрема, де зображене обличчя, дуже важливо дотримуватись певних умов щодо зображень, наприклад, робити фото при гарному освітленні, без зайвих елементів одягу та макіяжу, на відстані, що зможе охопити усе обличчя. Іноді такі вимоги неможливо виконати через непереборні обставини або існуючі умови можуть різнитися – освітлення обличчя буде різним у залежності від знаходження у кімнаті. У такому випадку, щоб уникнути зайвих помилок при ідентифікації обличчя, краще задіяти попереднє оброблення вхідних даних, а саме застосувати надійні технології програмної нормалізації облич для подальшого використання поліпшених даних.

У межах дослідження будуть розглянуті методи, що нормалізують рівень освітлення на зображеннях. Існує багато відомих способів вирішення даної проблеми, але для ідентифікації та подальшого надання доступу, доцільно використовувати «reference-based» методи, що вирівнюють рівень освітлення до еталонного рівня. За таких умов, еталонне зображення має бути зроблене за найкращих можливих умов.

Серед методів, які гарно зарекомендували себе під час вирішення такого класу задач, виділимо такі:

- Histogram matching або Histogram specification;
- Color Transfer (Reinhard та інші);
- Color space normalization на основі Face Parsing.

#### 1.1.1 Аналіз методу Histogram matching або Histogram specification

Метод Histogram matching, або специфікація гістограм, є одним з методів групи «reference-based». Основна його ідея полягає у побудові гістограм для вхідного і еталонного зображень з подальшим перетворенням та приведенням їхніх значень для подібності.

Одна із вагомих переваг методу Histogram matching полягає у індивідуальній «нормі» для кожного зображення. Більшість методів, що працюють з гістограмами, вирівнюють їх відносно певної «норми», яка може не завжди відповідати певній ситуації. Наприклад, може гарно працювати з групою облич «А», але погано з групою облич «Б» або за різного освітлення, чи різних природних умов.

Histogram matching визначає для кожного зображення свій еталон, до якого прагне. Це накладає обмеження на метод (як і на всі розглянуті далі), що, за умови поганого еталону, результат буде поганим. Якщо система буде мати гарно підібраний еталон, то результат роботи методу буде подібний «нормі», мати той самий візуальний стиль та характер освітлення.

Застосування методу Histogram matching є поширеним у галузях, де особливо важлива уніфікація наборів даних. Наприклад, у біометрії, де для ідентифікації обличчя важливою є кольорова схема та тип освітлення на фото, у гістології та інших медичних напрямленнях, де важливою є нормалізація кольору еталонів для кращої ідентифікації захворювань.

До недоліків даного методу можна віднести такий: за умови значних відмінностей у освітленні або кольоровій гамі, метод може породжувати неприродні відтінки шкіри чи інші артефакти. Зазначена проблема проявляється саме на портретних фото. Оскільки метод є глобальним та працює з обличчям як із однією областю, а не з багатьма, то він може втрачати локальні деталі і обличчя буде здаватись, у такому випадку, пластиковим.

У випадку нормалізації кольору усього обличчя відносно еталону, метод показує свою ефективність, оскільки для подальшої ідентифікації обличчя за допомогою біометрії, фактор гарного освітлення найчастіше є більш важливим, ніж збереження локальних деталей.

### 1.1.2 Аналіз методу Color transfer (Reinhard та інші)

Метод Color Transfer, запропонований Reinhard та співавторами, також є одним із потужних способів для нормалізації кольору відносно еталонного зображення. Головна ідея даного методу полягає у перенесенні *Lab*-простору одного зображення на той самий простір іншого зображення. *Lab*-простір – це колірний простір, де *L* показує, наскільки світлий колір, а *a* і *b* описують його відтінок у двох протилежних осях (зелений – червоний та синій – жовтий).

Використання простору є ефективним через розподілення яскравості та кольору, що дозволяє коректно виконувати операції над ними.

Для нормалізації кольору, метод обчислює середнє та стандартні відхилення кожного каналу, а потім намагається привести їх до подібності шляхом регуляції компонент простору.

Таким чином, результат роботи зазначеного методу є менш схильним до артефактів і, у той самий час, більш природним для сприйняття людиною.

Серед недоліків методу Color Transfer так само можна визначити його глобальність, через що результат роботи методу може бути неідеальним для певних областей. Крім того, через роботу із кожним компонентом окремо, при наявності сильного кольорового зсуву на еталоні, поточне зображення набуде його відтінку.

Метод застосовується у відео- та фотообробленні. Наприклад, у ситуації коли серію фотографій необхідно привести до одного відтінку шкіри або кольорової палітри. Крім того, метод набув популярності у кінематографії для вирівнювання кольору на декількох різних сценах, що зняті із різних ракурсів або в різний час чи різних умовах. Як і Histogram matching, даний метод застосовується у медицині для вирівнювання кольору шкіри та інших частин тіла на знімках для більш точної ідентифікації захворювань.

### 1.1.3 Аналіз методу Color transfer на основі Face Parsing

Метод Color transfer, що працює на основі Face Parsing є логічним продовженням Color transfer. Головна його перевага полягає у роботі із зображенням по фрагментах, а не в цілому. Спочатку вхідне та еталонні зображення розбиваються на певну кількість ділянок, для кожного із цих фрагментів застосовуються операції над простором *Lab*, що і для звичайного Color transfer.

Перевага такого методу полягає у тотальному контролі зображення. Наприклад, користувач може налаштувати роботу методу лише над обличчям, виключивши фон чи волосся. Дана можливість дозволяє ліквідувати один із недоліків оригінального методу, який міг некоректно вирівняти колір фону, вирівнюючи колір шкіри, і в результаті отримати елементи фону в некоректних відтінках (при коректних відтінках шкіри).

Працюючи із фрагментами, метод може більш гнучко налаштовувати перенесення налаштувань у випадках, коли оригінальний метод мав би їх узагальнювати із-за необхідності вирівняти колір рівномірно на усьому зображенні.

Серед недоліків методу Color transfer на основі Face Parsing можна зазначити не тільки необхідність коректного еталону, але й велику його залежність від способу розбиття на частини, способу виокремлення елементів на фото, а також кількості сегментів. Крім того, недоліком може бути відсутність усереднення між сегментами при нормалізації, що може привести до артефактів, наприклад, видимості меж кордонами. Ще одним недоліком є велика обчислювальна складність, оскільки тепер операція проводиться стільки разів, на скільки сегментів розбито зображення, що при великій кількості фрагментів може значно уповільнити роботу.

На відміну від оригінального методу, Color transfer на основі Face Parsing є менш універсальним і може застосовуватись для специфічних завдань.

Застосування методу Color transfer на основі Face Parsing багато в чому подібне до вище згаданих методів, зокрема, він використовується у медицині та комп'ютерній графіці. Серед унікальних сфер використання даного методу можна зазначити глибокі генеративні моделі, такі як GAN чи дифузійні мережі. Ці моделі є дуже залежними від подібності даних між зображеннями, а нормалізація кольору по частинах дозволяє більш якісно наблизити зображення один до одного. Серед сфер застосування методу Color transfer на основі Face Parsing можна виокремити, наприклад, deepfake-технології та face reenactment, а також для створення високоякісних візуальних образів у віртуальній та доповненій реальності.

### 1.1.4 Аргументація вибору методів

У межах даного дослідження для детального аналізу обрано три методи, що нормалізують колір обличчя відносно еталону:

- Histogram matching;
- Color Transfer;
- Color transfer на основі Face Parsing.

Вибір саме цих методів обґрунтовується їхньою ефективністю у обробці кольорових характеристик. Кожен із методів має свої особливості стосовно нормалізації, через це вони складають гарну основу для аналізу та дослідження.

Метод Color transfer на основі Face Parsing представляє менш розвинутий напрямок «reference-based» методів, орієнтованих на обробку із поділом на сегменти. Використання сегментації обличчя дозволяє здійснювати нормалізацію у найбільш необхідних зонах без врахування сторонніх об'єктів та об'єктів фону. Через таку особливість метод є менш залежним від варіацій освітлення та має вищу точність розпізнавання. Таким чином, метод дозволяє показати переваги інтеграції моделей ідентифікації та сегментації разом із класичними підходами до корекції кольору.

Color Transfer обрано як приклад одного з найбільш цитованих і поширених методів у цій сфері. Він дозволяє досягнути мінімальної різниці між візуальною природністю і подібністю значень кольорів зображень. Робота із кольорами та яскравістю дозволяє досягти гарного результату без масштабних спотворень.

Метод Histogram matching було включено як базовий метод, що забезпечує повну подібність розподілів кольору між зображеннями. Попри ризик втрати природності у випадках значних відмінностей між еталоном та аналізованим зображенням, цей підхід є важливим для дослідження, оскільки дозволяє оцінити переваги і недоліки жорсткого підлаштування під еталон.

Крім того, застосування у біометрії та медицині робить зазначений метод особливо необхідним для детального аналізу в межах даного дослідження.

Таким чином, зазначені методи обрані через те, що вони презентують різні підходи до нормалізації та часто застосовуються у сучасних застосунках. Порівняння результатів, отриманих під час тестування даних методів, а також врахування загальних переваг та недоліків, дасть змогу визначити метод, що найбільше підходить для вирішення поставленої задачі дослідження.

### 1.1.5 Приклади практичного застосування розглянутих методів

Вибрані для дослідження методи мають широке застосування у різних застосунках та бібліотеках, зокрема:

– метод *Histogram matching* традиційно застосовується у завданнях, де критично важлива статистична однорідність даних. У Adobe Photoshop функція *Match Color* дозволяє користувачам швидко вирівнювати палітру фотографій, наприклад, при створенні колажу. У геоінформаційних системах (Esri ArcGIS) метод використовується щодо даних із супутників, а також аерознімків для подальшого спрощенням аналізу. У медицині він інтегрований у бібліотеку *HistomicsTK* та застосовується вебплатформою *Digital Slide Archive* для нормалізації кольорових знімків у сфері гістології. Приклади застосування методу *Histogram matching* до чорно-білого та кольорового зображень наведено на рисунках 1.1 [1] та 1.2 [2];

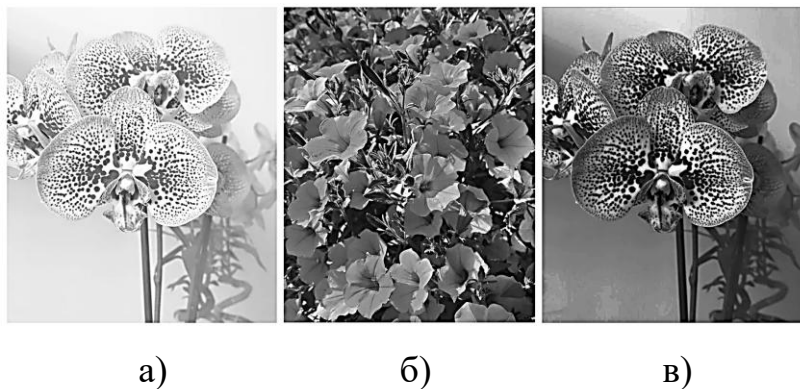


Рисунок 1.1 – Приклад застосування методу *Histogram matching* до чорно-білого зображення: а) аналізоване зображення; б) еталонне зображення; в) нормалізоване зображення



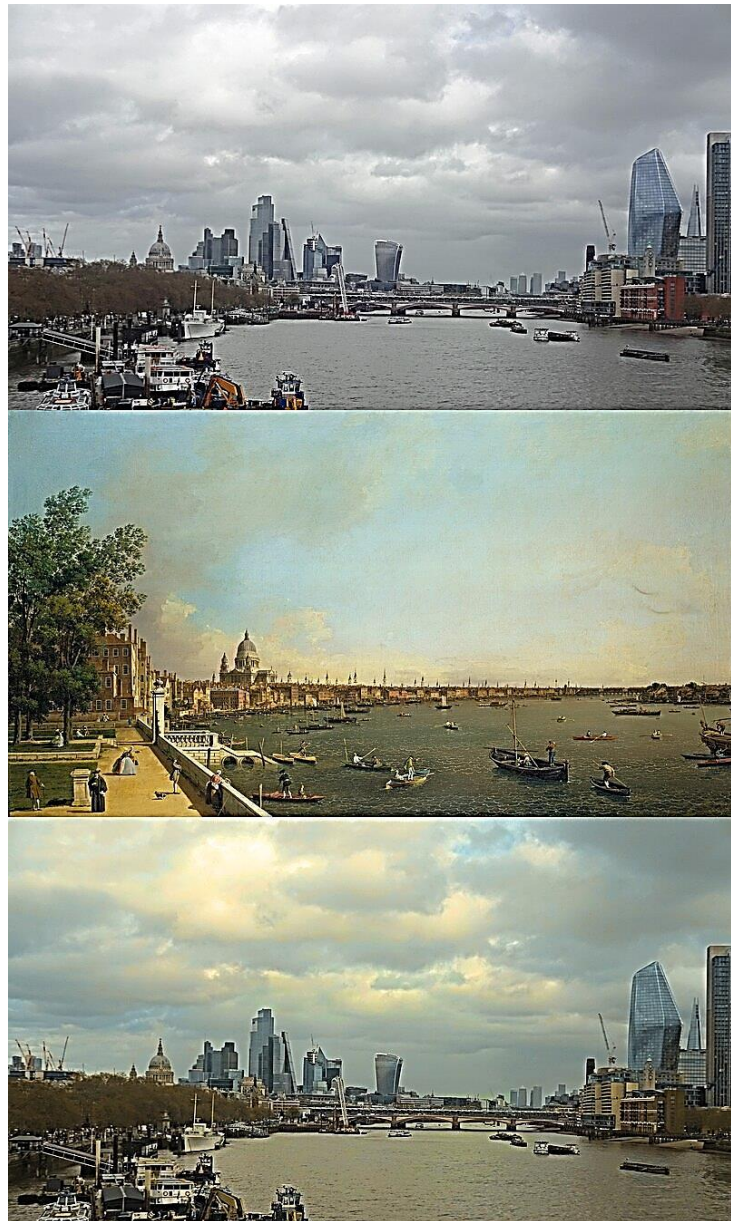


Рисунок 1.3 – Приклад застосування методу Color transfer до кольорового зображення

## 1.2 Аналіз літературних джерел щодо апробації результатів застосування існуючих методів нормалізації облич на зображеннях

Під час дослідження проведено аналіз літературних джерел. У цьому підрозділі будуть розглянуті методи і підходи до нормалізації кольору обличчя, які використано різними вченими у їх наукових дослідженнях.

У статті [4] пропонується глибока архітектура для автоматичного розфарбовування відеолінійних зображень у ту саму кольорову гаму, що і задані еталонні зображення. Структура архітектури складається із мережі перетворення кольорів та мережі часового уточнення на основі 3U-мереж. Мережа зміни кольору приймає цільові лінійні зображення, а також лінійні та кольорові зображення еталонних зображень, як вхідні дані, та генерує відповідні цільові кольорові зображення. Щоб обробити велику різницю між цільовим зображенням лінійного зображення та еталонними кольоровими зображеннями, автори пропонують шар уваги, який використовує нелокальний підбір подібності для визначення регіональної відповідності між цільовим зображенням та еталонними зображеннями, а потім перетворює поточну кольорову інформацію відповідно до еталонної. Щоб забезпечити глобальну узгодженість кольорової гами, додатково включена адаптивна нормалізація екземплярів (AdaIN) з параметрами перетворення, отриманими з багатосарової AdaIN, яка описує глобальний кольоровий стиль посилянь, визначених мережею. Експерименти показують, що метод має найкращу продуктивність при розфарбовуванні відео порівняно з сучасними методами.

У статті [5] розглянуто розфарбовування лінійного рисунку на основі еталонів. Колір, текстура та затінення відображаються на основі абстрактного ескізу, який, як зазначають автори, значною мірою залежить від точності моделювання залежностей між ескізом та еталоном. Також зазначено, що деякі існуючі методи посилюють складність навчання механізму уваги, наприклад, протокол самостійного навчання та втрати на основі GAN. Щоб зрозуміти наскільки таке навчання є нестабільним, автори виявляють градієнтний потік уваги та спостерігають за градієнтним конфліктом між її гілками. На основі цих експериментів, автори пропонують вирішення проблеми градієнта, шляхом зберігання домінуючої градієнтної гілку та видаляючи конфліктні. У дослідженні запропоновано новий механізм уваги, що використовує дану стратегію навчання, Stop-Gradient Attention (SGA), який значно перевершує базовий рівень уваги з кращою стабільністю навчання. Порівняно з сучасними

модулями для розфарбовування лінійного рисунку, підхід демонструє значні покращення початкової відстані Фреше (Fréchet Inception Distance, FID, до 27,21%) та індексу структурної подібності (Structure Similarity, SSIM, до 25,67%) за кількома контрольними показниками.

У дослідженні [6] розглянуто розфарбовування мультфільмів на основі еталонів. Автори вважають цей підхід більш інтуїтивно зрозумілим, ніж розфарбовування за іншими підказками, такими як кольорові точки або текстові підказки. Зазначається складність такого завдання оскільки розфарбовування за орієнтиром є складним завданням, тому що стиль розфарбованого зображення повинен відповідати стилю еталонного зображення як з точки зору глобальної кольорової композиції, так і локального кольорового відтінку. У статті пропонується нова платформа, заснована на навчанні, яка розфарбовує ескіз на основі особливості колірному стилю, витягнутої з опорного кольорового зображення. Платформа містить екстрактор колірному стилю для вилучення ознак з кольорового зображення, мережу розфарбовування для створення багатомасштабних вихідних зображень шляхом поєднання ескізу та отриманих ознак, а також багатомасштабний дискримінатор для покращення реальності вихідного зображення. Якісні і кількісні оцінки показують, що запропонований метод перевершує існуючі методи, забезпечуючи як чудову візуальну якість, так і узгодженість стилю з орієнтирами в завданні розфарбовування на основі орієнтирів.

У роботі [7] представлено концепцію нормалізації колірному простору (Color Space Normalization, CSN) та два методи CSN, тобто метод нормалізації всередині колірних компонентів (CSN-I) та метод нормалізації між кольоровими компонентами (CSN-II), для підвищення розпізнавальної здатності колірних просторів для розпізнавання облич. Різні колірні простори, зазвичай, демонструють різну розпізнавальну здатність, і експерименти авторів в межах Face Recognition Grand Challenge (FRGC) показують, що колірні простори *RGB* та *XYZ* слабші, ніж колірні простори *III2I3*, *YUV*, *YIQ* та *LSLM* для розпізнавання облич. У дослідженні застосовано методи CSN для

нормалізації слабких колірних просторів, таких як колірні простори  $RGB$  та  $XYZ$ , три гібридні колірні простори  $XGB$ ,  $YRB$  та  $ZRG$ , а також 10 випадково згенерованих колірних просторів. Експерименти з використанням найскладнішого FRGC версії 2, з 12776 навчальними зображеннями, 16028 контрольованими цільовими зображеннями та 8014 неконтрольованими зображеннями, показують, що запропоновані методи CSN можуть значно та послідовно покращити розпізнавальну здатність слабких колірних просторів. Зокрема, нормалізовані колірні простори  $RGB$ ,  $XYZ$ ,  $XGB$  та  $ZRG$  є ефективнішими або такими ж ефективними, як колірні простори  $III2I3$ ,  $YUV$ ,  $YIQ$  та  $LSLM$  для розпізнавання облич. Додатково проведені експерименти з використанням бази даних AR, підтверджують узагальнення запропонованих методів CSN.

