

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
« ____ » _____ 20__ р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА
ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Інформаційна система розпізнавання людського обличчя

Освітній ступінь «бакалавр»
Спеціальність 122 – «комп'ютерні науки»

Керівник проекту:

(підпис)

Барбарук В.М.

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Критська Я.О.

(ініціали, прізвище)

Здобувач вищої освіти:

(підпис)

Байбус Р.І.

(ініціали, прізвище)

Група:

КН-15з

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітній ступінь бакалавр
Напрямок підготовки _____
(шифр і назва)
Спеціальність 122 – комп'ютерні науки
(шифр і назва)

ЗАТВЕРДЖУЮ:

Завідувач кафедри КНІ
_____ І.С. Скарга-Бандурова
« _____ » _____ 20 ____ р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Байбусу Руслану Івановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна система розпізнавання людського обличчя

керівник проекту (роботи) Барбарук Віктор Миколайович, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "13" 05 2019 р. № 83/15.15

2. Термін подання студентом роботи 16.06.2019

3. Вихідні дані до роботи Математичні моделі знаходження обличчя на зображенні, теоретичні відомості про методи пошуку та класифікації даних, їх відображення

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Огляд існуючих систем та методів, теоретичні відомості, проектування та реалізація програмного додатку, охорона праці

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст. викл. Критська Я.О.		

7. Дата видачі завдання 30.04.2019

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз завдання та робота з літературою	05.05.2019 - 13.05.2019	
2	Аналіз технічних засобів	14.05.2019 - 22.05.2019	
3	Розробка алгоритму	22.05.2019 - 02.06.2019	
4	Програмна реалізація	02.06 .2019- 11.06.2019	
5	Оформлення пояснювальної записки	11.06.2019 - 14.06.2019	
6	Підготовка презентації та доповіді	14.06.2019 - 16.06.2019	

Здобувач вищої освіти

_____ (підпис)

Байбус Р.І.

_____ (прізвище та ініціали)

Керівник

_____ (підпис)

Барбарук В.М.

_____ (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 68с., 24 рис., 20 джерел.

У роботі розглядається проблема пошуку облич на серії знімків, а також перевірки належності цих облич одній і тій самій людині. Під час роботи над атестаційною роботою розглядались різноманітні засоби знаходження обличчя на зображенні та ідентифікації зображеної на ній особистості. Для виконання поточної роботи було обрано метод Віоли-Джонса в сукупності із методом скануючого вікна для пошуку облич на зображеннях. Ці ж методи використовуються для пошуку частин обличчя та перевірки того, що обличчя на знімках з різних ракурсів належать одній і тій самій людині.

Розроблений програмний додаток реалізований через REST архітектуру. Складається з тонкого клієнту, що є віконним додатком, написаним на мові С# із використанням WindowsForms, та серверного додатку, написаного на мові Java із використанням Spring Boot, Maven та бібліотеки комп'ютерного зору OpenCV.

Ключові слова: МЕТОД ВІОЛИ-ДЖОНСА, МЕТОД СКАНУЮЧОГО ВІКНА, КАСКАДИ ХААРА, КОМП'ЮТЕРНИЙ ЗІР, РОЗПІЗНАВАННЯ ОБЛИЧЧЯ, РОЗПІЗНАВАННЯ ОБРАЗІВ, С#, JAVA, OPENCV, REST, SPRING BOOT.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А., м. Сєверодонецьк, 93400с.

ЗМІСТ

ВСТУП	7
1 ОГЛЯД ІСНУЮЧИХ СИСТЕМ ТА МЕТОДІВ	9
1.1 Огляд сучасних систем розпізнавання облич	9
1.2 Огляд деяких методів розпізнавання образів.....	11
1.2.1 Класифікатор Хаара.....	12
1.2.2 ЛБШ класифікатор.....	14
1.2.3 Алгоритм SIFT	15
1.2.4 Алгоритм SURF.....	16
1.3 Вибір методів розробки	18
1.3.1 Вибір методів для пошуку облич на зображенні.....	19
1.3.2 Вибір методу розпізнавання обличчя з різних ракурсів	20
1.4 Постановка задачі	21
2 ТЕОРЕТИЧНІ ВІДОМОСТІ.....	23
2.1 Алгоритми, використані при пошуку обличчя	23
2.1.1 Метод скануючого вікна	23
2.1.2 Метод Віюли-Джонса	25
2.1.3 Ознаки Хаара.....	26
2.2 Визначення факту належності облич одній і тій самій людині	31
3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ДОДАТКУ	35
3.1 Вибір технологій розробки	35
3.1.1. Бібліотека OpenCV.....	35
3.1.2 Мова програмування Java	36
3.1.3 Spring Boot	36
3.1.4 С# та WindowsForms.....	38
3.2 Архітектура програмного засобу.....	39
3.2.1 Архітектура REST.....	39
3.2.2 REST архітектура у поточному додатку.....	40
3.3 Дизайн клієнтської частини	43

3.4 Інструкція користувача.....	45
3.4.1 Інструкція до встановлення та запуску серверної частини додатку	45
3.4.2 Інструкція щодо використання клієнтської частини додатку.	45
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.	47
4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектованого об'єкту, що мають вплив на персонал	47
4.2 Заходи щодо техніки безпеки	48
4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці.....	51
4.4 Рекомендації по пожежній безпеці	55
ВИСНОВКИ.....	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	62
ДОДАТОК А. Електронні плакати.....	64

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

XML – Extensible Markup Language

ЛБШ – локальні бінарні шаблони

SURF - speeded up robust features

SIFT - scale-invariant feature transform

ПЗ – програмне забезпечення

REST – Representational state transfer

HTTP – Hyper Text Transfer Protocol

JSON – JavaScript Object Notation

HTML – HyperText Markup Language

ВСТУП

Дедалі популярнішими стають системи комп'ютерного зору. У сучасному світі це забезпечує додаткові можливості користувачам найрізноманітніших програмних засобів сучасних пристроїв. Комп'ютерний зір наразі використовується у багатьох областях. Найбанальнішим прикладом може бути розпізнавання номерних знаків на машинах, що відкриває можливості як для попередження дорожньо-транспортних подій (бо водії їздитимуть значно обережніше, якщо знатимуть, що за ними стежать камери відеоспостереження), так і для знаходження винуватців дорожньо-транспортної пригоди у випадку, якщо вони втекли з місця події. Можливості комп'ютерного зору настільки великі, що їх можна використовувати навіть у більш-менш особистих цілях. Наприклад, інженер компанії Nvidia, втомився від постійних візитів котів на двір свого будинку, що спроектував систему, що, аналізувала відеопотік з камери відеоспостереження та, якщо знаходила на зображенні котів, вмикала розприскувачі води, лякаючи тварин та змушуючи їх покинути зону дискомфорту[1].

Однією з найперспективніших галузей розпізнавання образів є розпізнавання обличь. Воно використовується і у додатках для камер смартфонів, і у деяких соціальних мережах – цю технологію навіть почали використовувати замість паролю. А розпізнавання облич та визначення особистості цього обличчя у режимі реального часу наразі вже використовують держави, встановлюючи велику кількість камер відеоспостереження на вулицях великих міст для пришвидшення затримання злочинців, а іноді навіть виявлення особистості, що здійснила правопорушення, якщо воно було записане на камеру.

У рамках поточної атестаційної роботи розглядатиметься задача знаходження облич на фотографіях людини з різних ракурсів та підтвердження того, що на серії цих фотографій одна й та ж людина. Існує велика кількість алгоритмів, що дозволяють вирішувати завдання

розпізнавання образів, але вирішення задач поточної атестаційної роботи доцільніше усього проводити із використанням методу Віоли-Джонса, інакше відомого як каскади Хаара, оскільки цей метод достатньо швидкий та достатньо точний, за умови гарно натренованого каскаду. В роботі використовуватиметься каскад, що приймає рішення щодо знаходження обличчя у області пошуку, потрібно цю область пошуку виділяти. Оскільки заздалегідь невідомо у якій частині зображення буде знаходитись обличчя та якого воно буде розміру відносно розміру зображення, треба використовувати алгоритм, який змінюватиме положення та розмір вікна пошуку, тобто принцип скануючого вікна.

У подальшому планується розвинути функціонал цієї роботи, додавши можливість зробити 3D-модель обличчя людини, фотографії якої оброблялись. Пізніше, використовуючи цю 3D-модель, можна побудувати, наприклад, модель персонажа у комп'ютерній грі.

1 ОГЛЯД ІСНУЮЧИХ СИСТЕМ ТА МЕТОДІВ

1.1 Огляд сучасних систем розпізнавання облич

Наразі дедалі популярнішими стають різноманітні системи розпізнавання образів. Особливо популярними вони стають на хвилі розвитку штучного інтелекту та його використанні. Найпростішим прикладом є використання таких систем у безпілотних автомобілях, адже завдяки цьому машини можуть розпізнавати різноманітні перепони на дорозі та уникати зіткнення із ними, проте це не єдина галузь, де доцільно використовувати розпізнавання образів.

Зокрема, надзвичайної популярності набуло розпізнавання людських обличчя. Зараз воно використовується скрізь. Наприклад, без розпізнавання обличчя не обійдеться жодний додаток до камери сучасного смартфона (рис. 1.1), адже це дозволило ввести функцію автоматичного налаштування фокусу камери на обличчі. Розпізнавання обличчя використовується й у більш розважальних додатках, зокрема важко не згадати й такі, що дозволяють накладати так звані «маски» на обличчя користувача у режимі поточного часу. Проте у цих додатках зазвичай має місце лише розпізнавання того місця на зображенні, де, власне кажучи, знаходиться обличчя. А це лише одна з проблем, що розглядаються у цій роботі.



Рисунок 1.1 – Розпізнавання обличчя у камері сучасного смартфона

Десь поряд із знаходженням обличчя на зображенні завжди існує і проблема виявлення власника цього обличчя – і це друга проблема, що розглядається у поточній роботі. Вона також є надзвичайно актуальною, але вже у менш розважальному контексті – зазвичай її використовують із метою захисту. Зокрема, наприкінці 2017 року компанія Apple презентувала технологію FaceID, що дозволяє використовувати власне обличчя у якості паролю для розблокування смартфона або здійснення покупок у додатках Apple та деяких інших, що підтримують таку функцію – достатньо просто поглянути у фронтальну камеру смартфона. Компанія Samsung не стала відставати від своїх конкурентів на теренах розпізнавання облич та розробила для смартфонів S9 та S9+ функцію під назвою «селфімодзі», що розпізнає обличчя людини та робить невелику 3D-модельку із цим обличчям, яка надалі використовується для спеціальних анімованих емодзі. А соціальна мережа Facebook ввела автоматичне відмічування людей на фотографіях: система шукає на завантаженій користувачем фотографії обличчя та намагається знайти власника обличчя аналізуючи фотографії з профілів друзів користувача та автоматично відмічаючи знайдених друзів.

Ця функція може бути надзвичайно корисною і при забезпеченні безпеки населення. Безліч країн світу вже мають гарно налагоджену мережу камер відеоспостереження, що розміщені ледве не на кожному кроці принаймні у крупних районах великих міст (рис. 1.2). Системи розпізнавання обличчя допомагають не лише знайти тих злочинців, які вже у розшуку, але й виявити особистість людини, що скоїла правопорушення у полі зору камери.

Цей тренд потрохи підхвачує й ігрова індустрія. Наприклад, Star Citizen аносувала функцію, що розпізнає міміку гравця через веб-камеру у режимі реального часу та одразу ж відтворює її на обличчі персонажу. Проте у ігровій індустрії досі немає таких продуктів, що дозволяли б створювати модель персонажу, що був би максимально схожим на гравця, уникаючи довгої метушні у редакторі створення персонажу та використовуючи замість цього розпізнавання обличчя гравця з фотографій або ж із зображення з веб-

камери. Багато людей створюють схожих на себе персонажів у іграх, тож автоматизація цієї процедури може бути корисною та вигідною.



Рисунок 1.2 – Розпізнавання образів у відеопотоці з камери відеоспостереження

Метою ж поточної роботи є розробка програми, що буде розпізнавати обличчя людини на серії знімків. Це перший крок до кінцевої мети – розробки додатку, що буде будувати 3D-модель обличчя людини та використовувати ці дані під час створення 3D-моделі персонажу. Такий додаток може стати гарним доповненням для будь-якої відеогри, де потребується створювати персонажа самотужки.