У статті [8] авторами розглядається проблема продуктивності алгоритмів розпізнавання облич на основі зовнішності. Проблема полягає у негативному впливі змін освітлення. Зазначається, що нормалізація освітлення може значно покращити їхню продуктивність. У дослідженні представлено новий алгоритм для нормалізації освітлення кольорових зображень обличчя. Альbedo обличчя оцінюється на основі зображення обличчя одного кольору та відповідного йому еталонного тривимірного зображення (хмари точок). На відміну від існуючих підходів, алгоритм враховує як ламбертівські, так і дзеркальні відбиття, а також додані та прибрані тіні. Також алгоритм інваріантний до пози та виразу обличчя та може ефективно обробляти випадок кількох протяжних джерел світла. Підхід базується на моделі освітлення Фонга. Параметри моделі Фонга, а також кількість, напрямок та інтенсивність домінуючих джерел світла оцінюються автоматично. Блискавість зображення обличчя використовується для оцінки напрямків домінуючих джерел світла. Далі тривимірна модель обличчя трансформується в промені, щоб знайти тіні кожного джерела світла. Інтенсивність джерел світла та параметри моделі освітлення оцінюються шляхом підгонки моделі Фонга до даних шкіри обличчя. Експерименти проведені на складних даних FRGC v2.0, отримано задовільні, на думку

авторів, результати (середня похибка апроксимації становила 6,3% від максимального значення кольору).

У статті [9] автори розглядають проблему розпізнавання дерматологічних захворювань. У роботі пропонується автоматизований метод класифікації дерматологічних захворювань за кольоровими цифровими фотографіями. Ефективність запропонованого підходу забезпечується двома етапами. На 1-му етапі ураження виявляються та отримуються за допомогою методу варіаційного набору рівнів після зменшення шуму та нормалізації інтенсивності. На 2-му етапі ураження класифікуються за допомогою попередньо навченої архітектури DenseNet201 з ефективною функцією втрат. Експериментальні результати показують, що запропонований підхід може класифікувати ураження з високою продуктивністю (точність 95,24%).

У роботі [10] досліджується розпізнавання обличч з проблемами перекриття обличчя та неоднорідного освітлення. Автори звертають увагу на малу кількість досліджень у цій сфері, на відміну від розпізнавання обличчя без перешкод. Основною метою цієї статті є виявлення обличчя на нерухомому зображенні. Була проведена експериментальна робота на зображеннях з проблемою перекриття обличчя та неоднорідного освітлення. Експериментальні зображення взяті з публічного набору даних AR face dataset та набору даних Color FERET. Також був створений ручний набір даних для експериментальних цілей. Зображення в цьому ручному наборі даних були взяті з Інтернету. Дослідження передбачає створення машини, достатньо навченої, щоб вона могла набути людського сприйняття та знань для виявлення, локалізації та розпізнавання обличчя на довільному зображенні з такою ж легкістю, із якою це роблять люди. У статті пропонується ефективний метод розпізнавання обличчя на нерухомих зображеннях в умовах оклюзії та неоднорідного освітлення. Автори представили метод розпізнавання обличчя з використанням комбінації кольорних моделей  $YCbCr$ ,  $HSV$  та  $Lab$ . Запропонований метод покращив результати з точки зору точності, коефіцієнта виявлення, коефіцієнта хибного виявлення та прецизійності.

У статті [11] пропонується новий фреймворк для перетворення портретного фото обличчя на аніме-зображення. На відміну від існуючих методів перетворення, які не визначають конкретні стилі, автори прагнуть синтезувати аніме-обличчя, яке за стилем узгоджується з заданим еталонним аніме-обличчям. Однак, на відміну від типових завдань перетворення, таке перетворення є особливо складним через великі та складні варіації зовнішності серед аніме-облич. Оскільки існуючі методи часто не переносять стилі еталонних аніме-облич на згенеровані або вносять помітні спотворення, автори пропонують новий перетворювач аніме-обличчя на основі GAN, який називається AniGAN. Зокрема, пропонується нова архітектура перетворювача для одночасного перенесення стилів, кольорів, текстури та перетворення локальних форм обличчя на аніме-подібні аналоги на основі стилю еталонного зображення, зберігаючи при цьому глобальну структуру вихідного фотообличчя. Для генератора розроблені нові функції нормалізації для подальшого покращення перетворення зображення та перенесення стилю. Крім того, запропоновано дискримінатор з двома гілками для вивчення розподілів, специфічних для предметної області, через окремі гілки та вивчення спільних розподілів між предметними областями через спільні шари, що допомагає створювати візуально приємні аніме-обличчя та ефективно зменшувати артефакти/спотворення.

Таким чином проведена аналітична робота підтверджує актуальність подальших досліджень у сферах нормалізації кольору обличчя, а також методів нормалізації зображення відносно еталонного.

### 1.3 Постановка задачі дослідження

Таким чином, питання нормалізації кольору обличчя відносно еталону є актуальним завданням, враховуючи потреби сьогодення та зацікавленість науковців у розвитку даного напрямку.

Об'єктом дослідження є методи нормалізації кольору обличчя на зображенні.

Метою дослідження є порівняння методів нормалізації кольору обличчя відносно еталонного зображення.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз методів нормалізації кольору обличчя відносно еталону, виокремивши їх переваги та недоліки;
- провести аналіз літературних джерел, щодо питання використання методів нормалізації;
- провести дослідження обраних методів, сформувавши покроковий алгоритм для кожного з них;
- візуалізувати алгоритми обраних методів за допомогою блок-схем;
- розробити програмний застосунок, що дозволить реалізувати обрані методи та порівняти точність нормалізації;
- порівняти результати оцінок схожості нормалізованих зображень та еталонів.

## 2 ОСОБЛИВОСТІ ВИБРАНИХ МЕТОДІВ НОРМАЛІЗАЦІЇ ОБЛИЧЧЯ НА ЗОБРАЖЕННІ

### 2.1 Метод Histogram matching

#### 2.1.1 Основна ідея методу Histogram matching

Histogram matching – це один із базових але сучасних методів для розв’язання проблеми нормалізації кольору облич на зображенні відносно еталону. Основна ідея методу полягає у приведенні гістограми аналізованого зображення до еталонного за рахунок математичних перетворень.

Приклад гістограми зображення наведено на рисунку 2.1 [12].

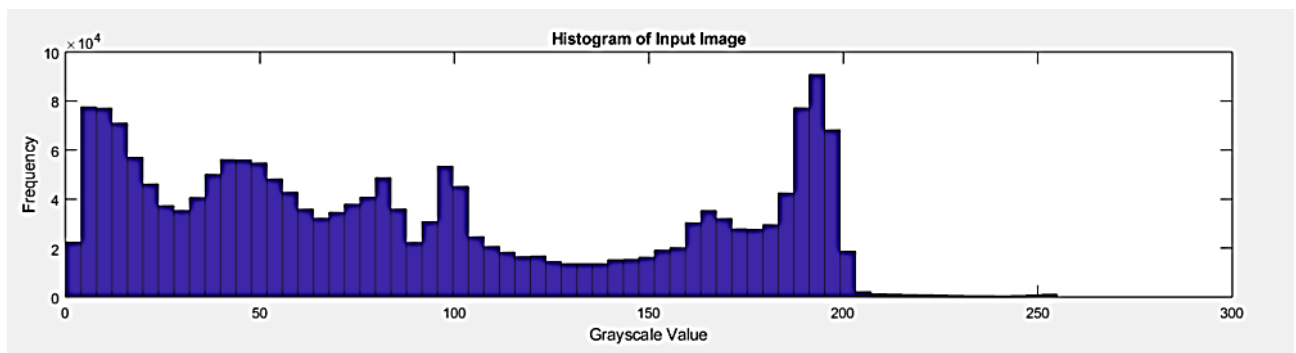


Рисунок 2.1 – Приклад гістограми аналізованого зображення

Особливістю такого підходу можна вважати те, що замість простої заміни значень пікселю на зображенні «А» на відповідні пікселі зображення «Б», метод, аналізуючи та підлаштовуючи гістограми, враховує глобальні характеристики. Наприклад, контраст або тональний баланс, що робить результати більш стабільними і дозволяє уникнути критичних спотворень зображення.

Також такий підхід мінімізує появу на зображенні небажаних шумів. Перенесення глобальних характеристик може серйозно вплинути на результат при наявності яскраво виражених ознак у еталонного зображення, наприклад, таких як поганий контраст чи перевага певного кольору.

### 2.1.2 Алгоритм методу Histogram matching

Метод Histogram matching дозволяє перенести глобальні характеристики з одного зображення на інше, за допомогою вирівнювання показників гістограм зображень:

Крок 1 (опціонально). Попередня обробка зображень. При обробці різних зображень або коли еталон та аналізоване зображення мають великі відмінності у розмірах чи інших показниках, то додатково краще попередньо їх привести до певного «нормального» стану в якому зображення будуть мати найменші відмінності у параметрах, які не стосуються кольору.

Крок 2. Підрахунок гістограм. Для аналізованого ( $S$ ) та еталонного ( $R$ ) зображень будують гістограми  $h_S$  та  $h_R$ , де по осі  $Y$  розташовують кількість пікселів, а по осі  $X$  – рівень яскравості. Приклад гістограми зображений на рисунку 2.2 [13].

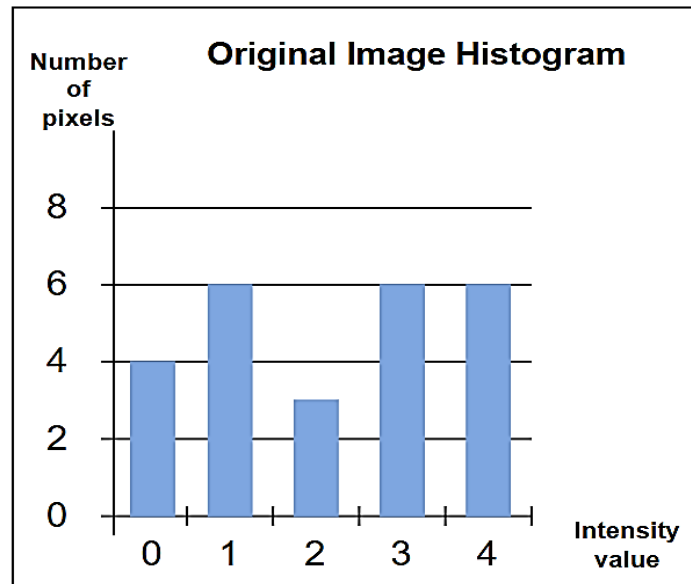


Рисунок 2.2 – Приклад гістограми

Крок 3. Обчислення функції густини ймовірності (Probability Density Function, PDF). Після побудови гістограм для кожного зображення необхідно розрахувати функцію густини ймовірності

$$PDF(i) = \frac{i}{N}, \quad (2.1)$$

де  $i$  – це рівень яскравості;

$N$  – відповідна кількість пікселів.

Крок 4. Обчислення кумулятивної функції розподілу (Cumulative Distribution Function, CDF). Після розрахунку функції густини, необхідно обчислити кумулятивну функцію розподілу кожного зображення  $i$  за такою формулою

$$CDF(i) = \sum_{j=0}^i PDF(j), \quad (2.2)$$

де  $i$  – це рівень яскравості.

Крок 5. Побудова таблиці відповідності значень (Lookup Table, LUT). Для перенесення характеристик від еталонного зображення до аналізованого необхідно побудувати таблицю відповідності значень.

Таблиця заповнюється за таким принципом: для кожного рівня  $i$  на зображенні  $R$  необхідно знайти таке значення рівня  $j$  на зображенні  $S$ , щоб  $CDF(i)$  було менше чи дорівнювало  $CDF(j)$ . Коли таке значення знайдено, то  $LUT[i] = j$ .

Крок 6. Застосування  $LUT$  до вихідного зображення. Завершальним кроком нормалізації буде перенесення відповідних значень з зображення  $R$  на зображення  $S$ .

Нове значення кожного пікселю на зображенні розраховується так:

$$G(p) = LUT(CDF(p)), \quad (2.3)$$

де  $p$  – це значення пікселю на зображенні.

Алгоритм методу Histogram matching наведено на рисунку 2.3.

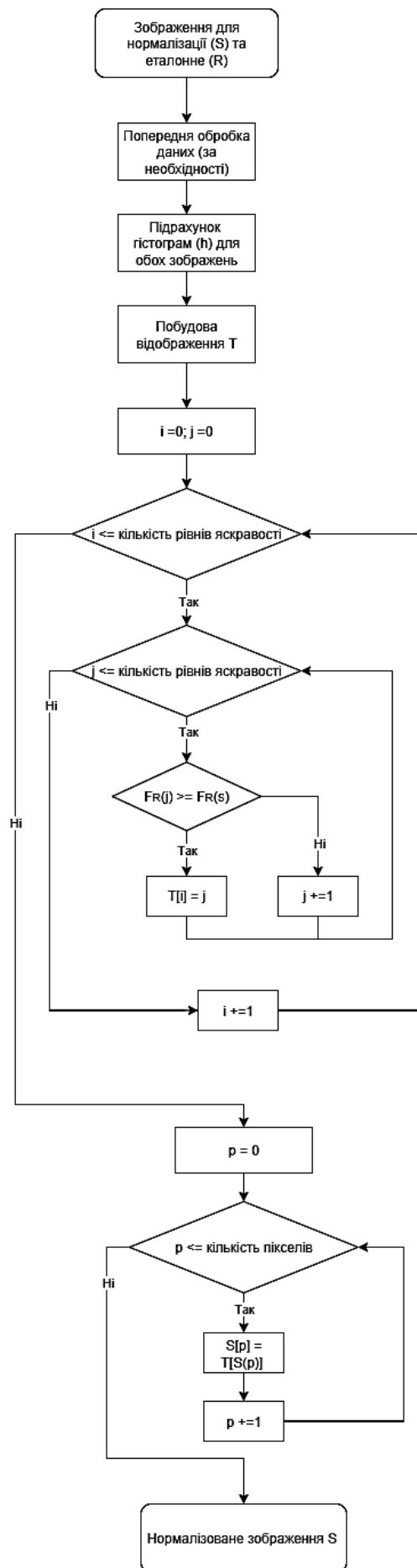


Рисунок 2.3 – Покроковий алгоритм методу Histogram matching

## 2.2 Метод Color transfer

### 2.2.1 Основна ідея методу Color transfer

Color transfer – це один з класичних представників методів перенесення кольору між зображеннями. Основна його ідея у початковій конвертації зображення у *Lab*-простір з подальшим перенесенням значень еталонного простору на аналізований. *Lab*-простір – це спосіб передачі кольорової гами шляхом трьох змінних:  $L$  – яскравість,  $a$  – відношення зеленого до червоного,  $b$  – відношення жовтого до синього.

Для перенесення значень між зображеннями використовуються загальні значення  $\mu$  – середнє та  $\sigma$  – середнє квадратичне відхилення. Тобто кольорова гама кожного пікселя аналізованого зображення у просторі *Lab* буде змінена відповідно до цих загальних значень еталону.

Ефективність такого підходу полягає у високій швидкості обробки, оскільки не треба змінювати кожен піксель на індивідуально розраховані значення. Також через простоту підходу до перенесення характеристик, легше модифікувати алгоритм, змінивши ступінь модифікації.

Важливим недоліком є глобальність розрахування характеристик, що виключає вплив індивідуальних особливостей невеликих областей на зображенні, що може призводити до спотворень. Наприклад, до такого явища як «сіра вуаль», коли зображення може втратити контрастність. Іншою проблемою можуть стати області, які занадто темні, яскраві або на яких якийсь колір суттєво переважає інші кольори на зображенні. Через глобальність характеристик їх застосування може призвести до некоректного перенесення і спотворення кольорової гами результату.

Також недоліком є й глобальність застосування самого методу. Глобальні характеристики, застосовані до кожного з пікселів, не враховують особливостей певних областей, зокрема, частин обличчя, через що навіть відсутність спотворень не гарантує, що необхідні для ідентифікації особи частини обличчя будуть добре видні на вихідному зображенні.

## 2.2.2 Алгоритм методу Color transfer

Метод Color transfer конвертує зображення з кольорового простору  $RGB$  у  $Lab$ , трансформації якого відносно еталону дають можливість отримати нормалізоване зображення. У методі є обов'язкове обмеження на вхідні дані: вхідні зображення мають бути у кольоровому просторі  $RGB$ . Також зображення можуть бути різного розміру.

Наступні кроки описують алгоритм методу:

Крок 1. Попередня обробка зображень.

Якщо вхідні дані знаходяться не у кольоровому просторі  $RGB$ , то необхідно конвертувати зображення з відповідного простору. Найчастіше характеристики пікселю у просторі  $RGB$  знаходяться у проміжку  $[0, 255]$ , де  $C \in \{R, G, B\}$ . У такому випадку треба зробити значення лінійними за допомогою формули:

$$C'(i,j,k) = \frac{C(i,j,k)}{255}, \quad (2.4)$$

де  $i$  – координати пікселю за віссю  $x$ ;

$j$  – координати пікселю за віссю  $y$ ;

$k$  – тип каналу ( $k \in \{R, G, B\}$ );

$C'$  – значення пікселю для каналу  $k'$ ;

$C$  – вихідне значення пікселю.

Після цього додатково необхідно прибрати гаму за допомогою:

$$C_{lin} = \begin{cases} \frac{C'}{12,92} & C' \leq 0,04045 \\ \left(\frac{C'+0,055}{1,055}\right)^{2,4} & C' > 0,04045 \end{cases}, \quad (2.5)$$

де  $C' \in \{R', G', B'\}$ ;

$C_{lin}$  – лінійне значення пікселю для каналу.

У випадку, коли зображення занадто темне чи світле, або наявні інші спотворення, рекомендується позбутися їх до приведення зображення до лінійного виду.

Крок 2. Приведення до простору  $LMS$ .

Для конвертації зображення до простору  $Lab$  необхідно отримати проміжне значення у просторі  $LMS$ . Значення кожного пікселю у новому просторі можна отримати за такою формулою:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0,3811 & 0,5783 & 0,0402 \\ 0,1967 & 0,7244 & 0,0782 \\ 0,0241 & 0,1288 & 0,8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \quad (2.6)$$

де  $L, M, S$  – відповідні значення простору  $LMS$ ;

$R, G, B$  – лінійні значення простору  $RGB$ .

Крок 3. Нейтралізація асиметрії даних (опціонально).

Для підвищення точності даних можна привести їх до логарифмічного виду за допомогою:

$$k' = \log k, \quad (2.7)$$

де  $k'$  – логарифмізоване значення  $k$  ( $k \in \{l, a, b\}$ ).

Крок 4. Приведення до простору  $Lab$ .

Для перенесення характеристик кольорової гама необхідно отримати значення зображення у просторі  $Lab$ . Для конвертації використовується формула:

$$\begin{bmatrix} l \\ a \\ b \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}, \quad (2.8)$$

де  $l, a, b$  – відповідні значення змінних простору  $Lab$ ;

$L, M, S$  – відповідні лінійні значення змінних простору  $LMS$ .

Крок 5. Розрахунок глобальних характеристик.

Після отримання значення усіх пікселів в просторі  $Lab$  необхідно обрахувати значення середнього математичного очікування ( $\mu$ ) та середнього квадратичного відхилення ( $\sigma$ ).

Розрахувати характеристики можна за формулами:

$$\mu^{(c)} = \frac{1}{N} \sum_{i \in N} I(i), \quad (2.9)$$

$$\sigma^{(c)} = \sqrt{\frac{1}{N} \sum_{i \in N} (I(i) - \mu)^2}, \quad (2.10)$$

де  $c \in \{l, a, b\}$ ;

$N$  – кількість пікселів на зображенні;

$I$  – вектор значень пікселів у каналі  $c$ .

Крок 6. Перенесення простору з еталонного на аналізоване зображення.

Для перетворення кожного з пікселів аналізованого зображення використовується така формула:

$$I_{S'}^{(c)} = \left( \frac{I_S^{(c)} - \mu_S^{(c)}}{\max(\sigma_S^{(c)}, \varepsilon)} \right) \sigma_T^{(c)} + \mu_T^{(c)}, \quad (2.11)$$

де  $c$  – значення каналу ( $c \in \{l, a, b\}$ );

$\mu_R$  – математичне очікування для зображення  $R$ ;

$\sigma_R$  – середнє квадратичне відхилення для зображення  $R$  ( $R \in [S, T]$ );

$S$  – аналізоване зображення;

$T$  – еталонне зображення;

$\varepsilon = 10^{-6}$ .

Крок 7. Зворотна конвертація у простір  $LMS$ .

Для отримання нормалізованого зображення, змінені пікселі  $S'$  необхідно обернути з простору  $Lab$  на  $RGB$ . Проміжне перетворення у простір  $LMS$  можна отримати за формулою:

$$\begin{bmatrix} L' \\ M' \\ S' \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & 0 \end{bmatrix} \begin{bmatrix} l' \\ a' \\ b' \end{bmatrix}, \quad (2.12)$$

де  $l', a', b'$  – відповідні значення змінних простору  $Lab$  на зображенні  $S'$ ;

$L', M', S'$  – відповідні лінійні значення змінних простору  $LMS$  зображення  $S'$ .

Крок 8. Приведення до лінійного вигляду (опціонально).

Якщо була застосована дія на Кроці 3 для зменшення розкидання даних, то для повернення до вихідного виду даних необхідно обернути логарифм за такою формулою:

$$k' = 10^k, \quad (2.13)$$

де  $k'$  – лінійне значення  $k$ , де  $k \in \{l', a', b'\}$ .

Крок 9. Конвертація у простір  $RGB$ .

Для отримання нормалізованого результату простір  $LMS$  треба конвертувати назад у  $RGB$ . Для зображення  $S'$  застосовується така формула:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 4,4679 & -3,5873 & 0,1193 \\ -1,2186 & 2,3809 & -0,1624 \\ 0,0497 & -0,2439 & 1,2045 \end{bmatrix} \begin{bmatrix} L' \\ M' \\ S' \end{bmatrix}, \quad (2.14)$$

де  $L', M', S'$  – відповідні значення простору  $LMS$  зображення  $S'$ ;

$R', G', B'$  – лінійні значення простору  $RGB$  зображення  $S'$ .

Крок 10. Конвертація значення пікселів у нелінійні.

Для отримання даних у тому ж форматі, у якому вони були на вході, необхідно провести дії зворотні Кроку 1: повернути гаму та перевести значення у 8-бітний формат. Повернення гами відбувається за такою формулою:

$$C'(i, j, k) = \begin{cases} 12,92 C_{lin}' & C_{lin}'' \leq 0,0031308 \\ 1,055 C_{lin}'^{1/2,4} - 0,55 & C_{lin}' > 0,0031308 \end{cases} \quad (2.15)$$

де  $i$  – координати пікселю за віссю  $x$ ;

$j$  – координати пікселю за віссю  $y$ ;

$k$  – тип каналу ( $k \in \{R', G', B'\}$ );

$C_{lin}'$  – лінійне значення пікселю за координатами  $i$  та  $j$  для каналу  $k$ ;

$C$  – значення пікселю з повернутою гаммою за координатами  $i$  та  $j$  для каналу  $k$ .

Якщо після повернення гами дані виходять за межі проміжку  $[0, 1]$ , наприклад, є від'ємними, то до всіх пікселів необхідно застосувати таку формулу:

$$C'(i, j, k) = \min(1, \max(0, C'(i, j, k))). \quad (2.16)$$

Для переведення у 8-бітну систему використовується формула:

$$C''(i, j, k) = \text{round}(255 * C'(i, j, k)). \quad (2.17)$$

Для дотримання меж на проміжку  $[0, 255]$  застосовується формула:

$$C''(i, j, k) = \min(255, \max(0, C''(i, j, k))). \quad (2.18)$$

Після завершення всіх етапів, отримані значення окремих пікселів конвертуються для відображення у вигляді зображення.

Покроковий алгоритм методу Color transfer наведено на рисунку 2.4.

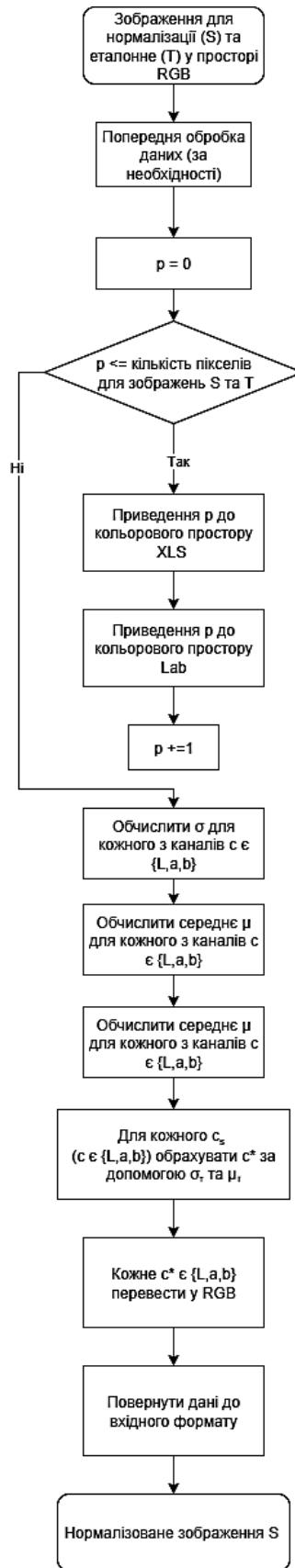


Рисунок 2.4 – Покроковий алгоритм методу Color transfer

## 2.3 Метод Color transfer на основі Face Parsing

### 2.3.1 Основна ідея методу Color transfer на основі Face Parsing

Color transfer на основі Face Parsing – це метод, що використовує описаний раніше метод, працюючи не з усім зображенням загалом, а розбиваючи його на окремі частини. Наприклад, при роботі з обличчям можна виділити його основні зони, такі як ніс, рот, очі, вуха, підборіддя. Це надає можливість виключити області, які можуть бути неважливими для конкретних задач. Наприклад, найважливіші області для проблеми розпізнавання обличчя є очі, ніс, та рот.

Після виділення областей, до кожної з них окремо застосовується обраний метод нормалізації кольору. У варіанті методу, що розглядається у межах роботи, використовується метод Color transfer, але можна використовувати будь який інший.

Основні переваги та недоліки підходу співпадають з відповідними характеристиками обраного методу нормалізації кольору обличчя. Особливістю методу є робота з зображенням по частинах, а не загалом, що може нівелювати негативний ефект від перенесення глобальних характеристик з еталонного зображення, шляхом відсіювання непотрібних областей обличчя, обрахунок яких найчастіше дає похибку у характеристиках або ж більшу точність підрахунку характеристик для відповідної області, ігноруючи дуже яскраві чи темні області на інших частинах зображення.

З недоліків варто зауважити неузгодженість нормалізованих областей обличчя. Оскільки характеристики для кожної області вираховуються окремо і не мають жодного узгодження чи глобальної характеристики, то після завершення нормалізації можуть з'явитися шви, які показують, у яких границях було виділено сегмент.

### 2.3.2 Алгоритм методу Color transfer на основі Face Parsing

Метод Color transfer на основі Face Parsing виділяє області на обличчі і кожну з них конвертує з кольорового простору  $RGB$  у  $Lab$ , колаж трансформацій яких відносно еталону дає можливість отримати нормалізоване зображення. Якщо на зображенні наявні кілька облич, то обробляється те, де впевненість моделі найбільша; якщо обличчя не знайдено, то повертається вхідне зображення без змін.

Для реалізації методу, що розглядається у межах роботи, використана модель для розпізнавання областей обличчя Media Pipe Face Mesh [14].

Наступні кроки описують алгоритм методу:

Крок 1. Попереднє оброблення даних.

Для кожного зображення  $I$  модель повертає  $N$ -нормалізованих двовимірних координат за формулою:

$$P = \{(x_i, y_i) \in [0,1]^2\}_{i=1}^N. \quad (2.19)$$

Необхідно перетворити їх на піксельні координати за допомогою формул:

$$X_i = [x_i W], \quad (2.20)$$

$$Y_i = [y_i H], \quad (2.21)$$

де  $H$  – висота зображення;

$W$  – ширина зображення.

Сформувати набір семантичних зон  $K$ .

Використана модель використовує такий набір:

$$K = \{\text{lips, left\_eye, right\_eye, left\_brow, right\_brow, nose, face\_oval}\}.$$

Крок 2. Побудова полігональних масок зон.

Для кожної зони  $k$  необхідно обрахувати збір опорних точок у піксельному просторі за формулою:

$$Q(k) = \{(X_i, Y_i) : i \in I_k\}, \quad (2.22)$$

де  $I_k$  – пікселі, що належать області  $k$  на зображенні  $I$ .

Потім для кожної зони  $k$  необхідно обрахувати опуклу оболонку ( $H_k$ ).

Після отримання опуклої оболонки необхідно визначити бінарну маску-полігон за формулою:

$$M_k(x, y) = \begin{cases} 1, & \text{якщо } (x, y) \in H_k \\ 0, & \text{якщо } (x, y) \notin H_k \end{cases}. \quad (2.23)$$

Крок 3. Морфологічна стабілізація дрібних зон (опціонально).

При наявності зон з великою деталізацією, наприклад, очей, губ, носу, брів рекомендується застосувати ерозію з радіусом  $[1, 2]$  для усунення захоплення невеликих зон фону. До вибірки областей  $K_{small}$  використовується формула:

$$M'_k = \varepsilon_{p_k}(M_k), \quad (2.24)$$

де  $k \in K_{small}$ .

Крок 4. Побудова маски шкіри.

Для відокремлення шкіри від елементів носу, очей, брів, необхідно створити маску шкіри і за її допомогою трансформувати маски для кожної з зон. Для створення маски шкіри застосовується формула:

$$M_{skin} = M_0 \vee \dots \vee M_{n-1}, \quad (2.25)$$

де  $n$  – кількість елементів множини  $K_{small}$ .

Для отримання первинної маски шкіри використовується така формула:

$$M'_{skin} = (M_{face\_oval} - M_{skin}). \quad (2.26)$$

Після отримання первинної маски шкіри значення  $M_{face\_oval}$  необхідно замінити на  $M'_{skin}$ .

Крок 5. Формування комплекту масок.

На цьому етапі методу проводиться перевірка площини області для відкидання занадто маленьких областей. Після цього необхідно сформувати комплект готових масок.

Попередній список зон до фільтрації:

$$S = \{M_{lips}, M_{left\_eye}, M_{right\_eye}, M_{left\_brow}, M_{right\_brow}, M_{nose}, M_{skin}\}.$$

Фільтрація областей відбувається за формулою:

$$\sum_{(x,y) \in I} M(x,y) \geq T, \quad (2.27)$$

де  $T$  – кількість пікселів які обмежують площину зони;

$$M \in S.$$

Крок 6. Формування порядку областей.

Для зменшення ймовірності появи спотворень під час нормалізації кольору аналізованої області рекомендується сформувати порядок у якому області будуть проходити нормалізацію. Спершу нормалізуються дрібні та контрастні структури (губи, очі, брови), щоб їхній колір не був захоплений іншими областями під час подальшого злиття. Ніс обробляється перед «шкірою», бо поєднує як гладкі, так і різкі переходи. «Шкіра» обробляється останньою як найбільша зона, що «підбирає» залишкові відмінності і мінімізує глобальні шви. У разі перетину масок пріоритет має пізніша в черзі область (за вказаним порядком), що мінімізує видимі шви на великих ділянках шкіри.

Для розглянутого переліку масок  $S$  порядок буде наступним:

lips  $\rightarrow$  left\_eye  $\rightarrow$  right\_eye  $\rightarrow$  left\_brow  $\rightarrow$  right\_brow  $\rightarrow$  nose  $\rightarrow$  skin.

Крок 7. Застосування алгоритму Color transfer.

Для кожної маски області із  $S$  за сформованою чергою застосовується алгоритм, описаний у підрозділі 2.2.

Після отримання нормалізованої області оновлюється підсумкове зображення: пікселі, де  $M_k = 1$ , заміщуються нормалізованими значеннями; поза маскою зберігається попередній стан. Після закінчення нормалізації кольору у всіх областях зображення вважається нормалізованим.

Покроковий алгоритм методу Color transfer на основі Face Parsing наведено у додатку А.

#### 2.4 Формування методики для нормалізації обличчя на зображенні

В основі сучасних систем, що працюють з аналізом та обробкою зображень, зокрема облич, велика увага приділяється точності розпізнавання людини та її обличчя на фото. На точність, у цьому випадку, може впливати велика кількість різних факторів, серед яких, один з основних – світло. За поганого освітлення людина може зовсім не розпізнаватися на зображенні. Через це у системах, що працюють із зображеннями, велика увага має приділятися нормалізації. Формування методики для нормалізації обличчя на зображенні стає важливим етапом попередньої обробки даних.

Основною задачею нормалізації відносно еталону є приведення зображень до уніфікованого виду, який дозволить підвищити точність розпізнавання та подальший аналіз. Методи, які нормалізують зображення відносно еталону, майже завжди базуються на перенесенні певних глобальних характеристик від еталонного зображення на аналізоване.

Одним з них є Histogram matching, що орієнтований на вирівнювання гістограми аналізованого зображення відповідно до еталонного. Зважаючи на те, що при роботі з усім зображенням, метод здатен опрацювати локальні області і досить рівномірно переносити характеристики, він також має недоліки. Серед них, наприклад, ігнорування просторової структури об'єкту та вірогідність створення спотворень при суттєвій різниці між еталоном та аналізованим зображенням.

Серед інших базових методів нормалізації кольору зображень можна виділити метод Color transfer, який базується на перенесенні *Lab*-простору між зображеннями. Переваги методу полягають у швидкості обробки та простоті модифікації. Одним із найбільших недоліків є робота з зображенням у цілому, що спричиняє велику чутливість до «ненормальних» областей на зображенні, наприклад, занадто світлих чи темних зон, або зон із яскраво вираженим кольором.

Розвитком методу можна вважати метод Color transfer на основі Face Parsing. Основна ідея методу полягає у попередньому розбитті на зображенні обличчя на окремі області: ніс, рот, очі, брови, вуха та окремі нормалізації кожної з них. Таким чином, метод нівелює основний недолік Color transfer, виключаючи спотворення нормалізації області «А» через «ненормальний» стан області «Б». Також перевагою методу є можливість працювати лише з важливими для аналізу областями, пришвидшуючи роботу. Важливим недоліком методу можна назвати чутливість до розбиття обличчя на сегменти відносно яскравості зображення, через що початкові області можуть бути невірно ідентифіковані. Також через неузгодженість нормалізації окремих сегментів на зображенні можуть створюватись спотворення: шви між окремими областями.

Після проведення дослідження, сформовано висновок, що методика нормалізації обличчя повинна включати кілька основних правил:

- вибір еталонів із «ідеальними» характеристиками для задачі;
- попередня стандартизація даних (вирівнювання, обрізка зображення);

- попереднє вирівнювання облич;
- розбиття на фрагменти та фільтрація неважливих для задачі ділянок;
- уникнення роботи з глобальними характеристиками зображення.

Практичне використання правил зустрічається у низці комерційних продуктів, наприклад, SDK Luxand FaceSDK та Neurotechnology VeriLook, що використовують модулі попередньої обробки для покращення стабільності розпізнавання у мобільних банківських застосунках і національних системах електронної ідентифікації.

Таким чином, формування методики нормалізації обличчя може показати основні переваги та недоліки аналізованих методів та сформувані перелік пунктів, згідно з якими буде проводитися розроблення нових методів та вирішення прикладних проблем.

## 2.5 Моделювання структури програмного застосунку для нормалізації обличчя на зображенні

Для програмної реалізації застосунку нормалізації обличчя на зображенні необхідно провести попереднє моделювання його структури, щоб майбутній застосунок задовольняв певні вимоги.

Функціональні вимоги до структури програмного застосунку:

- імпорт зображень із локального сховища та отримання даних через API;
- побудова масок зон обличчя (у випадку використання алгоритму Color transfer на основі Face Parsing);
- можливість вибору алгоритму нормалізації;
- локальне збереження результатів нормалізації та їхня передача за допомогою API;
- збереження результатів оцінки точності роботи застосунку.

Нефункціональні вимоги до структури програмного застосунку:

- можливість розширення застосунку: додавання нових метрик для оцінки нових алгоритмів нормалізації;
- безпека під час отримання та відправки фотографій, використання шифрування результатів.

Застосунок не передбачає повністю незалежної роботи. Він буде використовувати один із мікросервісів системи, у якій інший мікросервіс буде надсилати зображення для нормалізації, щоб отримати якісний очікуваний результат. У даній роботі запропонований застосунок є розширенням системи, що розроблена у межах роботи [15] та є апробованою як вебзастосунок банківської комп'ютерної системи.

Змодельований застосунок повинен отримувати поточне зображення користувача від мікросервісу «face-id-microservice» та еталонне з бази даних. Поточне зображення має бути нормалізоване відносно еталону і повернуте мікросервісу як результат. Архітектура застосунку показана на рисунку 2.5.