1.2 Огляд деяких методів розпізнавання образів

Однією із найважливіших частин розпізнавання образів є класифікація, тобто процес відрізнення зображення з певним класом шуканих об'єктів від зображення без таких. Комп'ютерна програма, яка це виконує, називається

класифікатором. Такі додатки часто використовують машинне навчання, аби, навчаючись на сотнях тисяч прикладів обличь, виявити ознаки, за якими можна буде коректно опізнати об'єкт новому зображенні [2]. У навчанні використовуються як позитивні зображення, на яких є об'єкти, що належать до певного класу, так і негативні, на яких таких об'єктів немає. Результатами навчання є файли у форматі XML, які містять ознаки, виявлені у навчальних даних. Їх можна знайти в мережі Інтернет у вигляді так званих «каскадів», вже готових до роботи. Певні розповсюджені каскади також часто йдуть у комплекті з бібліотеками, які їх використовують у своїх алгоритмах – наприклад, OpenCV постачається разом із каскадами, що дозволяють розпізнавати обличчя, певні їх частини, людей у повний зріст тощо.

Прикладами таких класифікаторів можуть служити класифікатор Хаара і ЛБШ класифікатор. Обидва приймають на вхід зображення у градаціях сірого. Фактично, їм не потрібен колір, щоб визначити, чи присутнє на зображенні обличчя, чи ні.

Окрім цих класифікаторів існують також алгоритми, засновані на виділенні ключових точок на зображенні – SIFT та SURF. Вони повільніші, аніж класифікатори Хаара та ЛБШ та часто застосовуються не лише для розпізнавання об'єктів, а й для автоматичного картографування та навігації, зшивання зображень, 3D-моделювання, розпізнавання жестів, відстеження відео, індивідуального оточення дикої природи та позиційного трекінгу.

1.2.1 Класифікатор Хаара

Класифікатор Хаара заснований на алгоритмі машинного навчання, який було створено Полом Віолою та Майклом Джонсом. Він полягає у тому, що на зображенні виділяються деякі ознаки, за якими у більшості випадків вдається виявити обличчя. Ці ознаки засновані на контрасті між певними частинами обличчя. Приклади таких ознак показані на рис. 1.3.

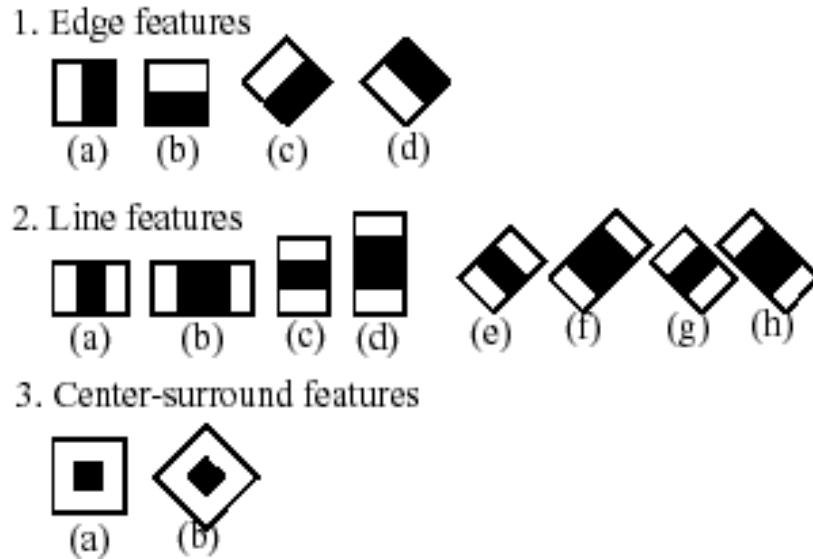


Рисунок 1.3 – Приклади ознак Хаара

Для виявлення ознак використовується простий алгоритм – обчислюється різниця між сумами пікселів у білій частині та у чорній. Але це виконується у всіх можливих частинах зображення для усіх можливих розмірів, що дає багато ознак.

На прикладі, який показано на рис. 1.4, зображено візуалізацію алгоритму: беруться 2 ознаки, перша з яких фокусується на тому факті, що область очей зазвичай темніша ніж область носа і щік; а другий – на тому, що очі, які розташовані по боках від носа, теж темніші за нього.

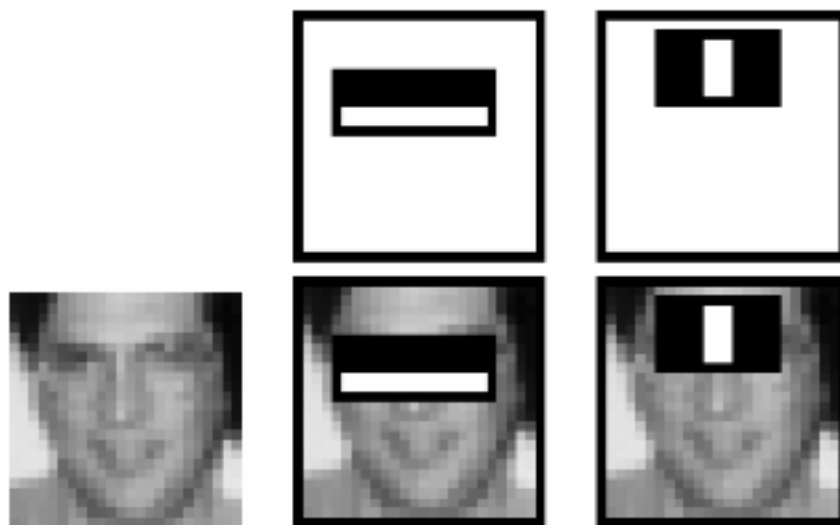


Рисунок 1.4 – Різні етапи візуалізації

Але велика кількість цих ознак виявляються не важливими. Наприклад, використовувати такі ознаки на щоках не має сенсу, бо різниці між яскравістю майже не буде. Тому алгоритм відразу відкидає незначні ознаки і залишає лише ті, які мають значення. Це відбувається завдяки іншому алгоритму машинного навчання, який використовується разом із методом Віоли-Джонса, під назвою AdaBoost. Він залишає лише ті ознаки, які підвищують ефективність класифікації.

До того ж, варто зазначити, що цей метод враховує той факт, що зазвичай на зображеннях більша частина не є обличчям. Він перевіряє наявність обличчя для того, щоб відкинути непотрібні частини зображення, аби не обробляти їх знову.

1.2.2 ЛБШ класифікатор

Інший класифікатор – ЛБШ, або локальні бінарні шаблони – також навчається на великій кількості зображень. Це класифікатор текстур, який виявляє мікроскопічні візуальні шаблони у зображенні. Він зберігає свої ознаки у векторі ознак.

Знаходяться ці ознаки наступним чином. Кожне навчальне зображення поділяється на блоки. Кожен блок розміром 3 на 3 пікселя – у сумі 9 пікселів. Найбільша увага надається центральному пікселю.

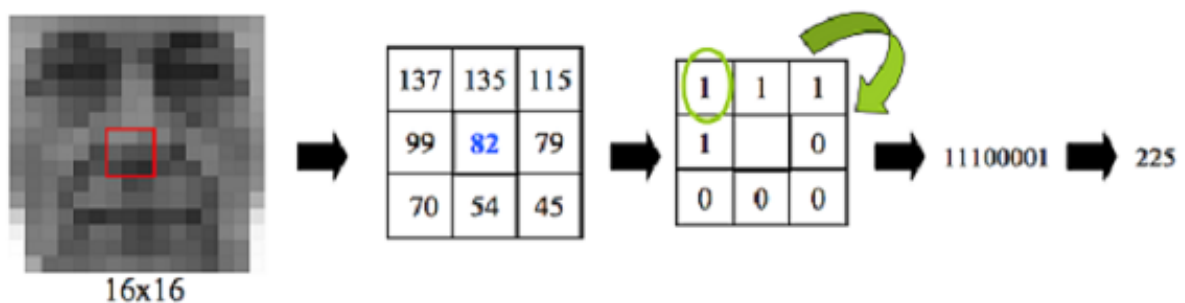


Рисунок 1.5 – Алгоритм ЛБШ

Для кожного блока обчислюється різниця між центральним та периферійними пікселями. Для кожного пікселя, який більше чи дорівнює центральному, встановлюється значення 1, для інших – 0. Нові значення читаються за годинниковою стрілкою, що створює двійковий код 8-значного числа. Це новостворене число встановлюється новим значенням центрального пікселя. Детально цей алгоритм показано на рис. 1.5.

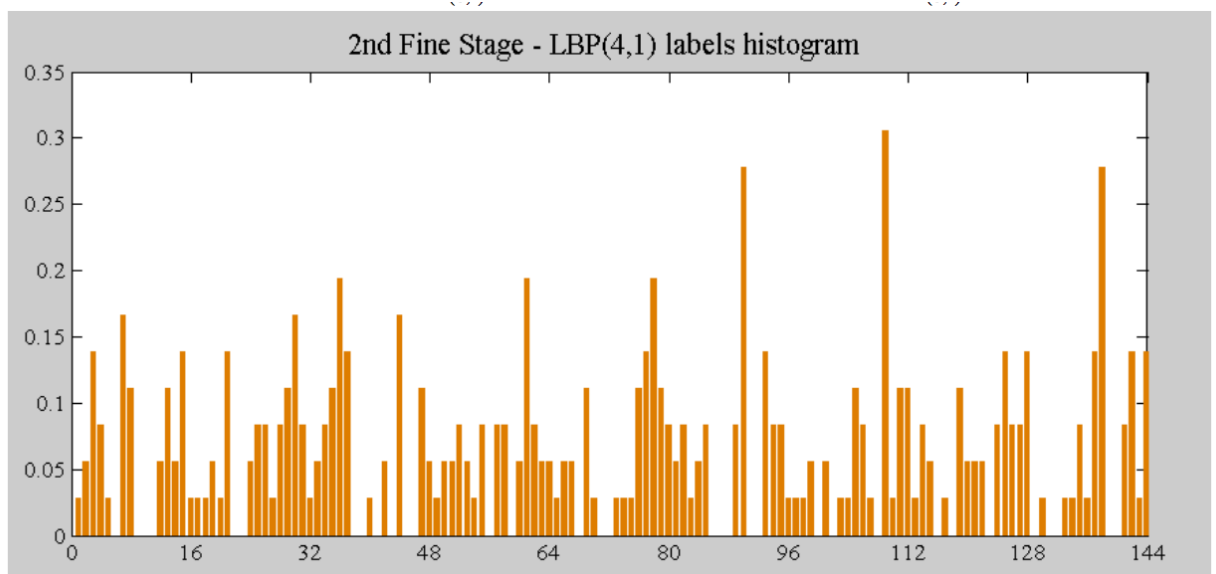


Рисунок 1.6 – Гістограма векторів ознак на базі зображення

Ці значення надалі формуються у гістограму, по одному стовпчику на кожен блок у зображенні. Збірні дані з декількох гістограм дозволяють створити той самий вектор ознак із усіма важливими ознаками (рис. 1.6).

1.2.3 Алгоритм SIFT

Масштабно-інваріантної функція перетворення (SIFT) є алгоритмом комп'ютерного зору для виявлення і опису локальних особливостей в зображеннях. Алгоритм був запатентований і опублікований Девідом Лоу в 1999 році [3]. Він дозволяє порівнювати зображення, що мають такі трансформації, як зміна масштабу, зміщення об'єкту на сцені, повороти

камери або об'єкту. Так само, як і попередні, алгоритм працює із зображеннями у градаціях сірого.

Спочатку знаходяться особливі ключові точки на зображенні. Після цього для кожної такої точки обчислюється один або декілька напрямів ключової точки, тобто кута з інтервалу $[0^\circ; 359^\circ)$. Для цього зазвичай беруть кілька сусідніх точок та підраховуються їх градієнти. Кінцевий напрям ключової точки знаходиться з гистограми напрямків ключових точок вікна – він буде у проміжку, що покривається максимальною компонентою гистограми.

Далі вираховується значення дескрипторів. Обирається кількість точок круг ключової, що будуть враховуватись (зазвичай 16 точок). Для кожної з них будується гистограма, що аналогічна гистограмі орієнтації ключової точки, але із значно меншим числом компонент, на основі цих даних формується вектор з 128 чисел (16 гистограм по 8 компонентів) – так званий дескриптор SIFT.

Для кожного зображення формуються сотні, а іноді й тисячі дескрипторів. Для їх зберігання та обробки використовуються спеціальні структури даних, такі, як K-мірне дерево, R*-дерево, SR-дерево.

Порівняння зображень проводиться шляхом порівняння відношення відстаней кожного із дескрипторів деякого зображення із двома найближчими сусідами. Якщо відношення відстаней менше встановленого порогового значення, дескриптори вважаються співпадаючими.

1.2.4 Алгоритм SURF

У області комп'ютерного зору прискорення надійних функцій (SURF) є запатентованим детектором та дескриптором локальних функцій. Він може дуже успішно використовуватись для таких завдань, як розпізнавання об'єктів, реєстрація зображень, класифікація або 3D-реконструкція. Частково це нагадує дескриптор перетворення значень інваріантних функцій (SIFT).

Стандартна версія SURF в кілька разів швидше, ніж SIFT, і її автори вимагають більшої надійності проти різних перетворень зображення, ніж SIFT. Його дескриптор опирається на суму вейвлет-відповіді Хаара в точці інтересу.

Дескриптори SURF використовувались для пошуку та розпізнавання об'єктів, людей або обличчя, для відтворення 3D-сцен, відстеження об'єктів та виявлення цікавих пам'яток.

SURF вперше була представлена Гербертом Бей та іншими, на Європейській конференції з комп'ютерного зору у 2006 році. Застосування алгоритму запатентовано у США[4].

Алгоритм базується на тих самих принципах та етапах, що і SIFT; але деталі на кожному кроці різні. Складається з трьох основних частин: виявлення проміжних точок, опису локальної сусідства та відповідності.

Спочатку виділяються деякі ключові точки зображення та невеликі ділянки навколо них. Останні обираються для полегшення розпізнавання об'єкту на зображенні. Виділяються саме малі ділянки, бо чим менше ділянка, тим менше на неї мають вплив великомасштабні спотворення. Так, якщо загальний об'єкт схильний до ефекту перспективи, то для меншої його частини перспективою можна знехтувати та замінити на зміну масштабу. Так само невеликий поворот об'єкту не матиме значного впливу на малі ділянки, навіть якщо картина загалом зміниться. На рис. 1.7 показано, що, не зважаючи на те, що об'єкт пошуку має інший масштаб, кут огляду та частково закриті іншим об'єктом, ключові точки досить точно ідентифікуються.

Метод шукає особливі точки за допомогою матриці Гессе, що дозволяє досить гарно детектувати плями, кути та краї ліній. Для кожної з цих точок рахується напрям максимальної зміни яскравості та масштаб, що береться із масштабного коефіцієнта матриці Гессе.



Рисунок 1.7 – Порівняння особливих точок на зразку та на сцені

Після цього формуються дескриптори для кожної з ключових точок – тобто набори з 64 або 128 чисел. Вони відображають відхилення градієнта оточень ключових точок, що знаходяться за допомогою фільтра Хаара.

1.3 Вибір методів розробки

Умовно можна поділити завдання на дві частини – визначення місця на зображенні, де знаходиться обличчя та підтвердження належності облич з серії зображень одній і тій самій людині. У якості вхідних даних до програми надходитиме три фотографії, де людина повинна бути сфотографована у анфас, з лівого профілю і з правого профілю. Маючи на увазі вищесказане, можна приймати рішення щодо вибору методів, що будуть використовуватись у програмі.

1.3.1 Вибір методів для пошуку облич на зображенні

З усіх методів розпізнавання, що розглядались у попередньому підрозділі, найбільш актуальним для задачі пошуку обличчя людини на зображенні є метод Віоли-Джонса. Він достатньо швидкий – настільки, що навіть дозволяє знаходити шукані об'єкти у режимі реального часу. Разом із цим, він має достатньо невеликий відсоток помилкових спрацьовувань за умови використання гарно натренованих каскадів. Він аналізує зображення на відповідність деяким ознакам, що властиві для шуканого класу об'єктів, а на основі цього аналізу приймається рішення щодо належності або неналежності зображеного об'єкту до шуканого класу об'єктів.

Цей метод був запропонований Паулом Віолою та Майклом Джонсом у 2001 році. Початковою задачею було розпізнавання обличч, проте цей метод дозволяє розпізнавати різноманітні класи зображень. Метод має безліч реалізацій, зокрема, у бібліотеці комп'ютерного зору OpenCV, яка і буде використовуватись у поточній атестаційній роботі.

Для ефективного використання цього методу потрібно спочатку змінити кольорову модель фото – перевести його у градації сірого, а потім виділити місце на фото, де повинно бути обличчя. Оскільки ми не можемо точно виділити таку область на зображенні, що прийде на обробку, необхідно послідовно обробити фрагменти зображення. Із цим нам допоможе принцип скануючого вікна. Його алгоритм полягає у наступному:

а) обирається мінімально можливий розмір скануючого вікна – області зображення, що буде скануватись на відповідність каскадові;

б) вікно обраного розміру рухається по зображенню починаючи з точки $(0, 0)$ із заданим кроком, скануючи кожну область, що потрапила у це вікно;

в) якщо вікно дійшло до кінця зображення, розмір вікна збільшується на задану кількість відсотків, але не більше заданого максимального розміру

вікна. У разі якщо вікно збільшується, переходимо до пункту б), інакше – виходимо з циклу.

Оскільки заздалегідь відомо, що надіслані фотографії зображуватимуть людину саме у анфас, саме з лівого та саме з правого профілю, для визначення місцезнаходження обличчя на кожній з цих фотографій будуть використовуватись відповідні каскади. Отож, це також момент деякої валідації надісланих даних, бо якщо на зображенні зображено щось інше або людина зображена з іншого ракурсу, повернеться помилка та подальша обробка даних здійснюватись не буде.

1.3.2 Вибір методу розпізнавання обличчя з різних ракурсів

Зазвичай, торкаючись теми розпізнавання людини на зображенні, опираються на деякі еталонні дані (наприклад, на заздалегідь зроблене фото людини), що порівнюються із вхідними даними, а на основі цього порівняння роблять висновок про те чи зображена на фотографії та чи інша людина. Це дуже гарно працює, якщо порівнюються фотографії з одного і того ж ракурсу та якщо є деяка база даних із фотографіями людей, яких буде розпізнавати програмний застосунок. У нашому додатку не передбачено функцій виявлення особистості на фотографії, проте з ціллю валідації прийнятих даних необхідно спроектувати механізм визначення того, чи належать фотографії одній і тій же людині чи ні. Завдання ускладнюється тим, що фотографії зроблені із різних ракурсів.

Тож, для здійснення перевірки належності обличчя одній і тій самій людині, було вирішено вирахувати співвідношення між частинами обличчя на усіх фотографіях та порівняти цей коефіцієнт. Цей коефіцієнт має збігатись для всіх зображень із припустимою похибкою Δ , введеною зважаючи через деяку ймовірність похибки алгоритму пошуку частин обличчя.

Вхідні дані до алгоритму – це три фрагменти надісланих зображень, саме ті, де є обличчя. Далі знаходимо необхідні нам частини облич за допомогою вже знайомих нам каскадів Хаара та методу скануючого вікна – знаходимо місцезнаходження очей, носу та рота.

Далі підраховуємо співвідношення відстаней від ока до носа та від носа до рота $\frac{h_i(\text{eye}, \text{nose})}{h_i(\text{nose}, \text{mouth})}$ для зображень облич з лівого та з правого профілів, а також з правої та лівої частин зображення обличчя у анфас. Після підрахунків отримуємо деякий коефіцієнт k для кожного з цих положень обличчя: для лівого профілю k_{LP} , для лівої частини фронтального обличчя k_{LF} , для правого профілю k_{RP} та для правої частини обличчя k_{RF} . Підраховуємо Δ_L та Δ_R за наступною формулою:

$$\begin{aligned} \Delta_L &= k_{LP} - k_{LF} \\ \Delta_R &= k_{RP} - k_{RF} \end{aligned} ;$$

Якщо $\Delta_L > \Delta$ або $\Delta_R > \Delta$, обличчя не вважаються належними одній і тій самій людині.

1.4 Постановка задачі

Для досягнення мети роботи необхідно розробити застосунок, що буде визначати місцезнаходження облич людини на серії знімків, а також перевіряти чи одна й та ж людина представлена на знімках. Фотографії людини мають бути зроблені з трьох ракурсів: в анфас, з лівого профілю та з правого профілю.

Було вирішено використовувати зображення людини з трьох різних ракурсів, тому що використовуючи саме такі знімки у подальшому можна буде створити максимально схожу на оригінальне обличчя 3D-модель.

Причиною, за якою було обрано саме вищезгадані три ракурси, стало оптимальне співвідношення складності роботи та якості подальшого результату. Зображень з цих ракурсів достатньо, аби у подальшому створити достатньо точну 3D-модель обличчя людини; збільшення кількості зображень з різних кутів є надмірним, а якщо залишити лише фото у анфас та одне фото у профіль, модель може вийти недостатньо точною.

Логічно, що така функція мала б більший попит у багатокористувацьких іграх, аніж у однокористувацьких, адже наявність у грі інших людей стимулює приділяти більше уваги до зовнішності персонажу. Аби не навантажувати комп'ютер користувача обробкою зображень, доцільніше буде розбити додаток на дві частини – клієнтську та серверну.

Клієнтська частина буде відправляти обрані фотографії на сервер та обробляти отриману від нього відповідь. Якщо сервер повертає помилку, клієнт повідомить про це користувача. Інакше клієнтська частина повинна відобразити фотографію та виділити на ній місцезнаходження обличчя.

Серверна частина повинна приймати запити від користувачів та обробляти їх: на кожній з фотографій знаходити обличчя та перевіряти чи належать усі фотографії одній і тій самій людині.

Отож, у рамках поточної роботи необхідно виконати наступні задачі:

- провести аналіз існуючих методів розпізнавання образів;
- реалізувати функцію пошуку облич на знімках у анфас і в профіль;
- реалізувати елементарну верифікацію даних, тобто у даному випадку, перевірку на те, що всі надіслані фотографії були зроблені із правильних ракурсів та на них зображена одна і та ж людина;
- реалізувати простий клієнтський додаток, що буде надсилати фотографії на сервер, який буде їх обробляти, та відобразити отриманий з серверу результат.

2 ТЕОРЕТИЧНІ ВІДОМОСТІ

2.1 Алгоритми, використані при пошуку обличчя

Для пошуку зображення на обличчі було вирішено використовувати метод Віоли-Джонса. Оскільки він працює із зображеннями у градаціях сірого, спочатку потрібно змінити кольорову модель зображення із стандартного RGB до градацій сірого.

Далі, за допомогою методу Віоли-Джонса на кожному зображенні шукатиметься обличчя. Оскільки цей метод лише виявляє чи є обличчя на вказанному фрагменті картинки, але не задає цей фрагмент, потрібно буде задавати його методом скануючого вікна.

2.1.1 Метод скануючого вікна

Метод скануючого вікна досить часто використовують разом із алгоритмами детектування, наприклад, все ж тим методом Віоли-Джонса.

Загальною ідеєю є наступне: виділяється деякий розмір скануючого вікна, що переміщується зліва направо та зверху донизу по зображенню із деяким кроком по осям. Частина зображень, що потрапили в це вікно обробляються методом Віоли-Джонса.

Алгоритм цього методу наступний [7]:

- задамо розмір скануючого вікна $H_{sw} \times W_{sw}$, де H_{sw} та W_{sw} – висота та ширина скануючого вікна відповідно, а також кроки за вертикальною та горизонтальною осями S_v та S_h , крок збільшення розміру вікна k та максимальні розміри вікна $\max H_{sw}$ та $\max W_{sw}$;

- нехай оригінальне зображення I має висоту H та ширину W . Необхідно доповнити його по краям так, аби його розміри були кратні розміру скануючого вікна та заданим за осями крокам. Розміри доповненого зображення \hat{I} обчислюються за допомогою таких виразів:

$$\hat{H} = \left[\frac{\left[\frac{H}{H_{SW}} \right] \cdot H_{SW}}{S_v} \right] \cdot S_v; \quad (2.1)$$

$$\hat{W} = \left[\frac{\left[\frac{W}{W_{SW}} \right] \cdot W_{SW}}{S_h} \right] \cdot S_h; \quad (2.2)$$

де $[\cdot]$ – операція округлення до найближчого меншого цілого числа.