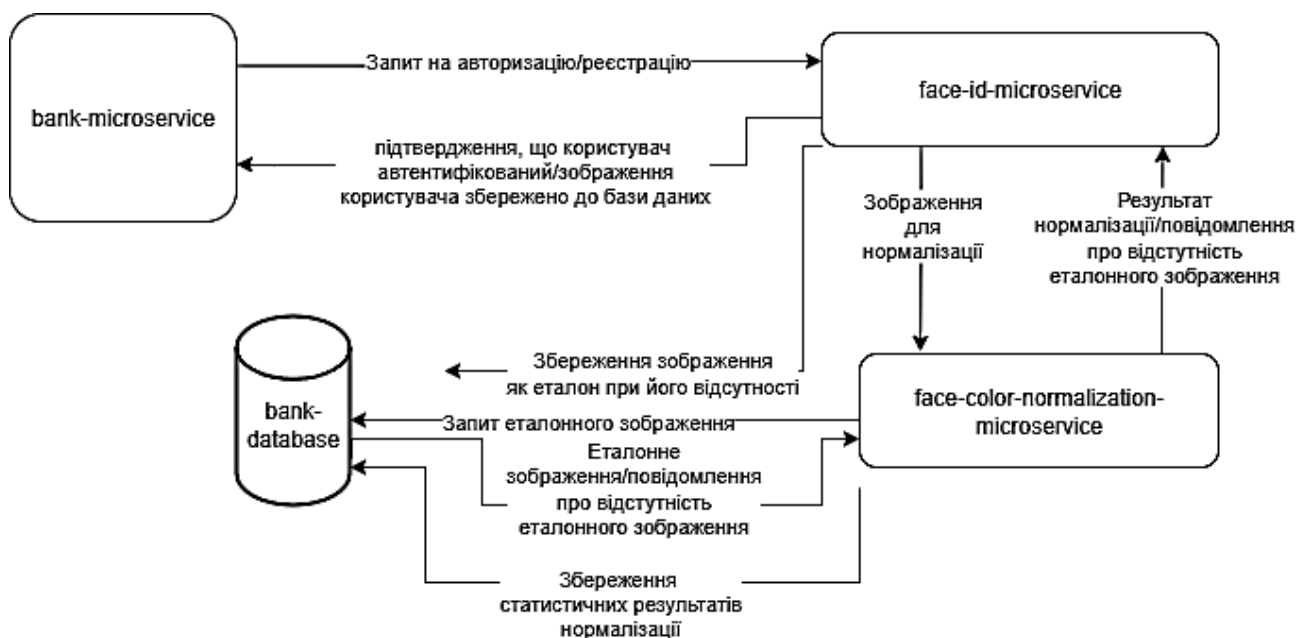


Рисунок 2.5 – Архітектура вебзастосунку банківської системи з мікросервісною реалізацією застосунку для нормалізації обличчя на зображенні

Для збереження статистичних результатів нормалізації створено нову таблицю у базі даних – `normalization_statistics`.

Структура `normalization_statistics` показана у таблиці 2.1.

Таблиця 2.1 – Таблиця `normalization_statistics`

Data	Type	Example
<code>normalization_id</code>	Integer (counter)	1
<code>user_id</code>	Integer	1
<code>statistics_file</code>	Bytea	/9j/4AgcJP2P/wCHcn7E/wD0QHTP/Ay6/wDjtf/Z...

Таблиця `normalization_statistics` зберігає дані про результати нормалізації зображення користувача у вигляді файлу. Поле `normalization_id` є первинним ключем, `user_id` посилається на `user_id` у таблиці `bankuser`.

Загальний вигляд бази даних банківської системи представлений на рисунку 2.6.

Лістинг 2.1 Код для створення таблиці `normalization_statistics`:

```
CREATE TABLE NormalizationStatistics (
    normalization_id SERIAL PRIMARY KEY,
    user_id SERIAL REFERENCES BankUser (user_id),
    statistics_file TEXT not null
);
```

У межах даного дослідження застосунок може проводити нормалізацію наступними методами:

- Histogram matching;
- Color transfer;
- Color transfer на основі Face Parsing.

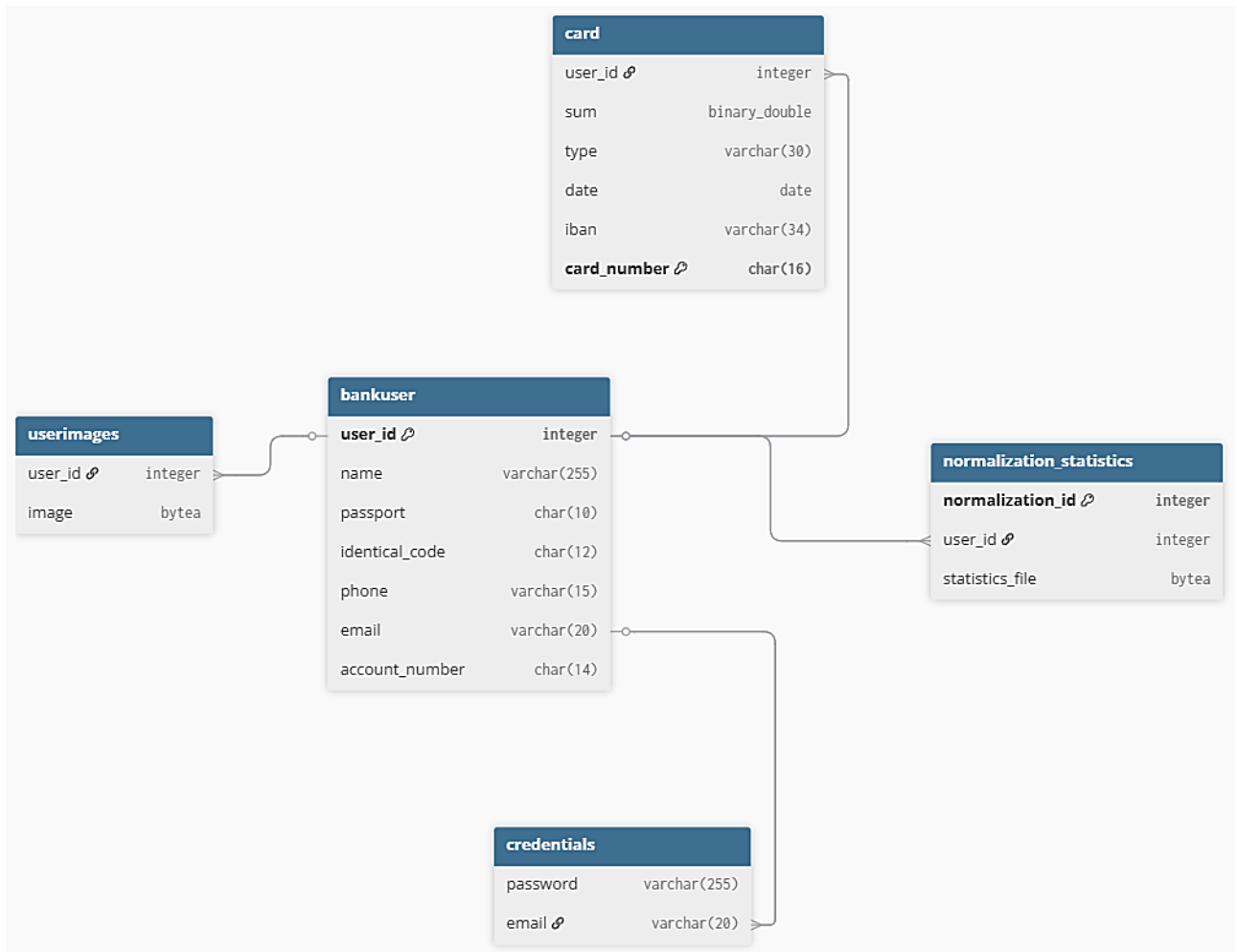


Рисунок 2.6 – Загальний вигляд бази даних банківської системи

Для роботи методів необхідно мінімум два зображення. Для методів Color transfer та Color transfer на основі Face Parsing необхідні зображення у кольоровому просторі *RGB* будь-якого розміру. Для Histogram matching велике значення має розмір зображення, тому загалом рекомендується використовувати два зображення у просторі *RGB* із однаковим розміром. Крім того, можна додати попередню обробку зображень.

Для оцінки точності нормалізації методів використовуватимуться такі метрики:

- SSIM – структурна подібність у відтінках сірого в діапазоні  $[0, 1]$  (чим ближче значення до 1, тим вища подібність до еталонної структури контрасту, текстури або країв);

- Bhattacharyya – відстань між нормованими гістограмами яскравості, канал  $L$  (значення 0 – означає однакові розподіли, якщо менше 0, то краще узгодження тону);
- Gray-World angle – кут між середнім кольором зображення та ідеально «сірим» (менший кут означає кращий баланс білого);
- $\Delta E^*ab$  (CIE76) – середня колірна різниця у просторі  $Lab$  (0 – відсутня відмінність, при значеннях менше 0 – точніше відтворення відтінку, насиченості або яскравості);
- помилка контрасту  $L$  (5%–95%) – модуль різниці міжперцентильного діапазону яскравості (ближче до 0 – подібний глобальний контраст);
- Clipping (black+white) – сума часток пікселів каналу  $L$ , «збитих» у 0 або 255 (значення менші за 0 та більші за 255 означають менше втрат деталей у тіні та світлі).

### 3 ДОСЛІДЖЕННЯ МЕТОДІВ НОРМАЛІЗАЦІЇ ОБЛИЧЧЯ НА ЗОБРАЖЕННІ ЩОДО ВИБРАННОЇ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 3.1 Вибір інструментальних засобів для реалізації вибраних методів

Для ефективної програмної реалізації застосунку критичний вплив має вибір програмного забезпечення. Після аналізу можливостей наявних програмних засобів було прийнято рішення про використання мови програмування Python 3.11 у поєднанні з бібліотеками: OpenCV, NumPy, scikit-image, matplotlib. Усі етапи розробки та моделювання виконано в інтегрованому середовищі PyCharm 2023.3.4.

Бібліотека OpenCV – універсальний інструмент для роботи з зображеннями та проблемами, пов'язаними з комп'ютерним зором. Бібліотека забезпечує можливості зчитування, показу та збереження зображень, конвертацію кольорних просторів, візуалізацію результатів.

Бібліотека NumPy використовувалась для роботи з багатовимірними масивами, зокрема, реалізацією математичних операцій над піксельними даними, розрахунку статистичних даних.

Для реалізації методу Histogram matching застосовано методи з бібліотеки scikit-image щодо роботи з гістограмами яскравості.

Для побудови діаграм за результатами статистичної оцінки використано бібліотеку matplotlib.

Для проведення дослідження за допомогою обраних методів застосовано дані з набору Extended Yale B [16]. Вхідні дані: чорно-білі зображення різних осіб, розмір зображень 192×168 пікселів у форматі .png.

PyCharm – редактор з відкритим кодом, що заснований на мові програмування Python, став ефективним вибором платформи для розроблення та запуску застосунку. Надані інструменти спростили процес розроблення, забезпечивши безперешкодну інтеграцію та оптимізацію робочого процесу.

Оскільки застосунок є мікросервісом, який не має бути напряму доступним користувачу, графічний інтерфейс відсутній. Для тестування використано застосунок Postman [17]. Приклад налаштувань показаний на рисунку 3.1.

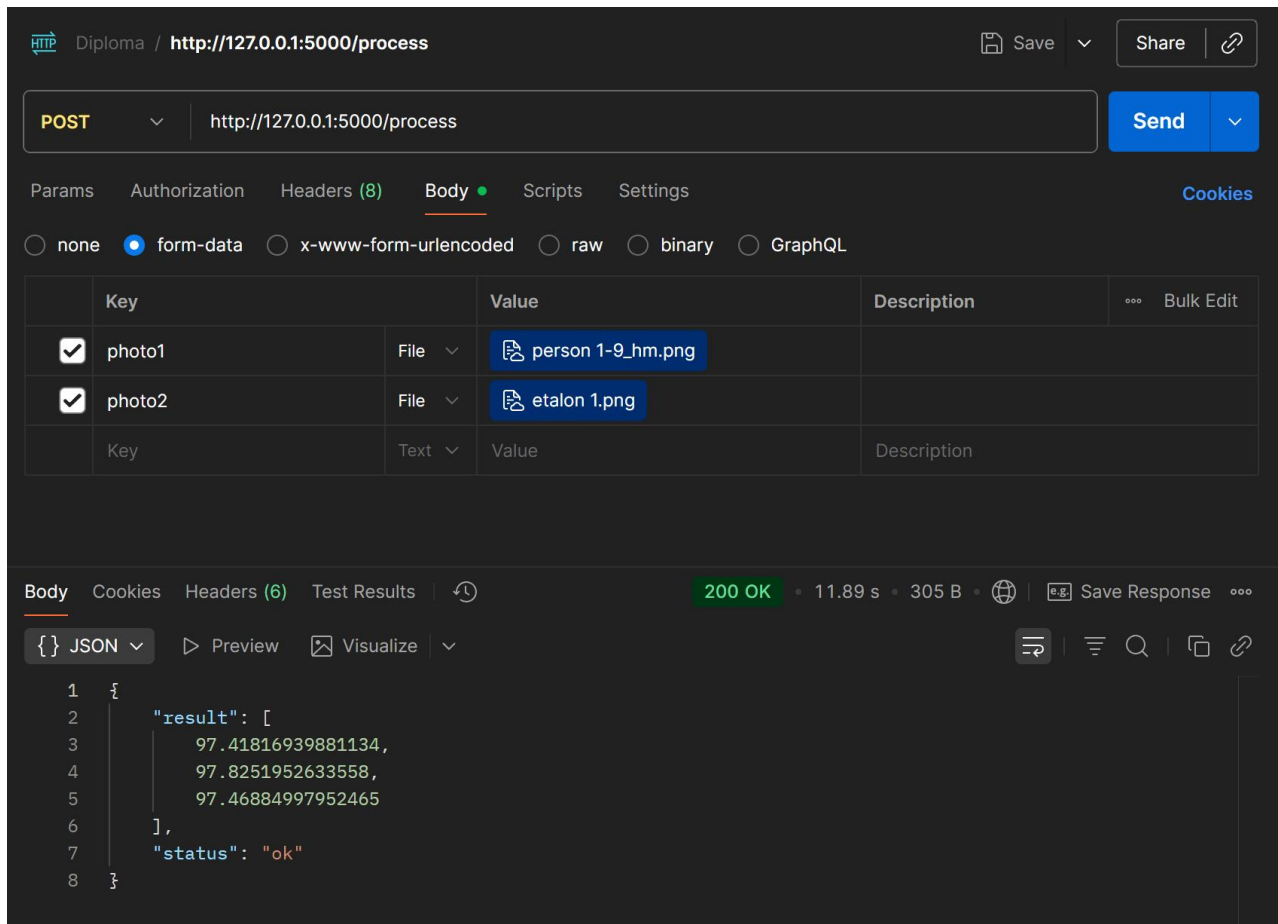


Рисунок 3.1 – Приклад тестувальних даних

## 3.2 Етапи програмної реалізації вибраних методів нормалізації обличчя на зображенні

### 3.2.1 Формування набору даних для тестування

Для тестування методів нормалізації кольору обличчя необхідно сформувати набір даних, який буде включати зображення з різними ступенями освітленості, яскравості, а також фотографії людей різної статі та рас. Для виконання цієї задачі обрано зображення з набору даних Extended Yale B [16].

З набору даних Extended Yale B [16] обрано 3 класи (3 різних людини), кожен з яких містить 57 зображень у форматі .png. Зображення створено шляхом фотографування людей під час обертання лампи навколо них. Обрана кількість зображень дозволяє продемонструвати ступінь освітленості обличчя з більшості можливих кутів. Назва файлу зображення містить характеристики, за яких воно було зроблене, тому для дослідження можна відібрати фото з різних класів, які були створені за однакових умов.

Зображення у обраному наборі були попередньо обрізані до розміру  $168 \times 192$  пікселі та приведені до чорно-білого формату. Такий формат даних найкраще підходить для тестування методів нормалізації кольору обличчя, оскільки обрізані непотрібні зони, які можуть призводити до спотворень через наявність специфічних даних. Чорно-біле зображення також допомагає уникати одного домінуючого кольору. Якщо при використанні програми будуть взяті інші вхідні дані, то вони мають бути приведені до зазначеного формату. Приклади зображень у сформованому наборі даних подано на рисунку 3.2.



Рисунок 3.2 – Приклади зображень класів

Для проведення нормалізації для кожного з класів необхідно обрати еталонне зображення – що показує обличчя в умовах найкращого освітлення. Шляхом експертної оцінки обрано 3 еталони, що наведені на рисунку 3.3.

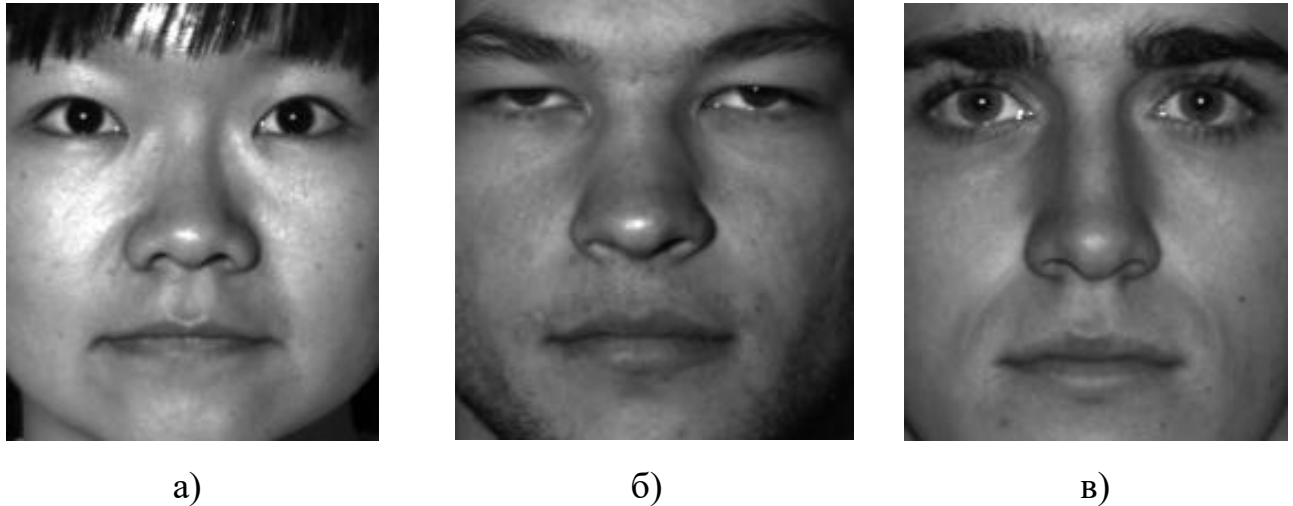


Рисунок 3.3 – Еталонні зображення класів:  
а) еталон класу 1; б) еталон класу 2; в) еталон класу 3

Сформований набір даних дає змогу оцінити методи нормалізації кольору обличчя на зображеннях осіб з унікальними особливостями обличчя на великому різноманітті варіантів освітлення.