а) обчислимо кількість кроків скануючого вікна по вертикалі та по горизонталі за наступними формулами:

$$C_v = \frac{\hat{H} - H_{SW}}{S_v} + 1; \quad (2.3)$$

$$C_h = \frac{\hat{W} - W_{SW}}{S_h} + 1; \quad (2.4)$$

б) нехай y та x задають положення верхнього лівого кута скануючого вікна за вертикаллю та горизонталлю відповідно. Формуємо фрагмент зображення розміром $H_{SW} \times W_{SW}$ з координатами верхнього лівого кута у точці y по вертикалі та x по горизонталі сканованого зображення \hat{I} :

$$i_{y,x} = \hat{I}[y, x, y + H_{SW}, x + W_{SW}]; \quad (2.5)$$

в) збільшуємо значення $x = x + S_h$ та застосуємо наступне правило:

1) якщо $x \geq \hat{W}$, то: $x = 0$ та $y = y + S_v$;

2) якщо $y < \hat{H}$, йдемо до пункту г;

г) задаємо новий розмір вікна $H_{sw} = H_{sw} + k$, $W_{sw} = W_{sw} + k$. Якщо $H_{sw} > \max H_{sw}$ або $W_{sw} > \max W_{sw}$, припиняємо рухати вікно, інакше йдемо до пункту б.

2.1.2 Метод Віоли-Джонса

До характеристик алгоритму Віола-Джонса [8], що роблять його оптимальним алгоритмом виявлення для вирішення поставленої задачі, належать:

- надійність – дуже висока швидкість виявлення (справжня позитивна швидкість) і дуже низька ймовірність похибки;
- можливість роботи у режимі реального часу – для практичних застосувань потрібно обробляти щонайменше 2 кадри на секунду.
- використовується лише для виявлення обличчя (не розпізнавання) – мета полягає в тому, щоб відрізнити обличчя від не обличчя, адже детектування є першим кроком у процесі розпізнавання.

Алгоритм складається з чотирьох етапів:

- вибір ознак Хаара;
- створення інтегрального зображення;
- тренінг Adaboost;
- каскадні класифікатори.

Важливо докладно розглянути кожен з етапів цього алгоритму аби отримати глибинне його розуміння.

2.1.3 Ознаки Хаара

Метод широко використовується у розпізнаванні образів. Хоча він створювався здебільшого для знаходження на зображеннях місцезнаходження обличь, за допомогою нього можна розпізнавати будь-які об'єкти.

У його основі лежить порівняння деякого фрагменту зображення із заданим набором шаблонів, що допоможуть ідентифікувати на зображенні об'єкт шуканого класу. Цими шаблонами є так звані ознаки Хаара або примітиви Хаара. Загалом шаблони є деякою сукупністю яскравих та затемнених зон на тій чи іншій частині зображення.

Для прикладу розглянемо базу даних з людськими обличчями. Загальним для всіх зображень є те, що область в районі очей темніше, ніж область в районі щік. Отже загальним ознакою Хаара для облич є два суміжних прямокутних регіону, що лежать на очах і щоках.

Існують прямокутні та похилі ознаки Хаара. Найпростішу прямокутну ознаку можна визначити як різницю сум пікселів двох суміжних областей всередині прямокутника, який може займати різні положення та масштаби на зображенні. Такий вигляд ознак називається 2-прямокутним. Віола та Джонс також визначили 3-прямокутні та 4-прямокутні ознаки. Кожна ознака може показати наявність або відсутність якоїсь конкретної характеристики зображення, такої як кордони чи зміни текстур. Наприклад, 2-прямокутна ознака може показати де знаходиться кордон між темним та світлим регіонами.

Задля збільшення розмірності простору ознак, Лінхарт та Майд [5] представили ідею похилих під кутом 45° ознак Хаара. Спосіб виявився вдалим та деякі похилі ознаки були здатні краще розпізнавати об'єкти. Наприклад, 2-прямокутна похила ознака Хаара може показати наявність краю, що нахилений на 45 градусів.

Мессом та Барзак [6] доповнили концепцію похилих ознак Хаара. Хоча ідея і є математично вірною, на практиці за використання ознак під різними кутами виникають проблеми. Для прискорення обчислень, детектор використовує зображення низької роздільної здатності, що призводить до помилки округлення. Виходячи з цього, похилі ознаки Хаара зазвичай не використовуються.

Якщо застосувати ці ознаки одразу до усієї картинки, обличчя знайдені не будуть, тому що вони будуть меншими за шаблони. Аби шукати на фото обличчя різних розмірів, використовується метод скануючого вікна. Саме всередині кожного такого вікна і вираховуються примітиви. Вікно ніби ковзає за всім зображенням. Після кожного проходження вікно збільшується, аби знайти обличчя більшого масштабу.

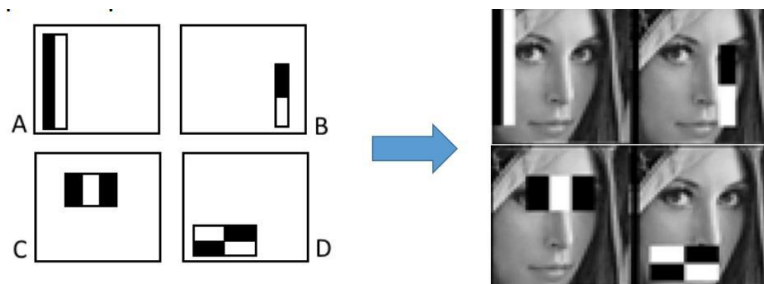


Рисунок 2.1 – Демонстрація застосування примітивів Хаара

Ознаки, які шукаються системою визначення, універсальним чином включають в себе суми пікселів зображення в прямокутних областях. Таким чином, вони мають деяку схожість з базовими ознаками Хаара, які раніше успішно використовувались в області виявлення об'єктів на основі зображення. Однак, оскільки ознаки, які використовують Віола та Джонс, покладаються на більше ніж одну прямокутну область, вони, як правило, складніші.

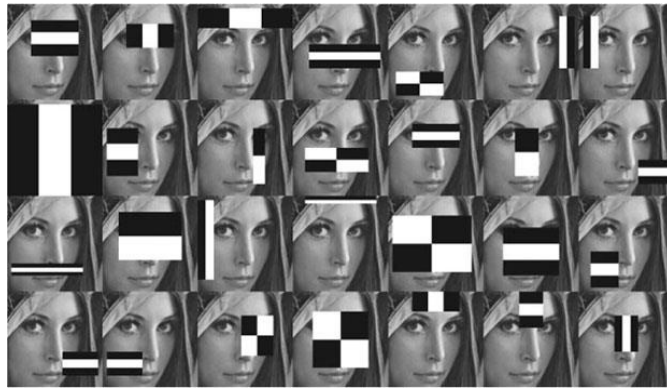


Рисунок 2.2 – Візуалізація алгоритму Віоли-Джонса

Значенням будь-якої заданої ознаки є сума пікселів у чітких прямокутниках, вирахованих із суми пікселів у затінених прямокутниках. Прямокутні особливості такого роду є примітивними у порівнянні з альтернативами, такими як керовані фільтри. Не зважаючи на те, що вони чутливі до вертикальних та горизонтальних особливостей, їх відгуки значно грубіші.

Всі людські обличчя мають подібні властивості. Ці закономірності можуть бути узгоджені з використанням ознак Хаара.

Приклади кількох властивостей, загальних для людини:

- область очей темніша, ніж верхня частина щоки;
- область носу є яскравішою, ніж очі.

Приклади властивостей, що формують риси обличчя:

- за розташуванням та розміром: очі, рот, ніс;
- за значенням: орієнтовані градієнти інтенсивності пікселів.

Особливості ознак:

- значення дорівнює різниці суми пікселів у темній частині до суми пікселів у світлій частині;
- загалом три типи ознак: 2-прямокутні, 3-прямокутні, 4-прямокутні (Віола і Джонс використовували 2-прямокутні);
- кожна ознака знаходиться у певному положенні всередині вікна пошуку.

Представлення зображення, яке називається інтегральним зображенням, оцінює прямокутні функції в постійному часі, що дає їм значну перевагу швидкості перед більш складними альтернативними функціями. Через те, що прямокутна область кожного об'єкта завжди прилягає до щонайменше одного іншого прямокутника, впливає, що будь-яка дво-прямокутна функція може бути розрахована у шести версіях масиву, у будь-якому трьох-прямокутному об'єкті з восьми, а в будь-якому чотирикутнику – з дев'яти.

Нажаль, швидкість підрахунку ознак Хаара недостатньо компенсує їх кількість. Наприклад, у стандартному вікні пошуку розміром 24x24 пікселі існує загальна кількість можливих функцій $M = 162, 336$, і було б надто довго оцінювати їх усі при тестуванні зображенні. Тому фреймворк детектування зображень надає варіант навчального алгоритму AdaBoost для того, щоб і вибрати найкращі ознаки, і натренувати класифікатор їх виконувати. Цей алгоритм конструює сильний класифікатор як лінійну комбінацію зважених простих класифікаторів:

$$h(x) = \text{sgn} \left(\sum_{j=1}^M \alpha_j h_j(x) \right).$$

Кожен слабкий класифікатор – це порогова функція, заснована на функції f_j .

$$h_j(x) = \begin{cases} -s_j & \text{if } f_j < \theta_j \\ s_j & \text{otherwise} \end{cases}.$$

Порогова вартість θ_j і полярність $s_j \in \pm 1$ визначаються у процесі навчання, так само, як і коефіцієнти α_j .

Цей класифікатор може виступати як перший шар серії, щоб відфільтрувати більшість негативних вікон. Другий шар з 10 функціями може

відкинути менш очевидно негативне вікно, що пройшли через 1-й шар, і так далі. Каскад поступово більш складних класифікаторів досягає ще кращого рівня виявлення. Оцінка надійних класифікаторів, створених в процесі навчання, може бути виконана швидко, але вона недостатньо швидка для запуску в режимі реального часу. З цієї причини сильні класифікатори розташовуються в каскаді в порядку складності, де кожен наступний класифікатор навчається тільки на тих обраних зразках, які проходять через попередні класифікатори. Якщо на будь-якому етапі каскаду класифікатор відкидає сегмент зображення, що знаходиться у вікні, подальша обробка не виконується і продовжується пошук в наступному вікні. Тому каскад має форму виродженого дерева.

Наприклад, у випадку знаходження обличч, перший класифікатор у каскаді використовує лише дві ознаки, таким чином, кількість помилкових відкидань обличч знизилась майже до 0%, а кількість помилково розпізнаних на цьому етапі обличч до 40%. Ефект цього єдиного класифікатора полягає в тому, щоб зменшити кількість разів, коли весь каскад оцінюється, приблизно наполовину.

У каскадному режимі кожен етап складається з сильного класифікатора. Таким чином, всі функції згруповані в кілька етапів, де кожен етап має певну кількість функцій.

Робота кожного етапу полягає в тому, щоб визначити, чи є певне вікно, на якому безумовно немає обличчя, чи то там все ж таки може бути обличчя. Дане вікно негайно відкидається як не обличчя, якщо воно терпить невдачу на одному з етапів.

У результаті навчання формується сукупність ознак для детекції певного класу об'єктів на зображенні. Дані згруповуються у XML файл, який можна використовувати для розпізнавання схожих об'єктів.

2.2 Визначення факту належності облич одній і тій самій людині

Будь-які вхідні дані потрібно валідувати. До нашого програмного додатку із клієнтської частини надходитиме три фотографії людини із різних ракурсів і ці дані також треба провалідувати за такими критеріями:

- усі надіслані фото зроблені із зазначених ракурсів;
- усі надіслані фото належать одній і тій самій людині.

Як вже говорилося у пункті 1.3.2, у додатку не передбачено можливості ідентифікації особистості за фотографією. Отож, для вирішення поставленої задачі було вирішено використовувати порівняння співвідношень різних частин обличчя із різних фотографій і на основі цього вирішувати чи належать фотографії одній і тій самій людині.

Коротко алгоритм був розглянений у тому ж таки пункті 1.3.2. У цьому ж розділі він наведений більш докладно.

Оскільки каскади Хаара, хоча і є достатньо точними для визначення обличчя, проте не є достатньо точними для пошуку таких об'єктів, як очі, ніс та рот, необхідно буде фільтрувати отримані «сирі» дані розпізнавання за методом Віоли-Джонса. Фільтрування буде проходити шляхом логічної вибірки найбільш вірогідних значень та сегментування зображення з ціллю звужити область пошуку конкретної частини обличчя лише до тієї зони, де, як передбачається, вона повинна бути.

Вхідні дані: три фрагменти надісланих зображень, що містять розпізнані за допомогою каскадів Хаара обличчя.

Вихідні дані: булеве значення – TRUE, якщо за цим алгоритмом обличчя належать одній і тій самій людині, FALSE у іншому разі.

Послідовність алгоритму зазначена нижче.

Крок 1. Визначаємо тип зображення – у анфас, з лівого профілю чи з правого профілю. Ці дані повинні надійти від клієнтського додатку разом із самими фото. Якщо це зображення з правого профілю, віддзеркалити його за віссю Y.

Крок 2. За допомогою каскадів Хаара знаходимо на зображенні усе, що каскад Хаара розпізнає як око. Отримуємо масив прямокутників *eyes*.

а) Якщо це зображення у анфас, знаходимо у *eyes* два прямокутника із найменшими значеннями *X* – назвемо їх *eyesEnFace*.

б) Якщо це зображення у лівий чи правий профіль, знаходимо лише один такий прямокутник – назвемо його *eyeProfil*.

Крок 3. Відсікаємо верхню частину зображення на *hCutted* пікселів:

Якщо зображення у анфас:

$$hCutted = \min(eyesEnFace[i].y + eyesEnFace[i].height).$$

а) Інакше:

$$hCutted = eyeProfil.y + eyeProfil.height.$$

Крок 4. Якщо це зображення у лівий чи правий профіль, відсікаємо четверту частину зображення з права – сегментуємо зображення, відсікаючи непотрібні на цьому етапі і, в той же час, заважаючи подальшому коректному розпізнаванню елементи – вуха.

Крок 5. Виконуємо другу частинку кроку 2, шукаючи ознаки носу замість ока, отримуємо – *nose*.

Крок 6. Оскільки ми обрізали зображення зверху, додаємо до *nose.X* значення *hCutted*.

Крок 7. Виконуємо другу частину кроку 3, тільки не для *eyeProfil*, а для *nose*. Знову отримуємо відсічене зображення.

Крок 8. Виконуємо другу частину кроку 2, шукаючи рот замість ока, отримуємо – *mouth*.

Крок 9. Повторюємо крок 6, беручи *mouth* замість *nose*.

Крок 10. Для кожного з розпізнаних прямокутників шукаємо медіанну лінію – це потрібно для того, щоб виключити спотворення аналізу через можливі різні розміри результуючого прямокутника із шуканою ознакою:

$$M_{rect} = rect.y + rect.width;$$

Крок 11. Підраховуємо співвідношення відстаней між медіанами частин обличчя для різних положень обличчя:

$$k_i = \frac{h_i(eye, nose)}{h_i(nose, mouth)};$$

де k_i – коефіцієнт відстаней між частинами обличчя для обраного типу обличчя (для обличчя у анфас підраховуються два значення – різні для лівого та правого ока), $h_i(eye, nose)$ – відстань між медіанами ока та носа, $h_i(nose, mouth)$ – відстань між медіанами носа та рота.

Крок 12. Підраховуємо значення Δ_I для лівого та правого боку:

$$\begin{aligned} \Delta_L &= k_{LP} - k_{LF} \\ \Delta_R &= k_{RP} - k_{RF} \end{aligned};$$

де Δ_L – різниця між коефіцієнтами відстаней між частинами обличчя для лівого боку обличчя, Δ_R – різниця між коефіцієнтами відстаней для правого боку обличчя, k_{LP} – коефіцієнт відстаней між частинами обличчя для лівого профілю, k_{LF} – коефіцієнт відстаней між частинами обличчя для лівої частини фронтального обличчя, k_{RP} – коефіцієнт відстаней між

частинами обличчя для правого профілю, k_{RF} – коефіцієнт відстаней між частинами обличчя для правої частини обличчя

Крок 13. Перевіряємо умови $\Delta_L > \Delta$ та $\Delta_R > \Delta$, де Δ – заздалегідь задана максимальна різниця коефіцієнтів. Якщо виконується хоча б одна з цих умов, обличчя на фотографіях не належать одній і тій самій людині.

Припустима похибка Δ введена через нездатність знаходження точного місцезнаходження на фотографії окремих частин облич за допомогою каскадів Хаара – цей алгоритм не ідеальний та загалом дає найкращі результати у пошуку саме облич на зображеннях, а не окремих його частин.

3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ДОДАТКУ

3.1 Вибір технологій розробки

Перед тим, як почати проектування архітектури програмного додатку, потрібно спочатку подумати над такими важливими питаннями, як: «Що буде робити мій застосунок?», «Для кого він буде це робити?», «Яких результатів буде очікувати користувач?», «Як програма буде виконувати свою роботу?», «Чи критично використовувати під час роботи додатку максимум ресурсів комп'ютера або ж важливіше знехтувати цим на користь швидкості розробки?», «Який додатковий функціонал окрім основного мені потрібні? Які додаткові додатки?», «Яким я бачу подальший розвиток цього проекту?» та інші. Відповіді на ці запитання – це те, на що потрібно орієнтуватись під час проектування додатку та вибору технологій, що будуть використовуватись під час розробки.

3.1.1. Бібліотека OpenCV

Зокрема, поточний додаток має реалізовувати функції пошуку та розпізнавання зображень. Класичним засобом вирішення цієї проблеми зазвичай стає бібліотека комп'ютерного зору OpenCV [10] – це біблія світу комп'ютерного зору. Бібліотека просто та швидко встановлюється, має під капотом реалізовані популярні алгоритми розпізнавання образів (як, наприклад, каскади Хаара або ЛБШ моделі) та достатньо проста у використанні. Сама по собі вона може вільно використовуватись у будь-якому програмному забезпеченні. Вона розроблена на C/C++ та доступна для таких мов, як Java та Python. Ще є декілька обгорток для C#, але деякі з них нестабільні, а деякі мають обмеження ліцензії. Тож, обирати доводиться із попередніх трьох мов.

3.1.2 Мова програмування Java

Вибір пав на мову Java – швидка розробка, кросплатформеність та зростаюча популярність цієї мови схиляє чашу терезів на її користь.

Java є мовою програмування загального призначення, яка є багатопотоковою та об'єктно-орієнтованою, а також спеціально розроблена, щоб мати стільки імплементацій залежностей наскільки це можливо. Вона призначена, щоб розробники додатків "написали (додаток – прим. авт.) один раз, і він працював скрізь" (WORA) [9]. Це означає, що скомпільований Java код може працювати на всіх платформах, що підтримують Java, без необхідності перекомпіляції. Java додатки, як правило, збираються у байт-код що виконується завдяки віртуальній машині Java (JVM).

На сьогоднішній час Java є однією з найпопулярніших мов програмування, особливо для розробки клієнт-серверних веб-додатків.

Java була розроблена Джеймсом Гослінгом в Sun Microsystems (яка згодом була придбана корпорацією Oracle) і випущена в 1995 році в якості основного компонента Java платформи Sun Microsystems. Мова запозичила більшість синтаксису з C і C ++, але вона має менше низькорівневих можливостей, ніж будь-яка з них. Як казав Джеймс Гослінг: «Java – це C++, з якого вилучили всі пістолети, ножі та дубинки».

3.1.3 Spring Boot

Spring Framework (або просто Spring) – це універсальний фреймворк з відкритим вихідним кодом для Java -платформи. Фреймворк був вперше випущений під ліцензією Apache 2.0 license в червні 2003 року . Перший стабільний реліз 1.0 був випущений в березні 2004.

Незважаючи на те, що Spring не забезпечував якусь конкретну модель програмування, він став широко поширеним в Java-співтоваристві головним чином як альтернатива і заміна моделі Enterprise JavaBeans . Spring надає

велику свободу Java-розробникам в проектуванні; крім того, він надає добре документовані і легкі у використанні засоби вирішення проблем, що виникають при створенні додатків корпоративного масштабу.

Spring дуже популярна через кілька причин:

- Dependency Injection спрощує написання тестового коду, та коду загалом;
- Проста у використанні, але потужна можливість керування транзакціями бази даних;
- Spring спрощує інтеграцію з іншими платформами Java, такими як JPA/Hibernate ORM, Struts/JSF та іншими веб-фреймворками;
- підтримка архітектури Web MVC для створення веб-додатків.

Окрім Spring Framework, існує багато інших проектів сімейства Spring, що допомагають створювати застосування, що відповідають сучасним потребам бізнесу, зокрема:

- Spring Data: спрощує доступ до даних з реляційних і NoSQL баз даних;
- Spring Security: впроваджує фреймворк захисту, що дозволяє керувати безпекою системи.

Загалом Spring Framework є дуже зручною платформою, що дозволяє уникати безлічі одноманітної тривіальної роботи. Тому було вирішено використати його RestController у зв'язці із Service. А завдяки заданню кількох змінних у властивостях додатку, час на налаштування програмного засобу значно скоротився.

3.1.4 Maven

Apache Maven — фреймворк для автоматизації збірки проектів на основі опису їх структури у файлах на мові POM, який є підмножиною XML.

Maven є дуже популярним фреймворком. Вона дозволяє буквально у кілька строчок налаштувати збірку проекту та підключити до нього усі необхідні залежності та плагіни. При чому, їх навіть не потрібно завантажувати або видаляти, якщо треба – це робиться автоматично одразу після написання або стирання кількох рядків, що описують цю залежність. Це надзвичайно зручно та корисно при розробці різноманітніших застосувань.

3.1.4 C# та WindowsForms

Було вирішено розробити клієнтський додаток на мові C# із використанням WindowsForms. Зазвичай за наявності серверного додатку на мові Java, доступ до системи надається користувачеві через веб-сторінки. Проте, рішення робити клієнтський додаток саме у вигляді настільного віконного додатку. По-перше, так значно швидше, аніж робити веб-сторінки. По-друге, у майбутньому планується розвинути цей програмний засіб таким чином, аби він з надходжених зображень ще й будував 3D-модель обличчя людини – це зручніше робити саме у настільному додатку. Передбачається також впровадження такого додатку безпосередньо у комп'ютерну гру, що також є причиною для того, аби відмовитись від ідеї із веб-сторінками.

До того ж, використовуючи C#, можна буде у ході розвитку програмного засобу переробити реалізацію клієнтської частини лише частково під нові умови. Існує багато бібліотек, що дозволять це зробити – як мінімум найвідоміші OpenGL та Unity3D. Але це не входить у множину задач поточної атестаційної роботи, тож докладніше на цьому можна не зупинятись.

3.2 Архітектура програмного засобу

3.2.1 Архітектура REST

REST (Representational state transfer) – це стиль архітектури програмного забезпечення для розподілених систем, що зазвичай використовується для побудови веб-служб. Термін був введений у 2000 році Роем Філдінгом, одним з авторів HTTP-протоколу. В основі REST закладено принципи функціонування Всесвітньої павутини і, зокрема, можливості HTTP. Філдінг розробив REST паралельно з HTTP 1.1 базуючись на попередньому протоколі HTTP 1.0. Системи, що підтримують REST, називають RESTful-системами.

Для передачі даних у REST зазвичай використовують HTTP протокол. Самі дані передаються у вигляді невеликої кількості стандартних форматів, найчастіше це HTML, XML, JSON [11].

Будь-який REST протокол (HTTP в тому числі) повинен підтримувати кешування, не повинен залежати від мережевого прошарку, не повинен зберігати інформації про стан між парами «запит-відповідь». Стверджується, що такий підхід забезпечує масштабованість системи і дозволяє їй еволюціонувати з новими вимогами.