### 3.2.2 Програмна реалізація мікросервісної архітектури застосунку

Мікросервісна архітектура – це спосіб побудови архітектури застосунку, яка передбачає використання не одного сервісу для вирішення усіх задач, а мережі з багатьох менших сервісів (мікросервісів), де кожен відповідає за свій етап задачі. Для комунікації мікросервісів у мережі використовуються кінцеві точки (endpoints). До наявних точок у застосунку [15] додано точку /normalize/<method\_type> (реалізацію подано у лістингу 3.1), що приймає фото 2 і скорочену назву методу, нормалізує фото 1 відносно фото 2, повертає результат у форматі base64 або викидає помилку при відсутності файлів.

Також додатково реалізовано допоміжну функцію `save_to_tempfile`, яка формує тимчасовий файл для подальшого оброблення.

Скорочені назви методів:

- `hm` – Histogram matching;
- `ct` – Color transfer;
- `fp` – Color transfer на основі Face Parsing.

Лістинг 3.1 Реалізація кінцевої точки `normalize/<method_type>`:

```
def save_to_tempfile(file_storage):
    filename = secure_filename(file_storage.filename or "image")
    _, ext = os.path.splitext(filename)
    tmp = tempfile.NamedTemporaryFile(delete=False, suffix=ext or ".jpg")
    file_storage.save(tmp.name)
    return tmp.name

@app.route("/normalize/<method>", methods=["POST"])
def normlize(method):
    if "photo1" not in request.files or "photo2" not in request.files:
        return jsonify({"error": "Upload two files with keys 'photo1' and
'photo2'"}), 400
    f1 = request.files["photo1"]
    f2 = request.files["photo2"]
    if f1.filename == "" or f2.filename == "":
        return jsonify({"error": "Both files must have filenames"}), 400
    path1 = path2 = None
    try:
        path1 = save_to_tempfile(f1)
        path2 = save_to_tempfile(f2)
        result = None
        if method == 'hm':
            result = run_histmatch(path1, path2)
```

```

elif method == 'ct':
    result = run_color_transfer(img1, img2)
elif method == 'fp':
    result = run_face_parsing(img1, img2)
return jsonify({"result": result}), 200
except Exception as e:
    return jsonify({"error": str(e)}), 500
finally:
    for p in (path1, path2):
        if p and os.path.exists(p):
            try:
                os.remove(p)
            except OSError:
                pass

```

Результат виклику кінцевої точки `/normalize/<method_type>` для методу Histogram matching подано на рисунку 3.4.

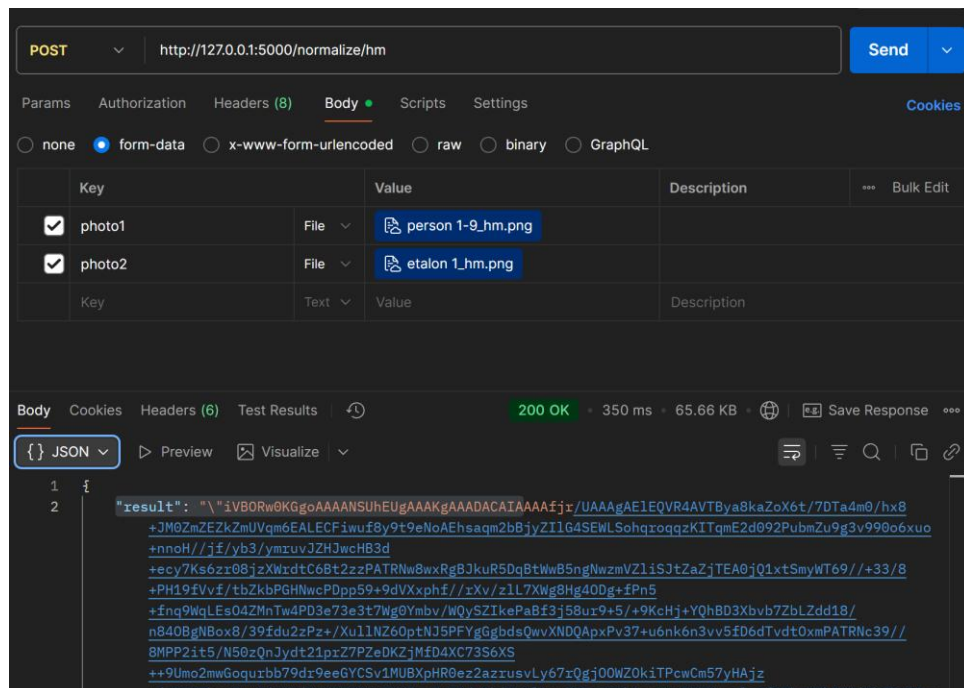


Рисунок 3.4 – Результат виклику кінцевої точки `/normalize/<method_type>` для методу Histogram matching

### 3.2.3 Реалізація допоміжних функцій

Для реалізації усіх трьох методів сформовано спільні функції:

- `read_bgr(path)` – читає зображення з диска у форматі BGR через OpenCV або кидає помилку, якщо файл не прочитано;
- `bgr_to_labL(img_bgr)` – конвертує BGR простір у простір *Lab* і повертає значення окремих каналів *l, a, b*;
- `labL_to_bgr(L, A, B)` – поєднує значення каналів *l, a, b* у *Lab* і конвертує назад у простір BGR;
- `ndarray_to_json_image(img_bgr, ext=".png", **encode_params)` – конвертує зображення з простору BGR у формат base64.

Повний код розглянутих функцій наведено у лістингу 3.2.

Лістинг 3.2. Реалізація допоміжних функцій:

```
def ndarray_to_json_image(img_bgr, ext=".png", **encode_params):
    ok, buf = cv2.imencode(ext, img_bgr, list(encode_params.items()))
    if not ok:
        raise ValueError("imencode failed")
    b64 = base64.b64encode(buf).decode("ascii")
    return b64

def list_images(folder: Path) -> List[Path]:
    return [p for p in sorted(folder.glob("*")) if p.suffix.lower() in IMG_EXTS]

def read_bgr(path: Path) -> np.ndarray:
    img = cv2.imread(str(path), cv2.IMREAD_COLOR)
    if img is None:
        raise RuntimeError(f"Не вдалося прочитати: {path}")
    return img
```

```
def bgr_to_lab(img_bgr: np.ndarray) -> np.ndarray:
    return cv2.cvtColor(img_bgr, cv2.COLOR_BGR2LAB)
```

```
def lab_to_bgr(img_lab: np.ndarray) -> np.ndarray:
    return cv2.cvtColor(img_lab, cv2.COLOR_LAB2BGR)
```

### 3.2.4 Структура реалізації методу Histogram matching

Для реалізації методу написано такі функції:

- `cdf_from_images_L(L_list)` – за списком каналів `L_list` рахує сумарну гістограму та нормалізовану `CDF`;
- `cdf_from_image_L(L)` – рахує нормалізовану `CDF` для одного каналу `L`;
- `build_mapping(source_cdf, target_cdf)` – будує `LUT` шляхом інтерполяції `source_cdf` на шкалі `target_cdf`;
- `apply_lut(L, lut)` – застосовує `LUT` до каналу `L`;
- `run_histmatch(analyz_photo_path, etalon_photo_path)` – запускає нормалізацію зображення відповідно до шляху `analyz_photo_path` згідно з еталонним відповідно до шляху `etalon_photo_path`.

Повна реалізація метода наведена у лістингу 3.3.

Лістинг 3.3 Реалізація методу Histogram matching:

```
import json
from pathlib import Path
from typing import List, Tuple
import cv2
import numpy as np

from tools import read_bgr, bgr_to_lab, lab_to_bgr, ndarray_to_json_image
```

```

def _save_png(path: Path, bgr: np.ndarray) -> None:
    path.parent.mkdir(parents=True, exist_ok=True)
    cv2.imwrite(str(path), bgr)

def _cdf_from_image_L(L: np.ndarray, smooth_sigma: float | None = 1.0) ->
np.ndarray:
    if smooth_sigma and smooth_sigma > 0:
        L_use = cv2.GaussianBlur(L, (0, 0), smooth_sigma)
    else:
        L_use = L
    hist, _ = np.histogram(L_use.ravel(), bins=256, range=(0, 256),
density=False)
    cdf = hist.cumsum().astype(np.float64)
    cdf /= cdf[-1] if cdf[-1] > 0 else 1.0
    return cdf

def _cdf_from_images_L(L_list: List[np.ndarray], smooth_sigma: float | None
= 1.0) -> np.ndarray:
    if smooth_sigma and smooth_sigma > 0:
        L_all = [cv2.GaussianBlur(L, (0, 0), smooth_sigma).ravel() for L in
L_list]
    else:
        L_all = [L.ravel() for L in L_list]

    all_vals = np.concatenate(L_all, axis=0)
    hist, _ = np.histogram(all_vals, bins=256, range=(0, 256), density=False)
    cdf = hist.cumsum().astype(np.float64)
    cdf /= cdf[-1] if cdf[-1] > 0 else 1.0
    return cdf

```

```
def _build_mapping(source_cdf: np.ndarray, target_cdf: np.ndarray) ->
np.ndarray:
```

```
    lut = np.searchsorted(target_cdf, source_cdf, side="left")
```

```
    lut = np.clip(lut, 0, 255).astype(np.uint8)
```

```
    return lut
```

```
def _apply_lut(L: np.ndarray, lut: np.ndarray) -> np.ndarray:
```

```
    return cv2.LUT(L, lut)
```

```
def run_histmatch(
```

```
    analiz_photo_path: Path,
```

```
    etalon_photo_path: Path,
```

```
    out_dir: Path,
```

```
    smooth_sigma: float | None = 1.0,
```

```
    apply_clahe: bool = False,
```

```
    clahe_clip: float = 2.0,
```

```
    clahe_grid: Tuple[int, int] = (8, 8),
```

```
) -> str:
```

```
    out_dir = Path(out_dir)
```

```
    out_dir.mkdir(parents=True, exist_ok=True)
```

```
    etalon_bgr = read_bgr(etalon_photo_path)
```

```
    eL, eA, eB = cv2.split(bgr_to_lab(etalon_bgr))
```

```
    etalon_cdf = _cdf_from_image_L(eL, smooth_sigma=smooth_sigma)
```

```
    bgr = read_bgr(analiz_photo_path)
```

```
    L, A, B = cv2.split(bgr_to_lab(bgr))
```

```
    src_cdf = _cdf_from_image_L(L, smooth_sigma=smooth_sigma)
```

```
    lut = _build_mapping(src_cdf, etalon_cdf)
```

```
L_new = _apply_lut(L, lut)
```

```
if apply_clahe:
```

```
    clahe = cv2.createCLAHE(clipLimit=clahe_clip, tileGridSize=clahe_grid)
```

```
    L_new = clahe.apply(L_new)
```

```
fixed = lab_to_bgr(cv2.merge([L_new, A, B]))
```

```
out_file = out_dir / f"{Path(analyz_photo_path).stem}_hm.png"
```

```
_save_png(out_file, fixed)
```

```
payload = ndarray_to_json_image(fixed, ext=".png")
```

```
json_str = json.dumps(payload, ensure_ascii=False)
```

```
return json_str
```

### 3.2.5 Структура реалізації методу Color transfer

Реалізація методу Color transfer, що наведена у лістингу 3.4, містить такі функції:

- *stats\_for\_refs*(*refs\_bgr*) – для еталонного BGR-зображення рахує середнє та стандартне відхилення по всіх пікселях у просторі *Lab*;

- *reinhard\_color\_transfer*(*src\_bgr*, *tgt\_mean*, *tgt\_std*) – реалізація Reinhard color transfer: нормує *Lab*-канали джерела за його середнім математичним очікуванням ( $\mu$ ) та середнім квадратичним відхиленням ( $\sigma$ ) і масштабує цілі; робить відсікання (clipping) до [0, 255] і повертає BGR;

- *run\_color\_transfer*(*analyz\_photo\_path*, *etalon\_photo\_path*) – нормалізує колір з фото відповідно до шляху *analyz\_photo* згідно зображення відповідно до шляху *etalon\_photo\_path* повертає результат у base64.

## ЛІСТИНГ 3.4 Реалізація методу Color transfer:

```

import json
import cv2
import numpy as np
import tools

def _stats_for_refs(refs):
    if isinstance(refs, np.ndarray):
        refs = [refs]
    elif not isinstance(refs, (list, tuple)):
        raise TypeError(f"_stats_for_refs expected ndarray or list of ndarrays,
got {type(refs)}")

    labs_list = []
    for i, img in enumerate(refs):
        if not isinstance(img, np.ndarray):
            raise TypeError(f"_stats_for_refs item {i} is {type(img)}, expected
np.ndarray")

        lab = tools.bgr_to_lab(img).reshape(-1, 3).astype(np.float32)
        labs_list.append(lab)

    concat = np.concatenate(labs_list, axis=0)
    mean = concat.mean(axis=0).astype(np.float32)
    std = concat.std(axis=0).astype(np.float32)
    std = np.maximum(std, 1e-6).astype(np.float32)
    return mean, std

def _reinhard_color_transfer(src_bgr: np.ndarray,
                             tgt_mean: np.ndarray, tgt_std: np.ndarray) -> np.ndarray:
    lab = tools.bgr_to_lab(src_bgr).astype(np.float32)
    src_mean = lab.reshape(-1, 3).mean(axis=0)

```

```

src_std = lab.reshape(-1, 3).std(axis=0, ddof=0)
src_std[src_std < 1e-6] = 1.0
lab[..., 0] = (lab[..., 0] - src_mean[0]) / src_std[0] * tgt_std[0] +
tgt_mean[0]
lab[..., 1] = (lab[..., 1] - src_mean[1]) / src_std[1] * tgt_std[1] +
tgt_mean[1]
lab[..., 2] = (lab[..., 2] - src_mean[2]) / src_std[2] * tgt_std[2] +
tgt_mean[2]

lab = np.clip(lab, 0, 255).astype(np.uint8)
return tools.lab_to_bgr(lab)

def run_color_transfer(analyz_photo_path, etalon_photo_path, out_path) ->
str:
refs_bgr = tools.read_bgr(etalon_photo_path)
tgt_mean, tgt_std = _stats_for_refs(refs_bgr)

bgr = tools.read_bgr(analyz_photo_path)
fixed = _reinhard_color_transfer(bgr, tgt_mean, tgt_std)
cv2.imwrite(out_path + f"\\{analyz_photo_path.stem}_ct.png", fixed)
payload = tools.ndarray_to_json_image(fixed, ext=".png")
json_str = json.dumps(payload)
return json_str

```

### 3.2.6 Структура реалізації методу Color transfer на основі Face Parsing

Для методу Color transfer на основі Face Parsing, поданому у лістингу 3.5, сформовано такі функції:

- `_get_lab_stats_masked(L, A, B, mask)` – обчислює середнє математичне очікування та середнє квадратичне відхилення для трьох каналів по пікселях, що входять у `mask`; якщо маска порожня, то застосовуються глобальні статистики зображення;

- `_reinhard_masked(L, A, B, mean_src, std_src, mean_ref, std_ref, mask)` – застосовує Reinhard color transfer тільки в межах *mask*: нормує за статистиками джерела і масштабує під статистики еталону; проводить відсікання до `[0, 255]`;
- `_indices_from_connections(connections)` – витягує унікальні індекси вершин з набору ребер MediaPipe (список «точок-індексів»);
- `_mask_from_indices(img_shape, landmarks, indices, shrink_erode=0)` – будує бінарну маску, як опуклу оболонку вибраних landmark-точок, за потреби зменшує маску ерозією;
- `_get_landmarks(image_bgr)` – знаходить 468-точкові face landmarks через MediaPipe FaceMesh; повертає landmarks або None;
- `_build_part_masks(img)` – повертає 7 масок обличчя: `skin`, `lips`, `left_eye`, `right_eye`, `left_brow`, `right_brow`, `nose`;
- `run_face_parsing(analyz_photo, etalon_photo_path)` переносить колір із еталонного обличчя на аналізоване за методом Reinhard, але окремо для 7 зон (*lips*, *left\_eye*, *right\_eye*, *left\_brow*, *right\_brow*, *nose*, *face\_oval*), а потім повертає результат як base64.

Лістинг 3.5 Реалізація методу Color transfer на основі Face Parsing:

```

import json
from typing import List, Tuple, Dict
import cv2
import mediapipe as mp
import numpy as np
import tools

def _get_lab_stats_masked(L: np.ndarray, A: np.ndarray, B: np.ndarray,
mask: np.ndarray) -> Tuple[np.ndarray, np.ndarray]:
    idx = np.where(mask > 0)
    if idx[0].size == 0:

```

```

    return np.array([L.mean(), A.mean(), B.mean()]), np.array([L.std(),
A.std(), B.std()])

```

```

    return np.array([L[idx].mean(), A[idx].mean(), B[idx].mean()]),
np.array([L[idx].std(), A[idx].std(), B[idx].std()])

```

```

def _reinhard_masked(L, A, B, mean_src, std_src, mean_ref, std_ref, mask):
    eps = 1e-6
    L_out, A_out, B_out = L.astype(np.float32), A.astype(np.float32),
B.astype(np.float32)
    idx = np.where(mask > 0)

    L_out[idx] = (L_out[idx] - mean_src[0]) * (std_ref[0] / (std_src[0] + eps))
+ mean_ref[0]
    A_out[idx] = (A_out[idx] - mean_src[1]) * (std_ref[1] / (std_src[1] + eps))
+ mean_ref[1]
    B_out[idx] = (B_out[idx] - mean_src[2]) * (std_ref[2] / (std_src[2] + eps))
+ mean_ref[2]

    L_out = np.clip(L_out, 0, 255).astype(np.uint8)
    A_out = np.clip(A_out, 0, 255).astype(np.uint8)
    B_out = np.clip(B_out, 0, 255).astype(np.uint8)
    return L_out, A_out, B_out

```

```

_mp_face_mesh = mp.solutions.face_mesh

```

```

def _indices_from_connections(connections) -> List[int]:
    idx = set()
    for a, b in connections:
        idx.add(a);
        idx.add(b)

```

```

return sorted(idx)

def _mask_from_indices(img_shape, landmarks, indices: List[int],
shrink_erode: int = 0) -> np.ndarray:
    h, w = img_shape[:2]
    if not indices or landmarks is None:
        return np.zeros((h, w), dtype=np.uint8)

    pts = []
    for i in indices:
        x = int(np.clip(landmarks[i].x * w, 0, w - 1))
        y = int(np.clip(landmarks[i].y * h, 0, h - 1))
        pts.append([x, y])
    pts = np.array(pts, dtype=np.int32)
    if pts.shape[0] < 3:
        return np.zeros((h, w), dtype=np.uint8)

    hull = cv2.convexHull(pts)
    mask = np.zeros((h, w), dtype=np.uint8)
    cv2.fillConvexPoly(mask, hull, 255)

    if shrink_erode > 0:
        k = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (shrink_erode,
shrink_erode))
        mask = cv2.erode(mask, k)
    return mask

def _get_landmarks(image_bgr: np.ndarray):
    with _mp_face_mesh.FaceMesh(
        static_image_mode=True,