Архітектурний стиль REST накладає такі архітектурні обмеження на програмні додатки:

- клієнт-серверна архітектура;
- відсутність стану – кожен запит є унарним та містить всю необхідну для його обробки інформацію, не покладаючись на те, що сервер знає щось із попереднього запиту;
- кешування;
- однорідний інтерфейс;
- поділ на шари абстракції.

3.2.2 REST архітектура у поточному додатку

Поточний програмний додаток має розділення клієнту та серверу. Клієнт тонкий – він лише передає інформацію на сервер та виводить прийнятий звідти результат.

Передача даних між клієнтом і сервером здійснюється у вигляді перекиду JSON даних через POST-запитання.

Клієнт надсилає серверові JSON формату, що приведений на рис. 3.1.

```
{
  "LP": "<image in base64 encoding>",
  "EF": "<image in base64 encoding>",
  "RP": "<image in base64 encoding>"
}
```

Рисунок 3.1 – Структура JSON даних, що клієнт посилає серверові

Оскільки не можна просто так взяти й переслати зображення у JSON, треба спочатку перетворити його у строку. Найуніверсальніший спосіб це зробити – це закодувати зображення у Base64 кодуванні.

Base64 – це стандарт кодування двійкових даних, при якому використовуються лише 64 символи ASCII. Абетка кодування містить латинські символи A-Z, a-z, 0-9 та ще два додаткових символа, що залежать від системи реалізації. Кожні три вихідних байти кодуються чотирма символами, тобто маємо збільшення даних на третину.

Оскільки у дійсності строки, що уособлюють закодовані зображення, дуже довгі, на ілюстрації власне їх немає.

Сервер приймає дані, кладе їх у структуру типу словник, де роль ключа виконують кодові назви положень облич (LP – left profil, тобто лівий профіль; EF – en face, тобто анфас та RP – right profil, тобто правий профіль), а роль значення – власне картинка.

Далі проходить пошук облич на цих зображеннях та верифікацію на їх належність одній і тій самій людині за вже описаними у другому розділі

алгоритмами. Демонстрація знаходження частин обличчя на зображенні із суто обличчям подано на рис. 3.2.

Якщо все проходить успішно, – обличчя знайдені та належать вони одній і тій самій людині, – клієнтові відправляється статус 200 ОК та дані у форматі JSON, що містять прямокутні області, що відповідають знайденим обличчям на кожній з фотографій, це продемонстровано на прикладі зображень, що генерувались під час мануального тестування додатку (рис. 3.3).



Рисунок 3.2 – Результати пошуку частин обличчя

```
{
  "faceCoordinates":
  {
    "EF":
    {
      "x":479,
      "y":1199,
      "width":1323,
      "height":1323
    },
    "LP":
    {
      "x":286,
      "y":1518,
      "width":1180,
      "height":1180
    },
    "RP":
    {
      "x":845,
      "y":1437,
      "width":1167,
      "height":1167
    }
  }
}
```

Рисунок 3.3 – Структура JSON-відповіді серверу, оптимістичний випадок

Якщо ж на сервері сталось виключення, повертається статус 500 Bad Request та структура JSON, що містить ім'я виключення, що наглядно продемонстровано на рис. 3.4.

```
{
  "name": "FaceNotFoundException"
}
```

Рисунок 3.4 – Структура JSON-відповіді серверу, песимістичний випадок

Для чого ж зроблено саме так? А справа у тому, що для деяких помилок є конкретна причина, що залежить від користувача. І, аби це виключення не з'являлось знову, користувач повинен припинити неправильне використання програмного продукту. Оскільки користувач, на жаль (або на щастя), не може читати думки розробників, потрібно якось йому натякнути на те, що винувата не машина та не злі програмісти.

```
private string GetExceptionMessage(string exceptionName)
{
    switch (exceptionName)
    {
        case "IOException":
            return "Вы отправили неверные данные.";
        case "FaceNotFoundException":
            return "На одном или нескольких изображениях лиц обнаружено не было.";
        case "FaceNotMatchException":
            return "Лица принадлежат разным людям.";
        default:
            return "Неизвестная ошибка.";
    }
}
```

Рисунок 3.5 – Лістинг методу отримання спеціального повідомлення про помилку

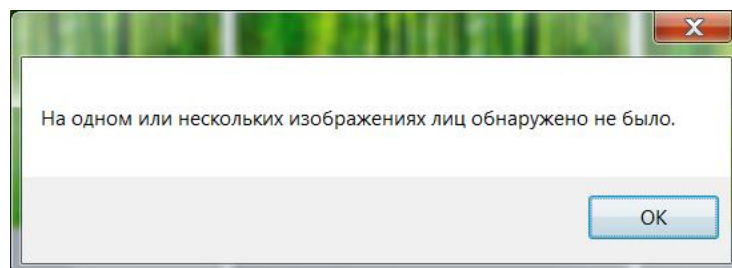


Рисунок 3.6 – Помилка, що показується користувачеві

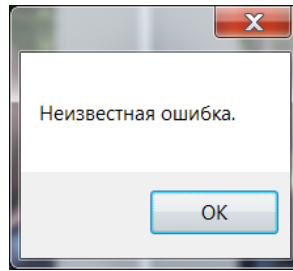


Рисунок 3.7 – Повідомлення про невідому помилку

Такі помилки мають унікальні типи та для кожного з них на клієнтській частині передбачене своє повідомлення про помилку із поясненням причин її виникнення (рис. 3.5), що показуються користувачеві у окремому віконці (рис. 3.6). Якщо ж виключення, що повернулось сервером, не входить до цієї множини, програма показує змістовне повідомлення (рис. 3.7).

3.3 Дизайн клієнтської частини

Клієнтська частина додатку має досить простий дизайн (рис. 3.8). Головний екран містить короткий опис того, що вимагається від користувача, кілька кнопок та кілька imagebox-ів, у яких будуть превью зображень.

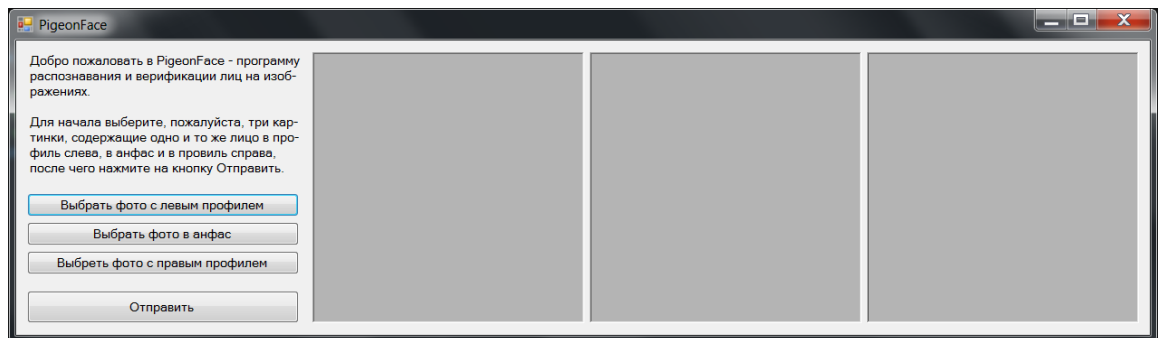


Рисунок 3.8 – Головний екран клієнтського додатку

За допомогою кнопок можна обрати зображення, що будуть відправлятися на сервер для розпізнавання. Коли це зроблено, на формі з'являються прев'ю зображення (рис. 3.9). Якщо клацнути на якомусь з цих прев'ю, відкриється окреме вікно із більшою версією зображення (рис. 3.10).

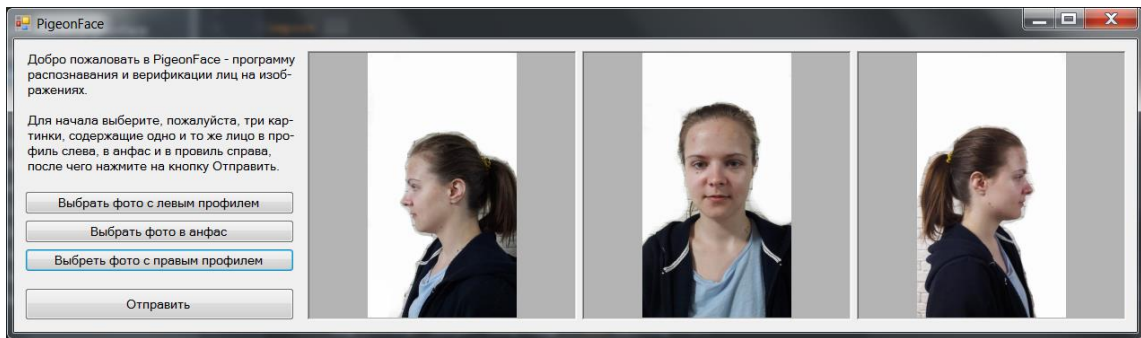


Рисунок 3.9 – Прев'ю зображень на формі

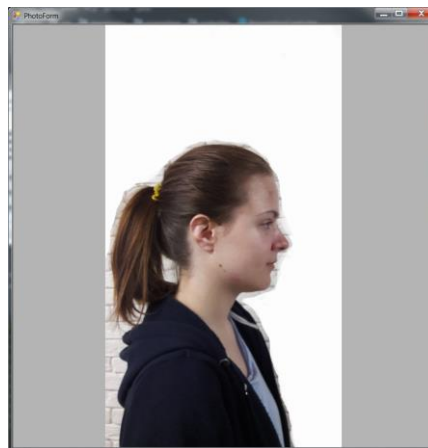


Рисунок 3.10 – Вікно із зображенням з фото з прев'ю

На формі присутня перевірка на те, щоб користувач не міг відправити запит на сервер, поки не обере усі три зображення (рис. 3.11).

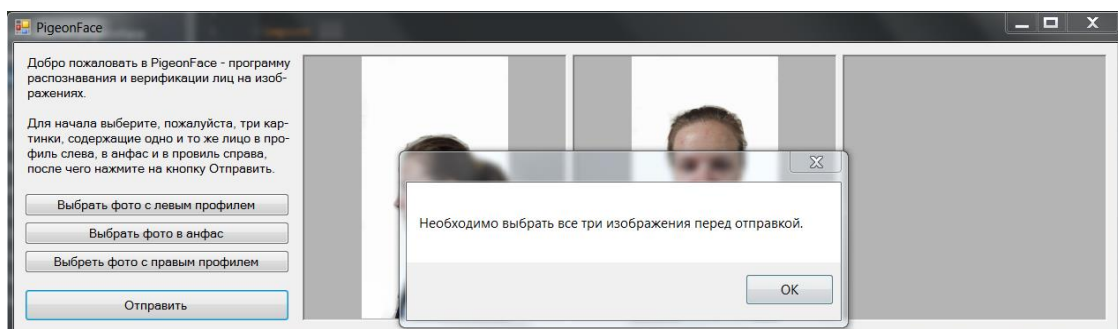


Рисунок 3.11 – Перевірка на наявність усіх зображень перед посиленням запиту на сервер

У разі коректної відповіді на зображеннях малюються прямокутники у місцях, де було знайдено обличчя. Це відображається на прев'ю (рис. 3.12).

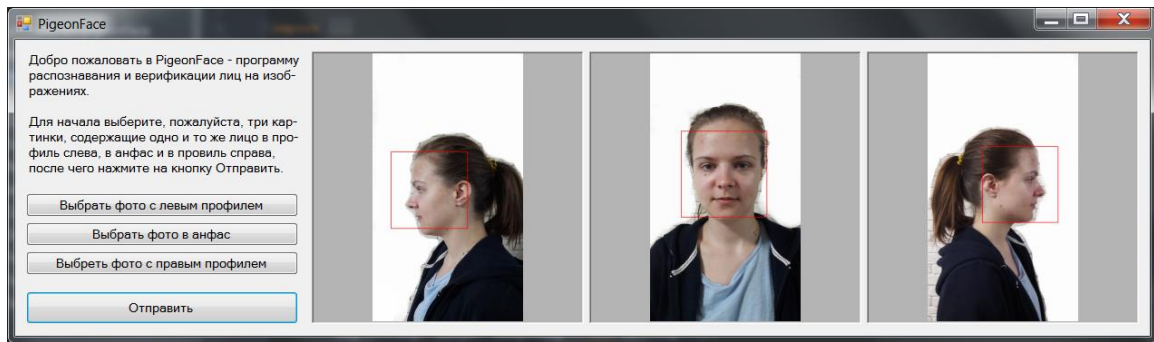


Рисунок 3.12 – Результат обробки відповіді з серверу

Частково інтерфейс також був показаний у попередньому розділі, а саме деякі повідомлення про помилки, що надіслані із серверу.