```

```

    max_num_faces=1,
    refine_landmarks=True,
    min_detection_confidence=0.5
) as fm:
    res = fm.process(cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB))
    if not res.multi_face_landmarks:
        return None
    return res.multi_face_landmarks[0].landmark

def _build_part_masks(img: np.ndarray) -> Dict[str, np.ndarray]:
    h, w = img.shape[:2]
    lms = _get_landmarks(img)
    masks = {k: np.zeros((h, w), dtype=np.uint8) for k in
        ["skin", "lips", "left_eye", "right_eye", "left_brow", "right_brow",
        "nose"]}

    if lms is None:
        return masks

    lips_idx = _indices_from_connections(_mp_face_mesh.FACEMESH_LIPS)
    left_eye_idx =
    _indices_from_connections(_mp_face_mesh.FACEMESH_LEFT_EYE)
    right_eye_idx =
    _indices_from_connections(_mp_face_mesh.FACEMESH_RIGHT_EYE)
    left_brow_idx =
    _indices_from_connections(_mp_face_mesh.FACEMESH_LEFT_EYEBROW)
    right_brow_idx =
    _indices_from_connections(_mp_face_mesh.FACEMESH_RIGHT_EYEBROW)
    nose_idx =
    _indices_from_connections(_mp_face_mesh.FACEMESH_NOSE)

```

```

    oval_idx =
    _indices_from_connections(_mp_face_mesh.FACEMESH_FACE_OVAL)

    masks["lips"] = _mask_from_indices(img.shape, lms, lips_idx,
    shrink_erode=2)
    masks["left_eye"] = _mask_from_indices(img.shape, lms, left_eye_idx,
    shrink_erode=2)
    masks["right_eye"] = _mask_from_indices(img.shape, lms, right_eye_idx,
    shrink_erode=2)
    masks["left_brow"] = _mask_from_indices(img.shape, lms, left_brow_idx,
    shrink_erode=1)
    masks["right_brow"] = _mask_from_indices(img.shape, lms,
    right_brow_idx, shrink_erode=1)
    masks["nose"] = _mask_from_indices(img.shape, lms, nose_idx,
    shrink_erode=1)

    face_mask = _mask_from_indices(img.shape, lms, oval_idx,
    shrink_erode=0)

    features = (masks["lips"] | masks["left_eye"] | masks["right_eye"]
    | masks["left_brow"] | masks["right_brow"] | masks["nose"])
    skin = cv2.subtract(face_mask, features)
    k = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
    skin = cv2.morphologyEx(skin, cv2.MORPH_OPEN, k)

    masks["skin"] = skin
    return masks

def run_face_parsing(analyz_photo_path, etalon_photo_path, out_path):
    ref_means: Dict[str, list] = {k: [] for k in

```

```

["skin", "lips", "left_eye", "right_eye", "left_brow",
"right_brow", "nose"]}

```

```

ref_stds: Dict[str, list] = {k: [] for k in
["skin", "lips", "left_eye", "right_eye", "left_brow",
"right_brow", "nose"]}

```

```

ref_bgr = tools.read_bgr(etalon_photo_path)

```

```

L, A, B = cv2.split(tools.bgr_to_lab(ref_bgr))

```

```

masks = _build_part_masks(ref_bgr)

```

```

for part, m in masks.items():

```

```

    mean, std = _get_lab_stats_masked(L, A, B, m)

```

```

    ref_means[part].append(mean)

```

```

    ref_stds[part].append(std)

```

```

mean_ref: Dict[str, np.ndarray] = {}

```

```

std_ref: Dict[str, np.ndarray] = {}

```

```

for part in ref_means:

```

```

    if len(ref_means[part]) > 0:

```

```

        mean_ref[part] = np.mean(ref_means[part], axis=0)

```

```

        std_ref[part] = np.mean(ref_stds[part], axis=0)

```

```

bgr = tools.read_bgr(analyz_photo_path)

```

```

L, A, B = cv2.split(tools.bgr_to_lab(bgr))

```

```

masks = _build_part_masks(bgr)

```

```

fixed = None

```

```

if all(m.sum() == 0 for m in masks.values()):

```

```

    fixed = bgr.copy()

```

```

else:

```

```

    L_new, A_new, B_new = L.copy(), A.copy(), B.copy()

```

```

order = ["lips", "left_eye", "right_eye", "left_brow", "right_brow",
"nose", "skin"]
for part in order:
    m = masks[part]
    mean_s, std_s = _get_lab_stats_masked(L_new, A_new, B_new, m)
    L_new, A_new, B_new = _reinhard_masked(L_new, A_new, B_new,
mean_s, std_s, mean_ref[part], std_ref[part], m)

lab = cv2.merge([L, A, B])
fixed = tools.lab_to_bgr(lab)
cv2.imwrite(out_path + f"\\{analyz_photo_path.stem}_fp.png", fixed)

payload = tools.ndarray_to_json_image(fixed, ext=".png")
json_str = json.dumps(payload)
return json_str

```

3.3 Застосування методів нормалізації обличчя на зображенні до вибраної предметної області

У даному підрозділі наведено приклади нормалізації кольору обличчя відносно еталону методу Histogram matching, Color transfer, Color transfer на основі Face Parsing.

Для демонстрації результатів взято 5 зображень з кожного із 3 класів і результат нормалізації кожного з них наведеними методами.

Зображення нормалізуються відповідно до еталонів на рисунку 3.2.

Для кожного зображення після нормалізації одним з методів створено колаж «до» та «після», де перше зображення – аналізоване, а друге – результат нормалізації (рис. 3.5 – рис. 3.49).



Рисунок 3.5 – Результат нормалізації зображення 11 з класу 1 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.6 – Результат нормалізації зображення 13 з класу 1 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.7 – Результат нормалізації зображення 16 з класу 1 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.8 – Результат нормалізації зображення 29 з класу 1 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.9 – Результат нормалізації зображення 39 з класу 1 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.10 – Результат нормалізації зображення 11 з класу 1 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.11 – Результат нормалізації зображення 13 з класу 1 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.12 – Результат нормалізації зображення 16 з класу 1 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.13 – Результат нормалізації зображення 29 з класу 1 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)

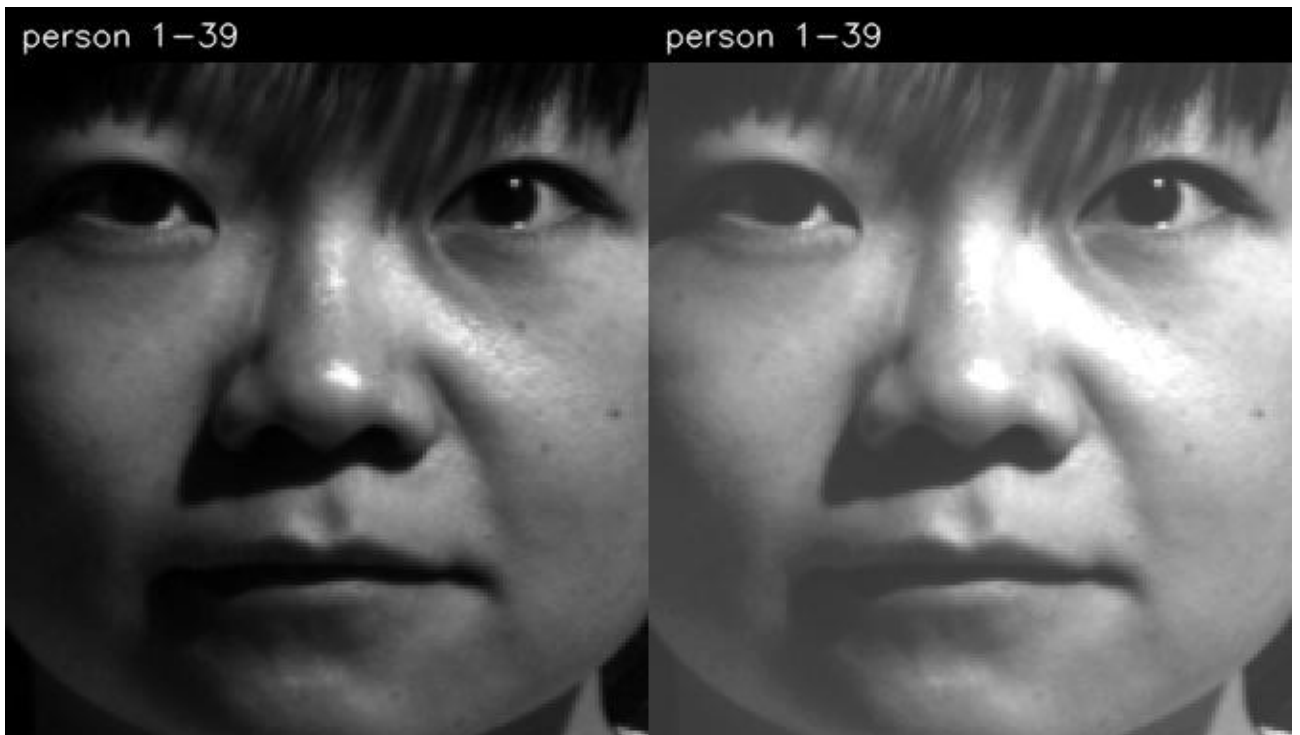


Рисунок 3.14 – Результат нормалізації зображення 39 з класу 1 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.15 – Результат нормалізації зображення 11 з класу 1 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.16 – Результат нормалізації зображення 13 з класу 1 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.17 – Результат нормалізації зображення 16 з класу 1 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.18 – Результат нормалізації зображення 29 з класу 1 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.19 – Результат нормалізації зображення 39 з класу 1 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.20 – Результат нормалізації зображення 11 з класу 2 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.21 – Результат нормалізації зображення 13 з класу 2 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.22 – Результат нормалізації зображення 16 з класу 2 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.23 – Результат нормалізації зображення 29 з класу 2 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)

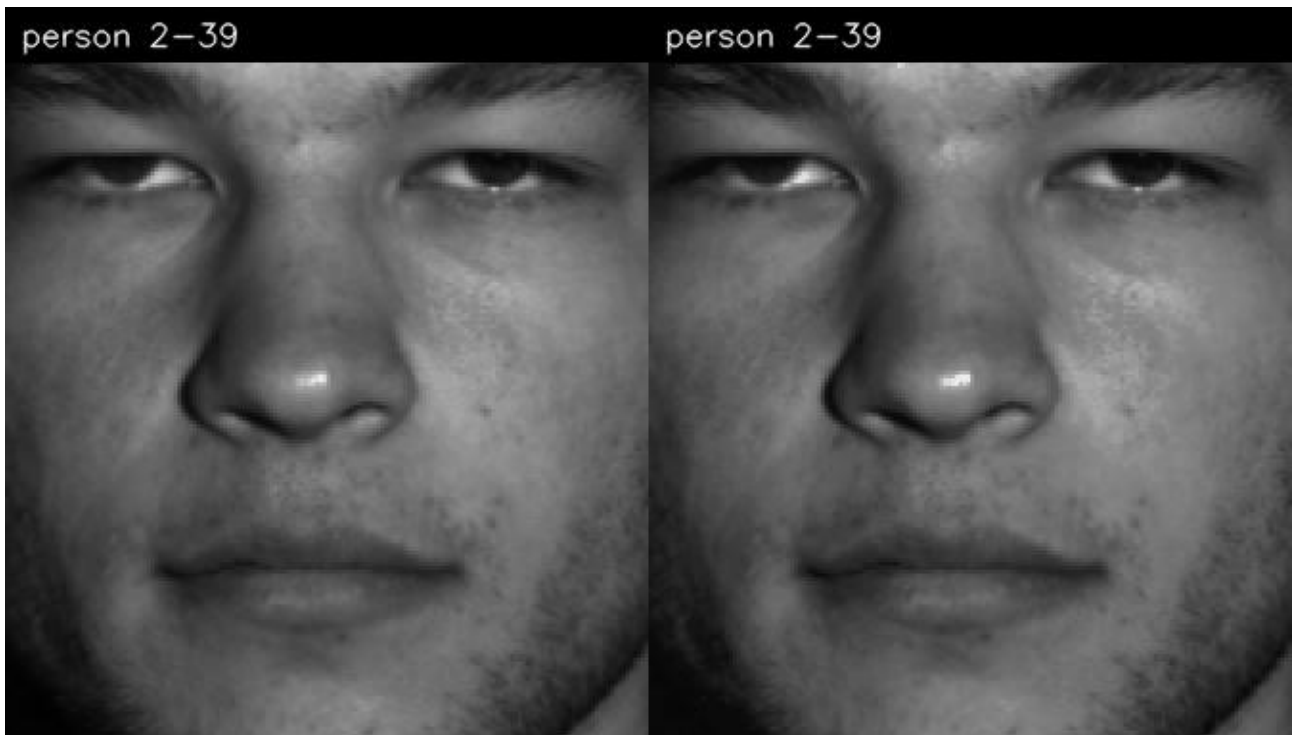


Рисунок 3.24 – Результат нормалізації зображення 39 з класу 2 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)

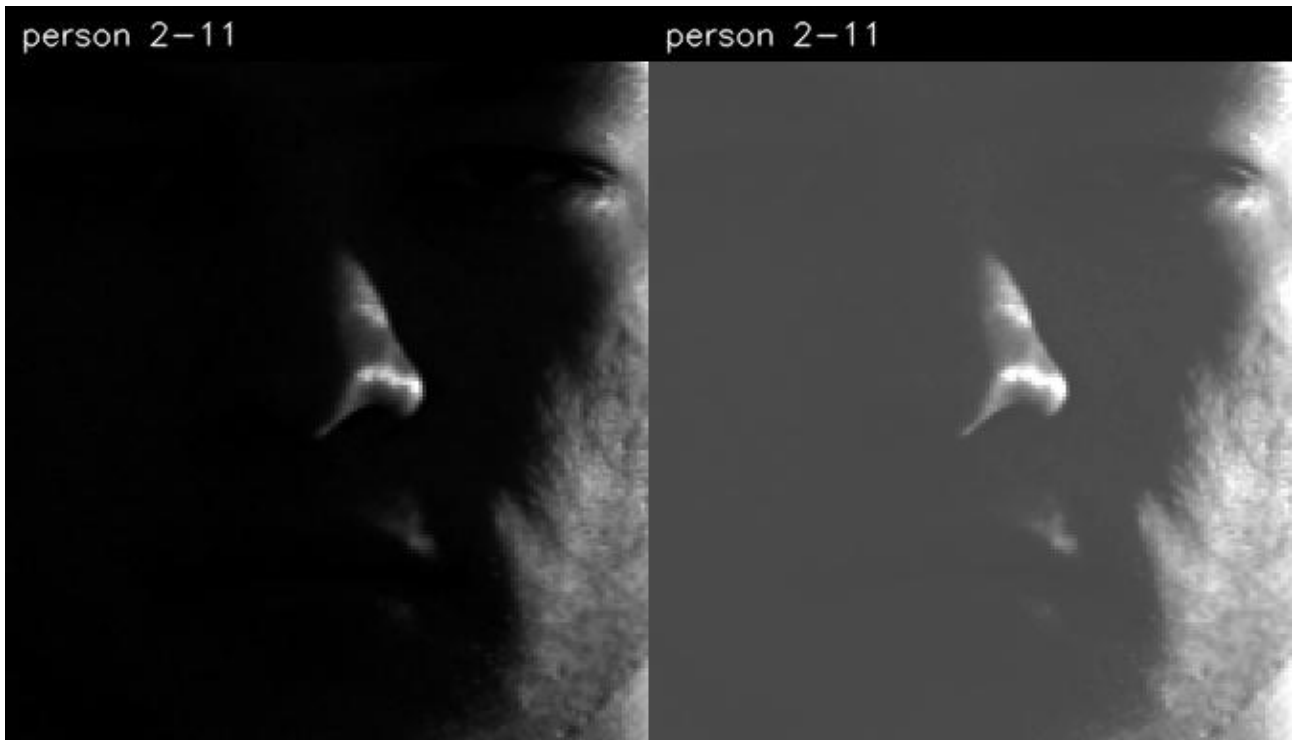


Рисунок 3.25 – Результат нормалізації зображення 11 з класу 2 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.26 – Результат нормалізації зображення 13 з класу 2 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.27 – Результат нормалізації зображення 16 з класу 2 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)

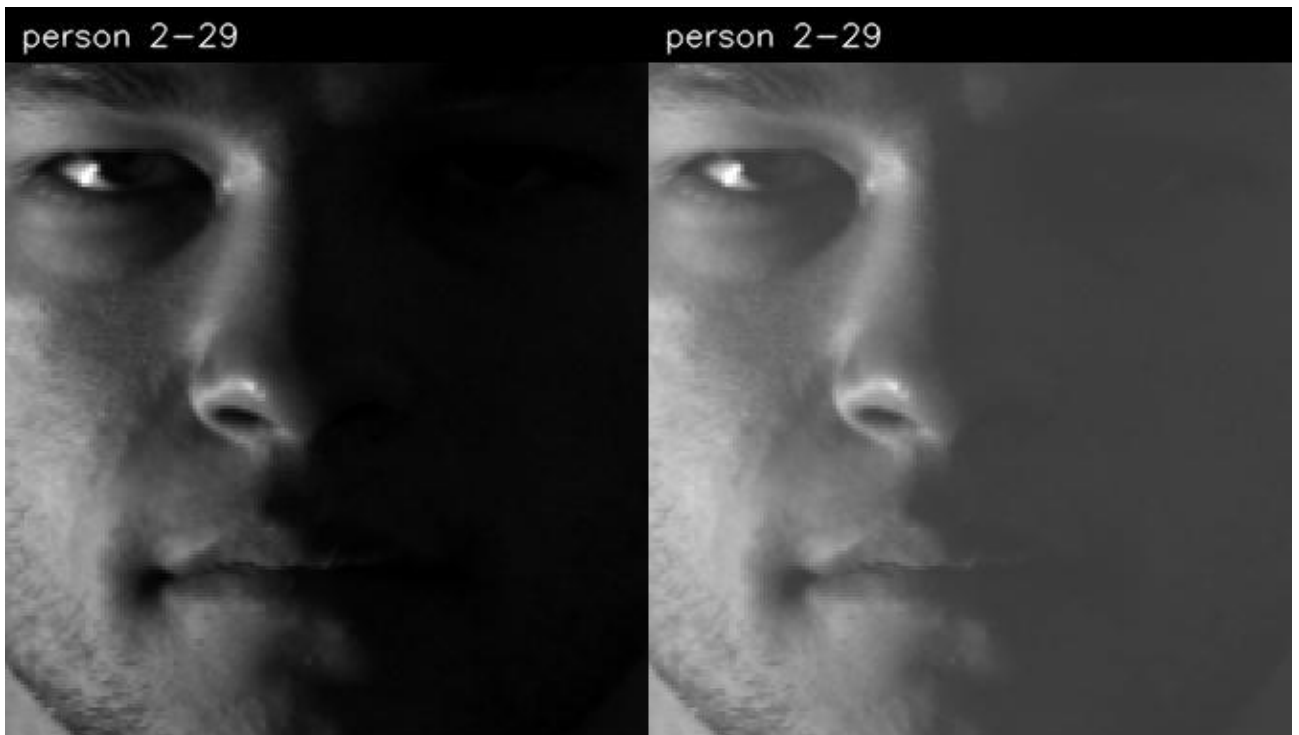


Рисунок 3.28 – Результат нормалізації зображення 29 з класу 2 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.29 – Результат нормалізації зображення 39 з класу 2 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.30 – Результат нормалізації зображення 11 з класу 2 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.31 – Результат нормалізації зображення 13 з класу 2 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.32 – Результат нормалізації зображення 16 з класу 2 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)

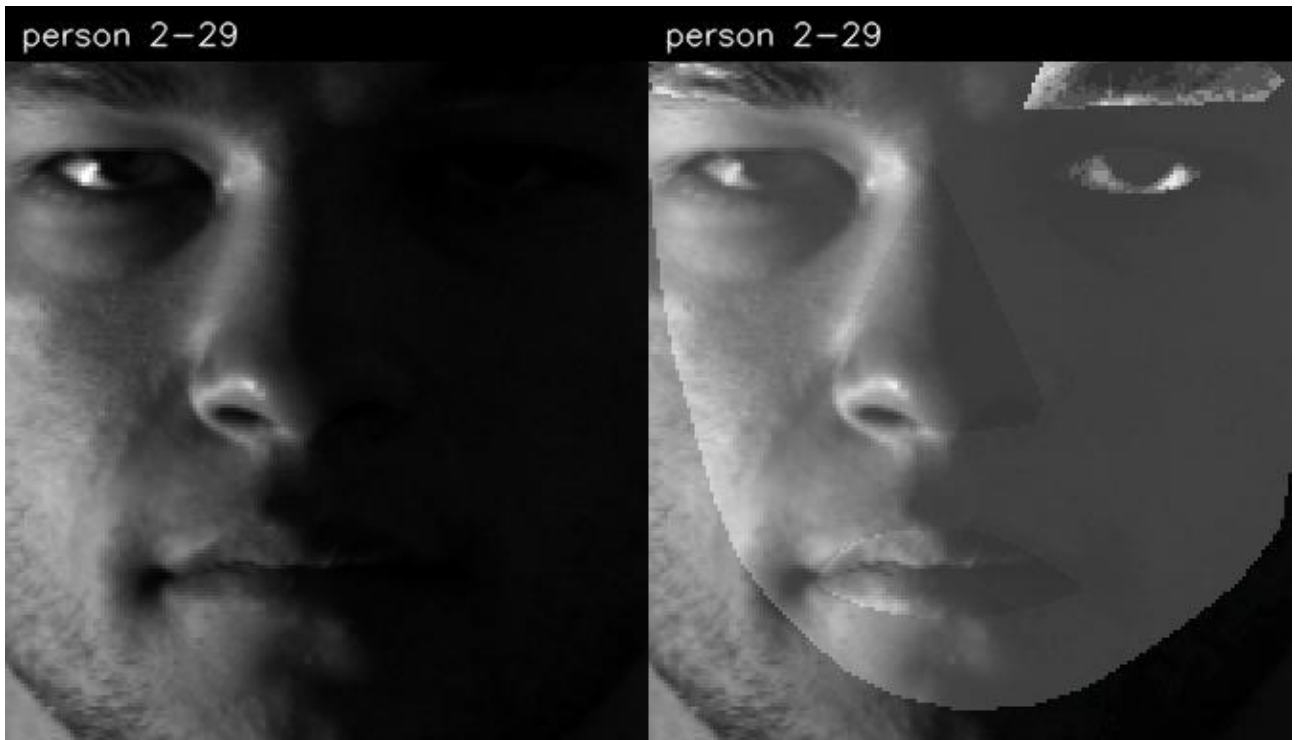


Рисунок 3.33 – Результат нормалізації зображення 29 з класу 2 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.34 – Результат нормалізації зображення 39 з класу 2 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.35 – Результат нормалізації зображення 11 з класу 3 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.36 – Результат нормалізації зображення 13 з класу 3 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.37 – Результат нормалізації зображення 16 з класу 3 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.38 – Результат нормалізації зображення 29 з класу 3 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.39 – Результат нормалізації зображення 39 з класу 3 методом Histogram matching (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.40 – Результат нормалізації зображення 11 з класу 3 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.41 – Результат нормалізації зображення 13 з класу 3 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.42 – Результат нормалізації зображення 16 з класу 3 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)

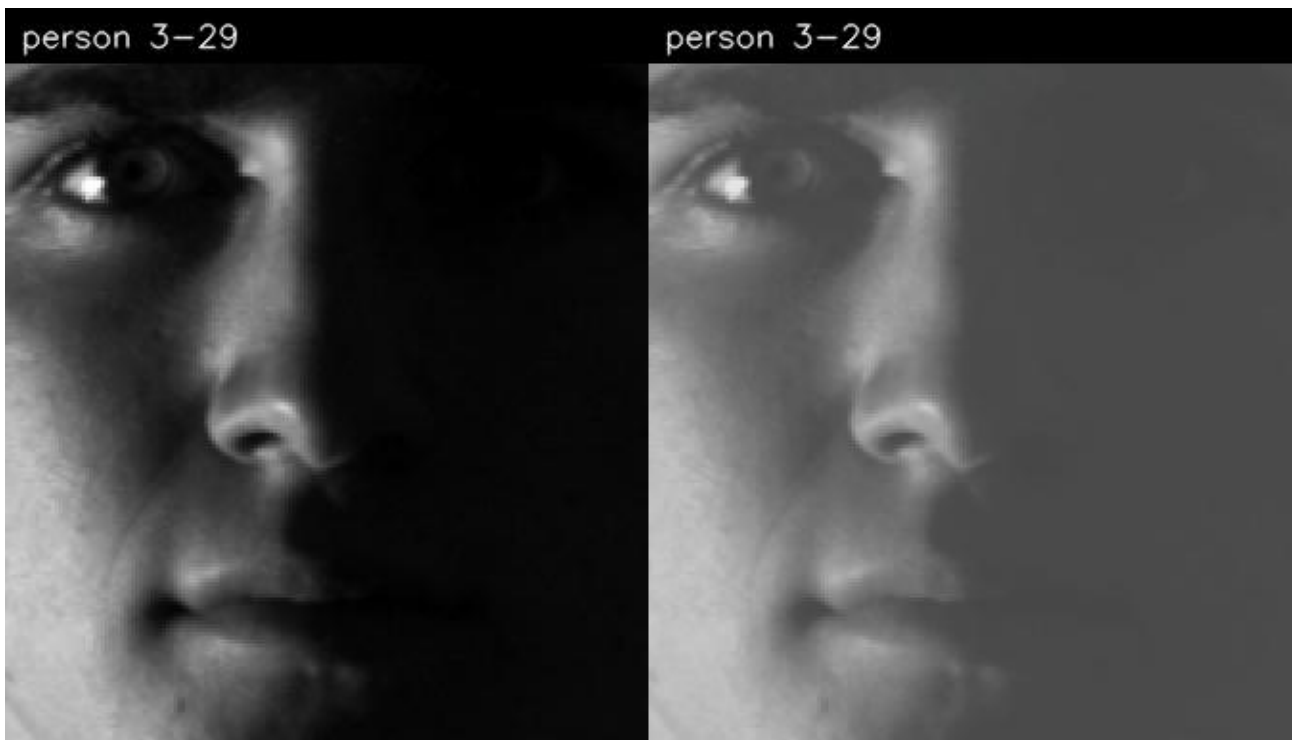


Рисунок 3.43 – Результат нормалізації зображення 29 з класу 3 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.44 – Результат нормалізації зображення 39 з класу 3 методом Color transfer (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.45 – Результат нормалізації зображення 11 з класу 3 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.46 – Результат нормалізації зображення 13 з класу 3 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.47 – Результат нормалізації зображення 16 з класу 3 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.48 – Результат нормалізації зображення 29 з класу 3 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)



Рисунок 3.49 – Результат нормалізації зображення 39 з класу 3 методом Color transfer на основі Face Parsing (вхідне (зліва) та нормалізоване (справа) зображення)

3.4 Порівняльний аналіз досліджених методів нормалізації обличчя на зображенні

#### 3.4.1 Аналіз метричних оцінок точності нормалізації методами

Після завершення нормалізації всіх зображень з усіх трьох класів проведено оцінку точності для кожного з методів.

Оцінка проводилась шляхом розрахунку наступних характеристик окремо на кожному класі:

- SSIM;
- Bhattacharyya;
- Gray-World angle;
- $\Delta E^*ab$ ;
- помилка контрасту  $L$ ;
- Clipping (black+white).

Характеристики розраховувались шляхом порівняння нормалізованого зображення до еталону відповідного класу.

Розраховані показники для першого класу наведені на рисунку 3.50.

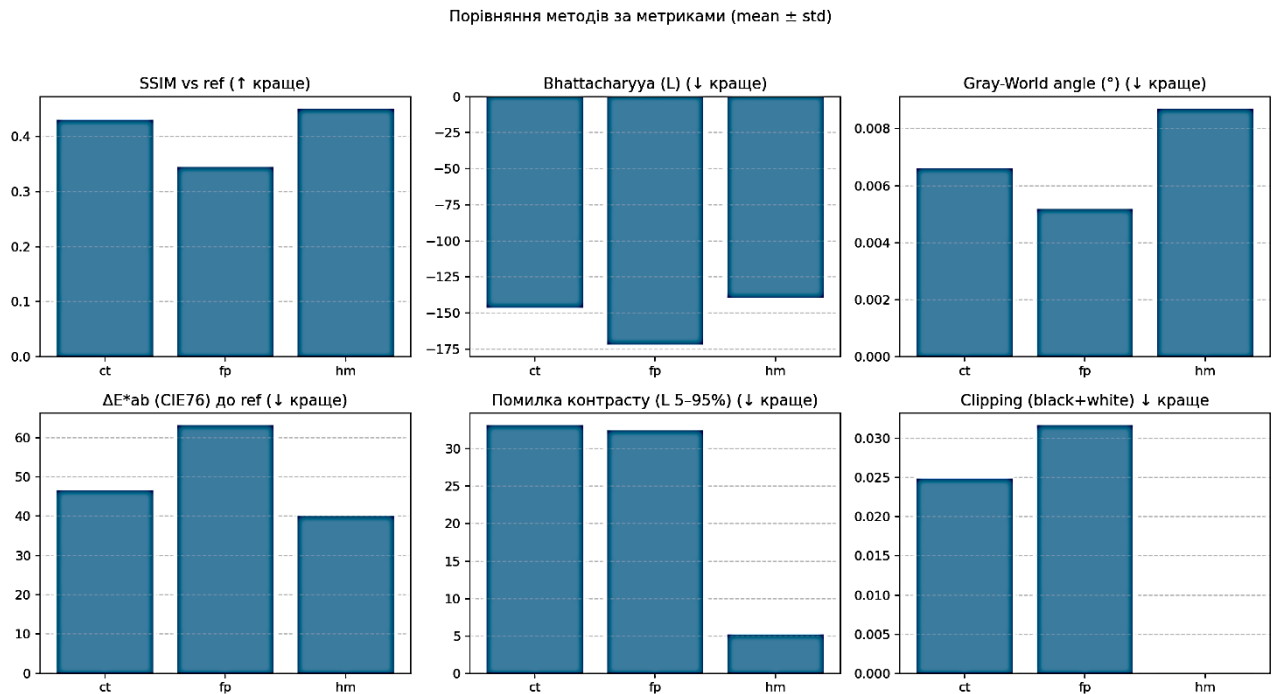


Рисунок 3.50 – Метричні оцінки точності нормалізації на зображеннях класу 1

Отримані дані показують, що за більшістю характеристик (окрім Bhattacharyya та Gray-World angle) найкраще на класі 1 себе продемонстрував метод Histogram matching.

За оцінкою SSIM близьким до результату Histogram matching був Color transfer. Перевага Color transfer на основі Face Parsing у показниках Bhattacharyya та Gray-World angle можна пояснити збереженням методом оригінальної гистограми, яка буде більш схожою до гистограми еталону.

Histogram matching, у свою чергу, підлаштовуючи гистограму під еталон, може змінювати її форму. У випадку Gray-World angle великий вплив має робота Color transfer на основі Face Parsing з кольорами, а не тільки з яскравістю. При нормалізації за допомогою Histogram matching зберігається колірний зсув, що впливає на показник.

Рисунок 3.51 демонструє метричні дані на класі 2.

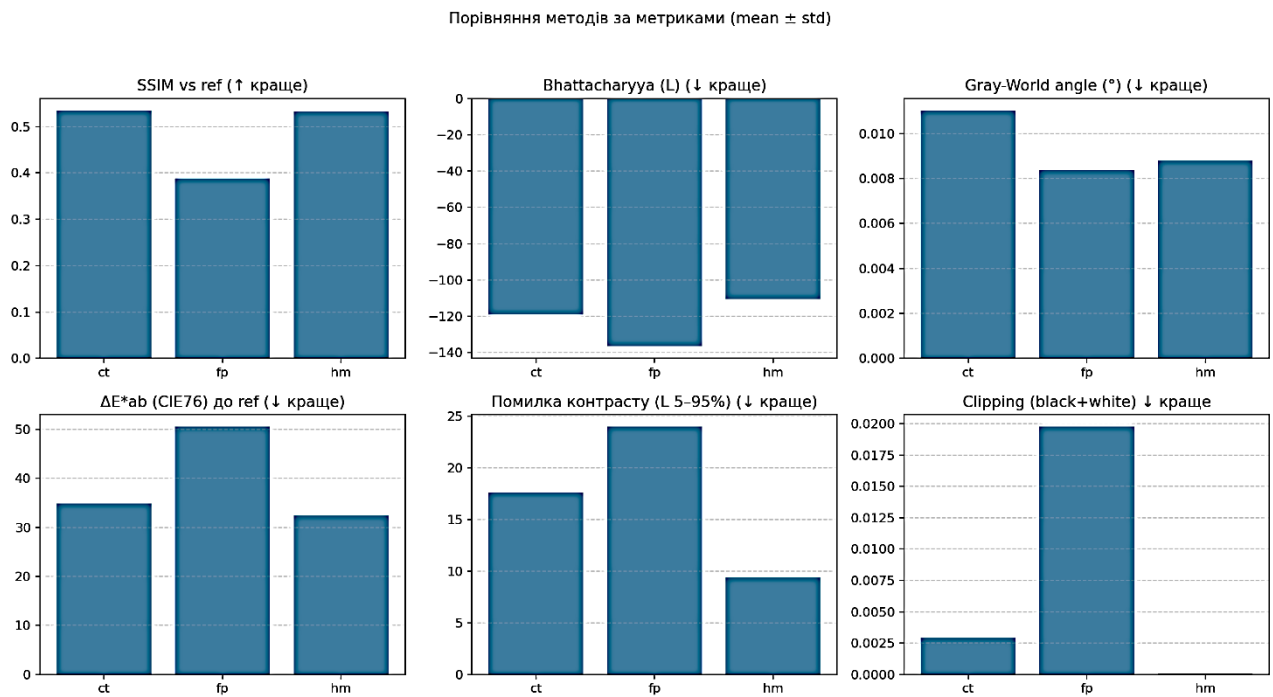


Рисунок 3.51 – Метричні оцінки точності нормалізації на зображеннях класу 2

Результати аналізу класу 2 демонструють схожість з показниками попереднього класу.

Відмінність нових результатів полягає у майже рівних показниках параметру SSIM для методів Color transfer та Histogram matching. Різниця між Color transfer на основі Face Parsing та Histogram matching на елементах класу 2 зменшилась. Це може бути пов'язано з відмінностями у природних кольорах шкіри людей, чії фото було розглянуто.

На відміну від зображень класу 1, збільшився показник Clipping для Histogram matching, що може свідчити про більшу втрату даних на наборі даних.

Результати аналізу зображень класу 3 наведені на рисунку 3.52.

Отримані показники показують збереження проблем Histogram matching, що вже розглянуто на попередніх класах.

Хоча на поточному класі зберігається мала різниця у показниках SSIM для методів Histogram matching та Color transfer, за показниками Bhattacharyya та Gray-World angle Histogram matching показав результат схожий до результату класу 1.

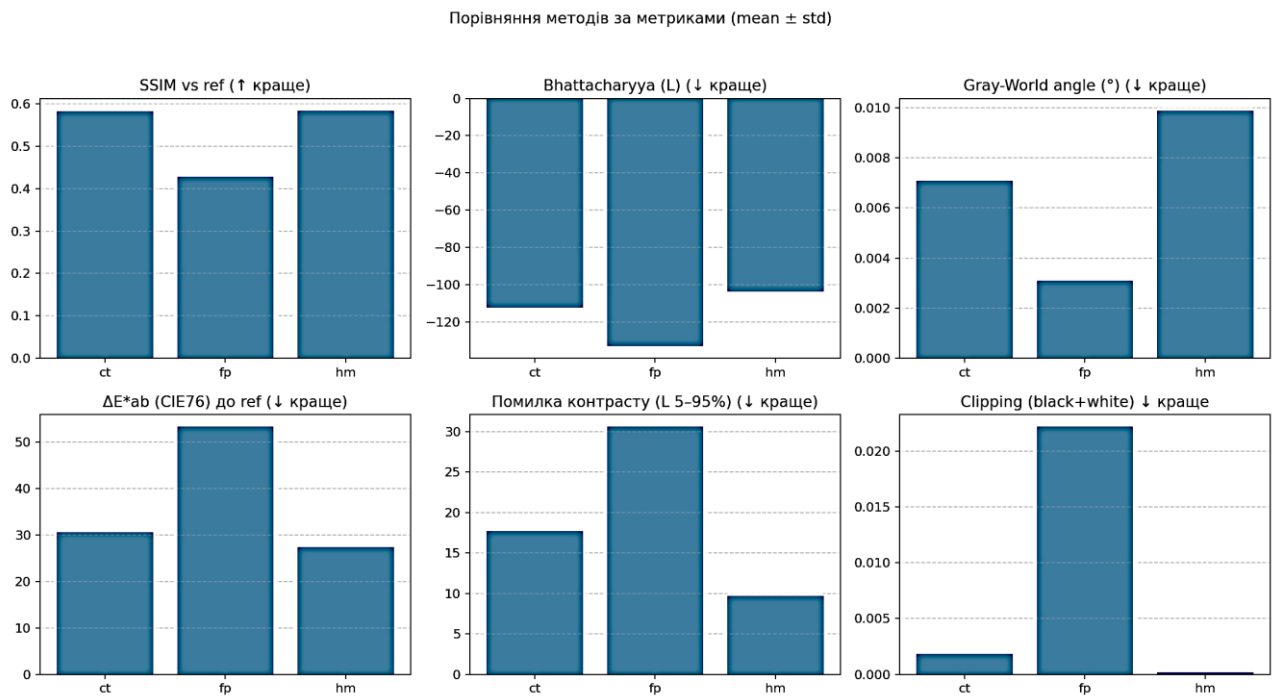


Рисунок 3.52 – Метричні оцінки точності нормалізації на зображеннях класу 2

### 3.4.2 Аналіз результатів порівняння нормалізованих зображень до еталону

У даному пункті розглянуто результати порівняння оригінальних та нормалізованих зображень відносно еталону за допомогою методів розглянутих у роботі [15].