3.4 Інструкція користувача

3.4.1 Інструкція до встановлення та запуску серверної частини додатку

Програмний засіб має запускатись на комп'ютері, який відповідатиме наступним вимогам:

- мати встановлену Java 1.8 або вище;
- встановлений Maven;
- встановлений Tomcat.

Запустити програмний додаток можна за допомогою командного рядка. Після запуску програми подальшого нагляду вона не потребує. Вона сама буде обробляти зображення та відправляти результат цієї назад до клієнта.

3.4.2 Інструкція щодо використання клієнтської частини додатку

Якщо технічні вимоги для серверної частини не спирались на операційну систему, на якій буде розгортатись застосунок, бо та частина написана на кросплатформеній мові Java, то із клієнтською частиною так не вийде, тож необхідні та достатні умови такі:

- ОС сімейства Windows
- .NET версії 4.1

Аби почати роботу із додатком, необхідно просто запустити виконуваний файл PigeonFace.exe – далі з'явиться вікно додатку. Інтуїтивно зрозумілий інтерфейс та текстові підказки не кинуть користувача напризволяще.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.

4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкту, що мають вплив на персонал

У даному дипломному проєкті розробляється програмне забезпечення.

Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Виходячи з приведених характеристик, вочевидь, що для людини існує небезпека поразки електричним струмом, унаслідок недбалого поводження з комп'ютером і порушення правил експлуатації, залишення частин ПЕОМ, що знаходяться під напругою, відкритими або знятих для ремонту вузлів.

Відповідно до [12] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною (невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу слід приділяти біологічним ефектам низькочастотних електромагнітних полів (аж до порушення ДНК).

Відповідно до [13], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання якої може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищений або знижений рух повітря;
- підвищена або знижена вологість повітря;
- відсутність або недостатність природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

4.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка посилюється тим, що органи

чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне розділення мережі;
- використання малої напруги;
- ізоляція частин, що проводять струм;
- огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує години, протягом яких людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_k = \frac{U_\phi}{Z_\Pi + \frac{Z_T}{3}}, \quad (4.1)$$

де U_ϕ - номінальна фазна напруга мережі, В;

Z_Π - повний опір петлі, створене фазними і нульовими дротами, Ом;

Z_T - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [14]: $Z_T/3 = 0,1$ Ом.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу(4.2.) :

$$Z_\Pi = \sqrt{R_\Pi^2 + X_\Pi^2}, \quad (4.2)$$

де $R_\Pi = R_\phi + R_0$ - сумарний активний опір фазного R_ϕ і нульового R_0 дротів, Ом;

X_Π - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту $S = 2.5$ мм, тоді згідно таблицям 5 і 6 [14], має такий опір:

$$X_\Pi = 0,11 \text{ Ом};$$

$$R_\phi = 7,55 \text{ Ом};$$

$$R_0 = 7,55 \text{ Ом}.$$

$$\text{Отже, } R_\Pi = 7,55 + 7,55 = 15,1 \text{ Ом}.$$

Тоді по формулі (4.2) знаходимо повний опір петлі :

$$Z_\Pi = \sqrt{15,1^2 + 0,11^2} \approx 15,1 \text{ (Ом)}.$$

Струм однофазного короткого замикання рівний:

$$I_k = \frac{220}{15,1 + 0,1} = 14,47 \text{ (А)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$I_k \geq k \cdot I_n, \quad (4.3)$$

де I_n - номінальний струм спрацьовування плавкої вставки, А;

k - коефіцієнт кратності нелінійного струму I_n , А.

Коефіцієнт кратності нелінійного струму I_n розраховується по формулі (4.4.):

$$I_n = P / U, \quad (4.4)$$

де $P = 220$ Вт - споживана потужність;

$U = 220$ В - робоча напруга;

$k = 3$ А - для плавких вставок.

Отже, $I_n = 220 / 220 = 1$ А.

Підставивши значення у вираз (4.3), одержимо:

$$14,47 > 3 \cdot 1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування(і захист) при підвищенні номінального струму.

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються [19], ДБН, відповідними ГОСТами і ОСТАми з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами.

Мікроклімат виробничих приміщень визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до [12] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні. За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу(категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл. 4.1).

Таблиця 4.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22...24	40...60	0,1
Тепло	23...25	40...60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й устатовленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створінню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної

кількості свіжого повітря, визначеного в ДБН (30 м³ в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибрати найбільш шумозахисні матричні принтери або з великою швидкістю роботи (струменеві, лазерні). Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання (під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплону).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання : обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту ТСО'03.

ТСО'03 пред'являє жорсткі вимоги в таких областях: ергономіка (фізична, візуальна і зручність користування), енергія, випромінювання (електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [15].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до [19] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(E_n) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛП02П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки $a=12$ м, шириною $b=10$ м, заввишки $z=4$ м, використовуючи формулу (4.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.5)$$

де F - світловий потік = 3120 лм;

E - максимально допустима освітленість робочих поверхонь = 200 лк;

S - площа підлоги = 120 м²;

Z - поправочний коефіцієнт світильника = 1,2;

k - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

n - кількість світильників;

U - коефіцієнт використання освітлювальної установки = 0,6;

M - кількість ламп у світильнику = 2.

З формули (4.5) виразимо n (4.6) і визначимо кількість світильників для даного приміщення:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.6)$$

Отже, $n = (200 \cdot 120 \cdot 1,2 \cdot 1,5) / (3120 \cdot 0,6 \cdot 2) = 12$.

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 4.1.

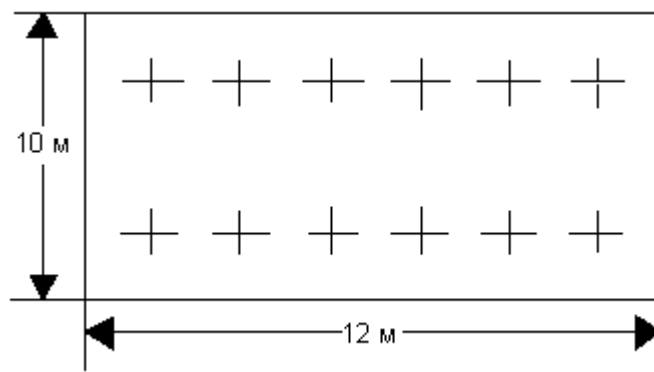


Рисунок 4.1 - Схема розташування світильників

4.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 °С ;

полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 °С, температура самозаймання 530 °С, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;

стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;

пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;

деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 °С, схильна до самозаймання.

Згідно [18] приміщення відносяться до категорії В (пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогнебезпечної зони класу П - Па (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегріву від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є : оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акропеїн, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза з коробкою марки В(жовта).

Пожежна безпека об'єктів народного господарства регламентується [18] і забезпечується системами запобігання пожежам і протипожежному захисту. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

- застосування устаткування, що задовольняє вимогам електростатичної безпеки;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;
- виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;
- підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з палим середовищем, нижче гранично допустимої, становить 80% якнайменшої температури самозаймання пального.
- заміна небезпечних технологічних операцій більш безпечними;
- ізолюване розташування небезпечних технологічних установок і устаткування;
- зменшення кількості палих і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;
- запобігання можливості утворення палих сумішей на лінії, вентиляційних системах і ін.;

- механізація, автоматизація та справність(потокова) виробництва;
- суворе дотримання стандартів і точне виконання встановленого технологічного режиму;
- запобігання можливості появи в небезпечних місцях джерел запалення;
- запобігання розповсюдженню пожеж і вибухів;
- використання устаткування і пристроїв, при роботі яких не виникає джерел запалення;
- виконання вимог сумісного зберігання речовин і матеріалів;
- наявність громовідводу;
- ліквідація можливості самозаймання речовин і матеріалів .

Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог :

- електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження і кондиціонування;
- система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;
- робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до правил пожежної безпеки для промислових підприємств приміщення категорії В підлягають устаткуванню системами автоматичної

пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1(димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для ліквідації пожежі використовують вогнегасники (хімічно-пінні, пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар (сокири, ломи, багри, шерстяну або азбестову ковдри). Як засіб індивідуального захисту проектом передбачається використання промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

ВИСНОВКИ

У результаті виконання атестаційної роботи було розроблено програмне забезпечення, що, по-перше, шукає обличчя на кожній фотографії з серії знімків людини у різних ракурсах, а, по-друге, порівнює коефіцієнти відстаней між ключовими об'єктами на обличчі, стежить, щоб вони були не більше певного заданого порогу – таким чином здійснюється перевірка на належність всіх облич одній і тій самій людині.

Підсумовуючи, у рамках поточної роботи були виконані наступні задачі:

- проведено аналіз існуючих методів розпізнавання образів;
- реалізовано функцію пошуку облич на знімках у анфас і в профіль;
- реалізовано елементарну верифікацію даних, тобто у даному випадку, перевірку на те, що всі надіслані фотографії зроблені із правильних ракурсів та на них зображена одна і та ж людина;
- реалізовано простий клієнтський додаток, що надсилає фотографії на сервер, який їх обробляє, та відображає отриманий з серверу результат.

Завдання було виконане саме у тому обсязі, у якому було поставлене.

Було вивчено кілька найвідоміших методів пошуку тих чи інших класів об'єктів на зображеннях, серед яких: каскади Хаара, локальні бінарні шаблони, а також більш складні алгоритми SIFT та SURF.

У подальшому планується покращення цього програмного додатку до такого, що буде здатне робити 3D-модель людини за серією знімків. Насамперед, для досягнення цієї мети, потрібно використання більш точних та складних алгоритмів для пошуку окремих деталей обличчя, бо каскади Хаара виявились недостатньо чутливими та точними для цього. До того ж, якщо, наприклад, використовувати алгоритм, що базується на пошуку та розміщенні на зображенні ключових точок (як SIFT, SURF, тощо), значно спрощується задача подальшого побудування 3D-моделі, адже вона також буде будуватись за ключовими точками.

У розділі «Охорона праці» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки, заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1) Нейросеть отпугнет метящих территорию котом. [Электронный ресурс] – Режим доступа [www/ URL : https://nplus1.ru/news/2016/07/11/ai-hits-cats](http://www.nplus1.ru/news/2016/07/11/ai-hits-cats)
- 2) Face Detection using OpenCV and Python: A Beginner's Guide [Электронный ресурс] – Режим доступа [www/ URL : https://www.superdatascience.com/opencv-face-detection/](http://www.superdatascience.com/opencv-face-detection/)
- 3) Lowe David G. "Object recognition from local scale-invariant features". / Proceedings of the International Conference on Computer Vision. 1999. №2. С. 1150 – 1157.
- 4) Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features" / Computer Vision and Image Understanding (CVIU). 2008. Vol.110, №3. С. 346 – 359.
- 5) Lienhart, R. and Maydt, J., «An extended set of Haar-like features for rapid object detection», ICIP02, pp. I: 900—903, 2002
- 6) Messom, C.H. and Barczak, A.L.C., «Fast and Efficient Rotated Haar-like Features Using Rotated Integral Images», Australian Conference on Robotics and Automation (ACRA2006), pp. 1-6, 2006
- 7) Суша А. «Алгоритм детектирования автомобильных номеров методом скользящего окна с использованием свёрточной нейронной сети» / Студенческий: электрон. научн. журн. 2017. № 9(9). URL: <https://sibac.info/journal/student/9/80477>
- 8) Viola–Jones object detection framework [Электронный ресурс] – Режим доступа [www/ URL : https://en.wikipedia.org/wiki/Viola–Jones_object_detection_framework](https://en.wikipedia.org/wiki/Viola–Jones_object_detection_framework)
- 9) Scott Guthrie: Silverlight and the Cross-Platform [Электронный ресурс] – Режим доступа: [www/ URL : https://channel9.msdn.com/shows/Going+Deep/Scott-Guthrie-Silverlight-and-the-Cross-Platform-CLR](https://channel9.msdn.com/shows/Going+Deep/Scott-Guthrie-Silverlight-and-the-Cross-Platform-CLR)

10) Глория Б. Г. и другие – «Обработка изображений с помощью OpenCV»: Москва, «ДМК Пресс», 2016. – 210 с.

11) REST [Електронний ресурс] – Режим доступу [www/ URL](http://www.URL) : <https://uk.wikipedia.org/wiki/REST>

12) ДСН 3.3.6.042-99 Державні санітарні норми мікроклімату виробничих приміщень. Режим доступу: [www. URL](http://www.URL): <https://zakon.rada.gov.ua/rada/show/va042282-99>

13) НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. Режим доступу: [www. URL](http://www.URL): <https://zakon.rada.gov.ua/laws/show/z0508-18>

14) ДСТУ 7237:2011 Національний стандарт України. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту. Режим доступу: [www. URL](http://www.URL): <https://zakon.rada.gov.ua/rada/show/ru/v0037831-11>

15) ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Режим доступу: [www. URL](http://www.URL): <https://zakon.rada.gov.ua/rada/show/v0007282-98>

16) ДБН В.2.5-67:2013. Опалення вентиляція та кондиціонування. Режим доступу: [www. URL](http://www.URL): <https://zakon.rada.gov.ua/rada/show/v0024858-13>

17) ДБН В.2.5-28-2006. Природне і штучне освітлення. Режим доступу: [www. URL](http://www.URL): <https://zakon.rada.gov.ua/rada/show/v0168667-06>

18) ДСТУ Б В.1.1-36-2016. Визначення категорії приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою. Режим доступу: [www. URL](http://www.URL): http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=65419

19) ДСП 173-96. Державні санітарні правила планування та забудови населених пунктів. Режим доступу: [www. URL](http://www.URL): <https://zakon.rada.gov.ua/laws/show/z0379-96>

20) Симметрон. Электронные компоненты. Каталог 2002, 2002г. – 192с.

ДОДАТОК А. ЕЛЕКТРОННІ ПЛАКАТИ

Кафедра комп'ютерних наук та інженерії

Інформаційна система розпізнавання людського обличчя

Студент КН-153
Байбус Р.І.

Керівник
доц. Барабрук В.М.

Актуальність роботи

Розпізнавання образів:

- Аутентифікація (FaceID)
- Автофокусування (камери гадж.)
- CV for AI
- Камери відеоспостереження (автономери, обличчя злочинців)
- Facebook
- SelfieMoji, маски для Instagram

Boston Dynamics

Постановка задачі

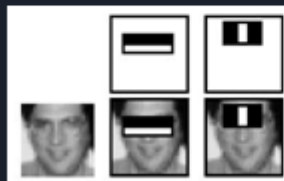
Задачі, які потрібно виконати у рамках атестаційної роботи:

- провести аналіз існуючих методів розпізнавання образів;
- реалізувати функцію пошуку обличчя на знімках у афас і в профіль;
- реалізувати елементарну верифікацію даних, тобто у даному випадку, перевірку на те, що всі надіслані фотографії були зроблені із правильних ракурсів та на них зображена одна і та ж людина;
- реалізувати простий клієнтський додаток, що буде надсилати фотографії на сервер, який буде їх обробляти, та відображати отриманий з серверу результат.

Існуючі методи пошуку обличчя на фото. (1/2)

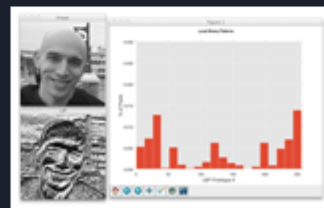
Класифікатор Хаара

Полягає у тому, що на зображенні виділяються деякі ознаки, за якими у більшості випадків вдається виявити обличчя. Ці ознаки засновані на контрасті між певними частинами обличчя.



ЛБШ класифікатор

Класифікатор текстур, який виявляє мікроскопічні візуальні шаблони у зображенні та зберігає їх у векторі ознак.



Існуючі методи пошуку обличчя на фото. (2/2)

Алгоритм SIFT

Виявлення і опис локальних особливостей у зображеннях. Заснований на знаходженні ключових точок на еталоні та їх порівнянням із сценою.



Алгоритм SURF

Має ті ж етапи, що і SIFT, проте деталі на кожному кроці різні.



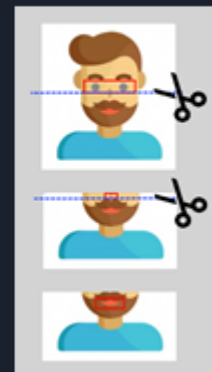
Метод Віопи-Джонса



- Швидкість;
- Навіть здатен працювати у режимі реального часу;
- Мало невірних спрацьовувань за умови правильно натренованого каскаду;
- Аналізує зображення на відповідність деяким ознакам, що властиві шуканому класу об'єктів, на основі чого приймається рішення щодо належності чи неналежності розглянутої частини зображення до шуканого класу об'єктів;

Розпізнавання певних частин обличчя - виділення зон інтересу

Виділення зон інтересу, - тобто таких, де абсолютно точно повинен бути шуканий об'єкт, а поява у цій зоні іншого об'єкту, що схожий на шуканий, малоймовірна, - допоможе пришвидшити пошук.



Розпізнавання певних частин обличчя - обчислення коефіцієнту відстані

Після розпізнавання місцезнаходжень очей, носу та роту, слід порівняти коефіцієнти цих відстаней для кожного з поворотів обличчя.

$$M_{rect} = rect.y + rect.width;$$

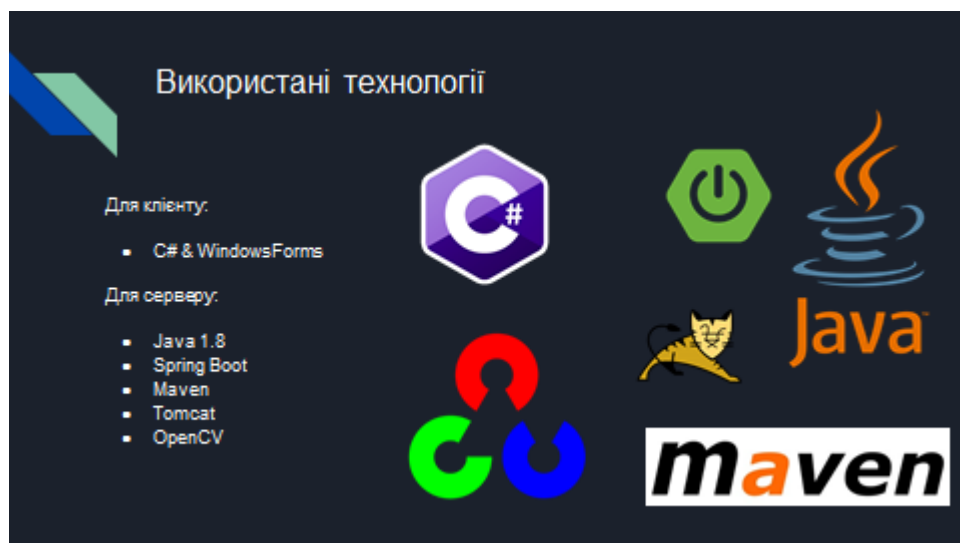
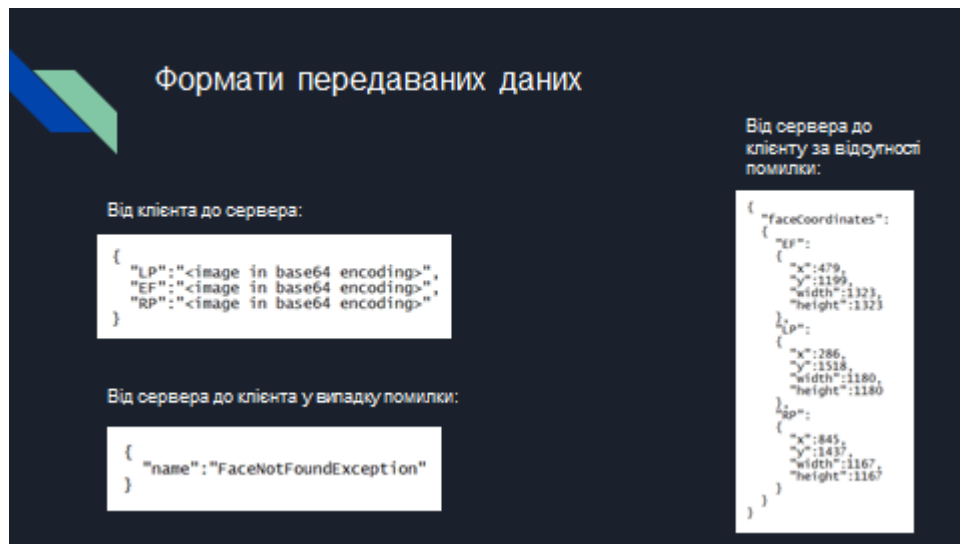
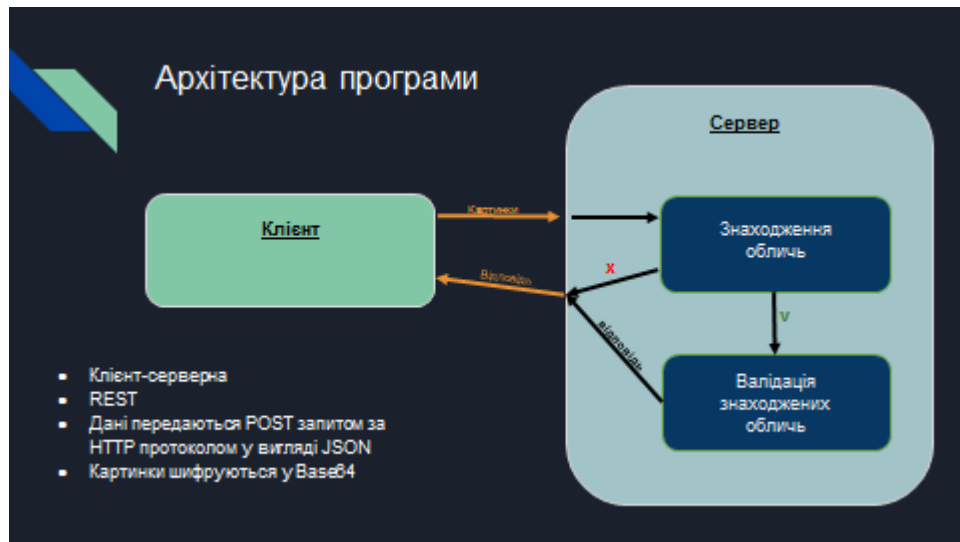
$$h_i(x, y) = M_y - M_x;$$

$$k_i = \frac{h_i(eye, nose)}{h_i(nose, mouth)}$$

$$\Delta_L = k_{LP} - k_{LP};$$

$$\Delta_R = k_{RP} - k_{RP};$$

Якщо будь-яка із знайдених дельт більше, ніж задана припустима похибка, приймається рішення про те, що обличчя належать різним людям.



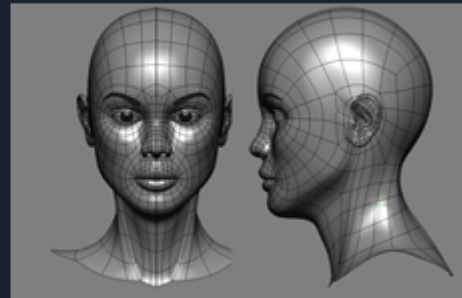
Відеодемонстрація роботи програми



Висновки та перспективи роботи

У результаті виконання атестаційної роботи було розроблено програмне забезпечення, що, по-перше, шукає обличчя на кожній фотографії з серії знімків людини у різних ракурсах, а, по-друге,

У подальшому планується покращення цього програмного застосування до такого, що буде здатне робити 3D-модель людини за серією знімків.



Дякую за увагу :)