Оригінальні зображення наведено у лівій частині на рисунках 3.5 – 3.49.

Кожне зображення порівнювалось лише з еталоном свого класу, що наведені на рисунку 3.3.

У таблиці 3.1 наведено результати порівняння вихідних зображень до еталонів.

Отримані дані показують, що зображення з відносно високим рівнем освітлення, наприклад 13, 16 та 39, мають високий ступінь схожості, навіть без додаткової нормалізації. При цьому зображення з низьким рівнем освітлення показують високі результати тільки методом FaceNet, що може свідчити про необхідність його подальшого покращення.

Таблиця 3.1 – Результати порівняння оригінальних зображень до еталонів

Назва класу	Номер зображення	Відсоток схожості зображень (%)			Середня схожість (%)
		Eigenfaces	FaceNet	OpenFace	
Person 1	11	34,65	85,55	83,64	67,95
	13	86,91	97,18	96,55	93,55
	16	76,74	89,52	92,32	86,19
	29	95,15	99,34	98,93	97,81
	39	69,56	94,43	90,13	84,71
Person 2	11	1,32	77,81	80,4	53,18
	13	96,59	98,5	96,97	97,35
	16	77,9	99,16	97,29	91,45
	29	28,15	85,53	85,71	66,46
	39	90,5	98,6	99,3	96,13
Person 3	11	25,5	91,71	80,39	65,87
	13	93,9	99,08	98,33	97,1
	16	78,03	98,7	95,66	90,8
	29	40,36	88,03	79,5	69,3
	39	95,6	96,97	94,68	95,75

У таблиці 3.2 наведено результати порівняння зображень, нормалізованих методом Histogram matching.

Результати порівняння нормалізованих зображень показують ріст ступеня схожості. Особливо вирости показники зображень, які не мали високого рівня освітлення. Після нормалізації, вони можуть бути ідентифіковані з високою точністю (>90%) і тільки в двох випадках схожість зображень є нижчою за цей поріг.

Таблиця 3.2 – Результати порівняння зображень нормалізованих методом Histogram matching до еталонів

Назва класу	Номер зображення	Відсоток схожості зображень (%)			Середня схожість (%)
		Eigenfaces	FaceNet	OpenFace	
Person 1	11	93,9	94,91	94	94,27
	13	95,96	99,29	97,9	97,72
	16	92,85	96,59	95,53	94,99
	29	97,6	99,56	98,37	98,51
	39	91,74	97,12	96,07	94,98
Person 2	11	97,31	98,16	97,64	97,7
	13	96,65	97,76	97,97	97,46
	16	79,58	88,05	88,63	85,42
	29	89,96	98,78	99,24	95,99
	39	90,43	96,45	89,46	92,11
Person 3	11	95,63	99,09	95,61	96,78
	13	93,65	97,62	94,74	95,34
	16	80,13	94,69	83,33	86,05
	29	96,33	97,67	94,94	96,31
	39	93,9	94,91	94	94,27

У таблиці 3.3 наведено результати порівняння зображень, нормалізованих методом Color transfer.

Зростання ступеню схожості після нормалізації є, але воно нижче за показники, що отримані методом Histogram matching.

Зафіксоване стрімке зростання відсотків схожості для зображень з низьким рівнем освітлення, зокрема, для зображення 11 у класах 1 та 3 (на приблизно 19% в обох випадках).

Таблиця 3.3 – Результати порівняння зображень нормалізованих методом Color transfer до еталонів

Назва класу	Номер зображення	Відсоток схожості зображень (%)			Середня схожість (%)
		Eigenfaces	FaceNet	OpenFace	
Person 1	11	88,71	86,04	88,20	87,65
	13	94,61	98,61	96,58	96,60
	16	90,52	92,55	92,47	91,85
	29	97,56	99,55	98,71	98,61
	39	88,51	95,19	88,76	90,82
Person 2	11	27,03	78,61	80,30	61,98
	13	97,36	98,63	97,45	97,81
	16	96,74	98,93	98,59	98,09
	29	76,86	85,85	85,21	82,64
	39	90,43	98,55	99,22	96,07
Person 3	11	88,81	89,49	81,90	86,73
	13	95,67	98,94	97,09	97,23
	16	93,87	98,91	94,91	95,90
	29	78,17	85,82	82,38	82,12
	39	96,43	97,67	93,38	95,83

Для зображення 11 у класі 2 ступінь також виріс, але все ще знаходиться у межах низького рівня схожості.

Середній ступінь схожості (90,66%) після нормалізації методом Color transfer можна вважати високим, але він значно поступається методу Histogram matching з середнім рівнем схожості (94,5%).

У таблиці 3.4 наведено результати порівняння після нормалізації методом Color transfer на основі Face Parsing.

Таблиця 3.4 – Результати порівняння зображень нормалізованих методом Color transfer на основі Face Parsing до еталонів

Назва класу	Номер зображення	Відсоток схожості зображень (%)			Середня схожість (%)
		Eigenfaces	FaceNet	OpenFace	
Person 1	11	89,22	86,60	83,09	86,30
	13	95,27	96,21	94,16	95,21
	16	91,74	89,51	85,84	89,03
	29	97,53	99,49	98,22	98,41
	39	89,50	93,36	88,53	90,46
Person 2	11	1,32	77,81	80,40	53,18
	13	96,60	98,49	96,98	97,36
	16	78,05	99,14	97,29	91,49
	29	28,38	85,45	85,71	66,51
	39	90,53	98,60	99,30	96,14
Person 3	11	25,50	91,71	80,39	65,87
	13	92,95	94,84	85,44	91,08
	16	78,03	98,70	95,66	90,80
	29	40,36	88,03	79,50	69,30
	39	93,81	94,31	84,78	90,97

З отриманих показників можна побачити необхідність подальшої роботи над вдосконаленням методу Color transfer на основі Face Parsing, оскільки нормалізація змогла істотно підвищити (на декілька %) ступінь схожості лише для трьох зображень (13, 16 та 39 для класу 1).

У решті випадків рівень схожості залишився тим самим або показав незначне зростання (до 0,4%). У випадку зображень 13 та 39 у класі 3, то відсоток схожості істотно знизився (на 5%–7%).

Середній рівень схожості для метода Color transfer на основі Face Parsing також залишається значно нижчим на відміну від попередніх методів (84,8%) і не сильно відрізняється від початкового (83,57%). Причиною таких низьких показників методу може бути погане взаємне налаштування моделі розпізнавання обличчя та методу нормалізації кольору.

Таким чином, проаналізувавши результати оцінки точності нормалізації вибраними методами, можна зробити висновок, що для проблеми нормалізації кольору зображення з подальшою оцінкою схожості з еталоном найкраще підходить метод Histogram matching навіть за умов невисоких показників при порівнянні за допомогою Bhattacharyya та Gray-World angle.

### 3.5 Аналіз спотворень на нормалізованих зображеннях

Приклади зображень отримані після нормалізації, що представлені на рисунках 3.5 – 3.49, демонструють основні види спотворень, які можливі під час використання розглянутих методів.

Наприклад, застосування методу Histogram matching може призводити до пікселізації частин зображення, особливо на темних зображеннях, наприклад, на рисунках 3.5 та 3.7. Такі спотворення зазвичай з'являються через використання CDF-функції, яка може стискати дані на певній кількості бітів або LUT-таблиці, яка є дуже чутливою до особливо темних чи світлих пікселів.

На зображеннях, що нормалізувались за допомогою Color transfer з'являється спотворення «Сіра вуаль», наприклад, на рисунку 3.11. Така особливість може з'являтися через великий спектр причин. Серед них, використання глобальних характеристик, через що, при занадто сильному вирівнюванню, зображення може стати більш сірим, оскільки усі кольори будуть підлаштовуватись під нейтраль. Також причиною може бути одночасна зміна усіх каналів  $\{l,a,b\}$  без фіксації нормалі.

Прикладом спотворення є поява меж між сегментами зображення, наприклад, на рисунку 3.49. Такий ефект з'являється через неузгодженість окремих сегментів між собою і може потребувати додаткового оброблення.

### 3.6 Перспективи подальших досліджень

Розроблений мікросервісний застосунок має значний потенціал для подальшого розвитку систем нормалізації кольору обличчя відносно еталону. При цьому існує низка аспектів, які відкривають можливість вдосконалення та перспективи подальшого дослідження даного питання.

Серед таких аспектів варто виділити:

- підвищення точності нормалізації: шляхом тонкого налаштування методів розглянути методів нормалізації кольору обличчя, привести оцінку метриками до одного рівня, який буде достатнім для отримання високого рівня схожості (>90%) усіма методами за допомогою розробленого застосунку [15];
- покращення методів нормалізації кольору обличчя: за допомогою додаткового тестування та налаштування методів добитися відсутності спотворення на 95% тестового набору даних;
- додавання нових методів нормалізації: впровадження нових методів нормалізації кольору обличчя дасть змогу порівнювати різні підходи до вирішення проблеми і отримувати більш точний результат;
- впровадження отримання найкращого результату: створення функціоналу, який дозволить швидко отримати найкращий результат нормалізації різними методами. Для цього необхідне покращення точності методів нормалізації та методів оцінки результатів;
- удосконалення системи оцінки результатів нормалізації: впровадження нових метрик для оцінки схожості нормалізованого зображення та еталону, покращення точності існуючих метрик;

– впровадження шифрування даних: для додаткового захисту результатів та вхідних зображень, необхідно впровадити додаткове шифрування усіх даних при обміні між мікросервісами.

Робота над розглянутими перспективами сприятиме розширенню та поліпшенню програми, збільшенню попиту серед розробників та власників мікросервісних застосунків.

## ВИСНОВКИ

У рамках кваліфікаційної роботи проведено дослідження методів нормалізації кольору обличчя відносно еталону шляхом розроблення мікросервісного десктоп-застосунку, що надає можливість отримати нормалізоване зображення, використовуючи вхідне аналізоване зображення та еталон.

Проведено аналіз літературних джерел щодо застосування різних способів нормалізації кольору на зображенні, що дало можливість оцінити поточний стан розвитку проблематики [18–36].

Проведено аналіз сучасних методів нормалізації кольору відносно еталону, що дало можливість виокремити та оцінити перспективні методи для їхнього подальшого детального розгляду.

У рамках кваліфікаційної роботи досліджено методи Histogram matching, Color transfer, Color transfer на основі Face Parsing, що дало змогу для кожного описати покроковий алгоритм роботи.

Візуалізовано алгоритми роботи методів шляхом побудови блок-схем, що дало змогу зобразити кожен крок та створити програмну реалізацію для тестування на наборі даних. Встановлено, що для задачі нормалізації обличчя при подальшому порівнянні з іншим зображенням кращим є метод Histogram matching.

На основі набору даних Extended Yale B [16] сформовано власний тестовий набір даних з 3 класів (зображення 3 людей) по 56 елементів у кожному, де елементом є фото людини при різному освітленні. Сформований набір дозволив оцінити роботу методів у за різних умов світла і при різних особливостях обличчя.

Розроблено можливість тестування за допомогою звернення до API застосунку, що дозволив перевірити систему і отримати результат без необхідності запуску інших мікросервісів.

Науковою новизною є виявлення можливостей використання методики вирівнювання кольорових розподілів за допомогою гістограм, а також перенесення *Lab*-простору за допомогою розподілу обличчя на сегменти для подальшої ідентифікації обличчя та надання доступу до системи.

Практична значущість дослідження полягає у:

- формуванні підходу до попередньої нормалізації біометричних даних;
- підвищенні надійності та точності ідентифікації за різного освітлення;
- дослідженні використання методів у застосунках комп'ютерного зору для нормалізації зображень.

Результати дослідження апробовано у вигляді 2 тез доповідей під час «III Міжнародної науково-практичної конференції «Research on the development of science and the implementation of technology» [37] та VIII Міжнародної науково-практичної конференції «Academic research by scientists in the field of modern technologies» [38], а також у вигляді статті у зарубіжному науковому журналі «International Journal of Academic Information Systems Research» [39].

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. What is histogram matching? URL: <https://www.educative.io/answers/what-is-histogram-matching> (дата звернення 14.10.2025).
2. Histogram matching. URL: [https://scikit-image.org/docs/0.25.x/auto\\_examples/color\\_exposure/plot\\_histogram\\_matching.html](https://scikit-image.org/docs/0.25.x/auto_examples/color_exposure/plot_histogram_matching.html) (дата звернення 14.10.2025).
3. Wikipedia contributors. Image color transfer. URL: [https://en.wikipedia.org/w/index.php?title=Image\\_color\\_transfer&oldid=1308681023](https://en.wikipedia.org/w/index.php?title=Image_color_transfer&oldid=1308681023) (дата звернення 14.10.2025).
4. Shi, M., Zhang, J. Q., Chen, S. Y., Gao, L., Lai, Y. K., & Zhang, F. L. (2022). Reference-based deep line art video colorization. *IEEE Transactions on Visualization and Computer Graphics*, 29(6), 2965-2979.
5. Li, Z., Geng, Z., Kang, Z., Chen, W., & Yang, Y. (2022, October). Eliminating gradient conflict in reference-based line-art colorization. In *European Conference on Computer Vision* (pp. 579-596). Cham: Springer Nature Switzerland.
6. Liu, X., Wu, W., Li, C., Li, Y., & Wu, H. (2022). Reference-guided structure-aware deep sketch colorization for cartoons. *Computational Visual Media*, 8(1), 135-148.
7. Yang, J., Liu, C., & Zhang, L. (2010). Color space normalization: Enhancing the discriminating power of color spaces for face recognition. *Pattern Recognition*, 43(4), 1454-1466.
8. Al-Osaimi, F. R., Bennamoun, M., & Mian, A. (2006, November). Illumination normalization for color face images. In *International symposium on visual computing* (pp. 90-101). Berlin, Heidelberg. Springer Berlin Heidelberg.
9. Goceri, E. (2021). Deep learning-based classification of facial dermatological disorders. *Computers in Biology and Medicine*, 128, 104118.

10. Kumar, A., Kumar, M., & Kaur, A. (2021). Face detection in still images under occlusion and non-uniform illumination. *Multimedia Tools and Applications*, 80(10), 14565-14590.

11. Li, B., Zhu, Y., Wang, Y., Lin, C. W., Ghanem, B., & Shen, L. (2021). Anigan: Style-guided generative adversarial networks for unsupervised anime face generation. *IEEE Transactions on Multimedia*, 24, 4077-4091.

12. Wikipedia contributors. Histogram matching. URL: [https://en.wikipedia.org/w/index.php?title=Histogram\\_matching&oldid=1316719125](https://en.wikipedia.org/w/index.php?title=Histogram_matching&oldid=1316719125) (дата звернення 14.10.2025).

13. Histogram Matching. How to generate a histogram for an image, how to equalize the histogram, and finally how to modify your image histogram to be similar to another histogram. URL: <https://medium.com/data-science/histogram-matching-ee3a67b4cbc1> (дата звернення 14.10.2025).

14. Media Pipe Face Mesh. URL: [https://mediapipe.readthedocs.io/en/latest/solutions/face\\_mesh.html](https://mediapipe.readthedocs.io/en/latest/solutions/face_mesh.html) (дата звернення 05.11.2025).

15. Ларін І. П. Розроблення вебзастосунку розпізнавання та ідентифікації користувачів банківської системи : пояснювальна записка до кваліфікаційної роботи здобувача вищої освіти на першому (бакалаврському) рівні, спеціальність 122 Комп'ютерні науки / І. П. Ларін ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. Харків, 2024. 88 с.

16. ExtYaleBCroppedPNG. The Extended Yale Face Database B Cropped. URL: <https://www.kaggle.com/datasets/tbourton/extyalebcroppedpng> (дата звернення 11.11.2025).

17. Postman. URL: <https://www.postman.com/> (дата звернення 05.11.2025).

18. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Tools for fast metric data search in structural methods for image classification, *IEEE Access*, 10, pp. 124738-124746.

19. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

20. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, *Сучасні інформаційні системи*, 6(3), С. 5-12.

21. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.

22. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.

23. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.

24. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, *International Journal of Academic and Applied Research*, 7(9), pp. 57-70.

25. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.

26. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 113-125.

27. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S»*. *Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine*, pp. 25-27.
28. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, vol. 11, pp. 126938-126949.
29. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, *International Journal of Academic and Applied Research*, 7(11), pp. 134-145.
30. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2024) Improving the effectiveness of image classification structural methods by compressing the description according to the information content criterion, *Computers, Materials & Continua*, vol. 80, no. 2, pp. 3085-3106.
31. Gorokhovatskyi V., Chmutov Y., Tvoroshenko I., and Kobylin O. (2025) Reducing computational costs by compressing the structural description in image classification methods, *Advanced Information Systems*, vol. 9, no. 1, pp. 5–12.
32. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2025) Image description compression in classification structural methods, *IEEE Access*, vol. 13, pp. 43631-43641.
33. Yakovleva O., Matúšová S., Tvoroshenko I., and Isaiev Y. (2024) Visitor counting based on video stream analysis from surveillance cameras to solve various business problems, *Verejná správa a regionálny rozvoj ekonómia, manažment a marketing*, XX(1), pp. 67-87.
34. Bohdan N., Tvoroshenko I., Gorokhovatskyi V., and Kobylin O. (2025) Development of a hybrid method to enhance context memory for a chatbot application based on large language models, *International Journal of Academic Information Systems Research*, 9(10), pp. 7-18.

35. Гороховатський В.О., Творошенко І.С. (2025) Оцінювання значущості ознак для підвищення продуктивності структурних методів класифікації зображень. *Проблеми інформатики та моделювання (ПІМ-2025). Тези двадцять п'ятої міжнародної науково-технічної конференції (25 – 28 вересня 2025 року)*. Харків: НТУ «ХПІ», С. 38-43.

36. Suprun A., Tvoroshenko I., Gorokhovatskyi V., and Yakovleva O. (2025) Development and research of a method for the combined use of large language models for text generation, *International Journal of Academic and Applied Research*, 9(10), pp. 249-263.

37. Ларін І.П. (2025) Особливості застосування методу Histogram Matching для нормалізації кольору обличчя, *Abstracts of III International Scientific and Practical Conference «Research on the development of science and the implementation of technology»*, (September 16 – 19, 2025). Hamburg, Germany, pp. 53-55.

38. Ларін І.П. (2025) Деякі аспекти застосування Color space transfer на основі Face Parsing у задачах обробки зображень обличчя, *Abstracts of VIII International Scientific and Practical Conference «Academic research by scientists in the field of modern technologies»*, (October 21 – 24, 2025). Milan, Italy, pp. 33-37.

39. Larin I., Tvoroshenko I., Gorokhovatskyi V., and Liubchenko V. (2025) Research and comparison of facial color normalization methods relative to an etalon image, *International Journal of Academic and Applied Research*, 9(11), pp. 36-47.