МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського

«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп’ютерних систем та інфокомунікацій

Кафедра комп’ютерних систем, мереж і кібербезпеки

**Курсова робота**

|  |  |
| --- | --- |
| на тему | «Додаток для розпізнання обличчя» |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Виконав: студент | | | *4* | курсу групи № | | | | *545А* |
| напряму підготовки (спеціальності) | | | | | | | | |
| *123 – комп’ютерна інженерія* | | | | | | | | |
| (шифр і назва напряму підготовки (спеціальності)) | | | | | | | | |
| *Побережнік Микита Сергійович* | | | | | | | | |
| (прізвище й ініціали студента) | | | | | | | | |
| Керівник: | *Асистент* | | | | | | | |
| *Землянко Григорій Андрійович* | | | | | | | | |
| (посада, науковий ступінь, прізвище й ініціали) | | | | | | | | |
| Національна шкала: | | | | |  | | | |
| Кількість балів: | | | | |  | | | |
| Оцінка ECTS: | | | | |  | | | |
| Члени комісії: | |  | | | |  |  | |
|  | | (підпис) | | | |  | (прізвище й ініціали) | |
|  | |  | | | |  |  | |
|  | | (підпис) | | | |  | (прізвище й ініціали) | |
|  | |  | | | |  |  | |
|  | | (підпис) | | | |  | (прізвище й ініціали) | |

Харків – 2021

ЗМІСТ

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 3](#_Toc90222017)

[ВСТУП 5](#_Toc90222018)

[1 ПОСТАНОВКА ЗАДАЧІ 6](#_Toc90222019)

[1.1 Актуальність завдання розпізнавання обличчя 6](#_Toc90222020)

[1.2 Аналіз існуючих способів розпізнавання обличчя 9](#_Toc90222021)

[1.3 Аналіз існуючих бібліотек для розпізнавання обличчя 17](#_Toc90222022)

[1.4 Постановка задачі 25](#_Toc90222023)

[Висновок 26](#_Toc90222024)

[2 ПРОЕКТУВАННЯ 27](#_Toc90222025)

[2.1. Аналіз мови програмування 27](#_Toc90222028)

[2.2. Середа розробки 30](#_Toc90222029)

[2.3. Алгоритми 32](#_Toc90222030)

[2.4. UML-діаграми 38](#_Toc90222031)

[2.5. Архітектура додатку 40](#_Toc90222032)

[2.6. Архітектура сіамської нейронної мережи 42](#_Toc90222033)

[Висновок 45](#_Toc90222035)

[ПЕРЕЛІК ДЖЕРЕЛ 46](#_Toc90222036)

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

|  |  |
| --- | --- |
| ПЗ | * Програмне забезпечення |
| IoT | * Інтернет речі (Internet of Things) |
| CAGR | * Сукупний середньорічний темп зростання (Compound annual growth rate) |
| FRR, FNMR | * Хибно негативний результат (False Rejection Rate або False Non-Match Rate) |
| FAR, FMR | * Хибно позитивний результат (False Acceptance Rate або False Match Rate) |
| FR | * Кількість хибних не розпізнавань (False Reject) |
| FA | * Кількість хибних розпізнань (False Acceptation) |
| FRVT | * Тест постачальника на розпізнання обличчя (Face Recognition Vendor Test) |
| N/A | * Не доступний (Not available) |
| БД | * База даних |
| 2D | * Двомірний простір (2-dimensional) |
| 3D | * Тримірний простір (3-dimensional) |
| PCA | * Метод головних компонентів (Principal component analysis) |
| API | * Програмний інтерфейс додатку (Application Programming Interface) |
| МП | * Мова програмування |
| ГП | * Графічний процесор |
| ЦП | * Центральний процесор |
| HOG | * Гістограма спрямованих градієнтів (Histogram Of Oriented Gradients) |
| CNN | * Згорткова нейронна мережа (Convolutional Neural Network) |
| SNN | * Siamese neural network |
| MVC | * Model View Controller |
| MTV | * Model Template View |
| CUDA | * Compute Unified Device Architecture |
| ШІ | * Штучний інтелект |
| IDE | * Інтегрована середа розробки (Integrated development environment) |
| CRUD | * Create, read, update, delete |
| HTML | * Мова гіпертекстової розмітки (HyperText Markup Language) |
| DTL | * Мова шаблонів Django(Django Template Language) |

# ВСТУП

Нейронні мережі розвиваються ще з минулого століття, проте розкрити свій потенціал та можливості змогли лише за часів поширення інформаційних технологій та високої швидкодії машин на яких вони виконуються. Сфера застосування нейронних мереж різноманітна: від простої вибірки та класифікації даних до складних обчислень та прийняття рішень за людину в типових ситуаціях, наприклад, у визначенні неполадок у критичному обладнанні чи системах у цілому – все це має потенціал у розвитку нейронних систем. Вони беруть за основу і подобу реально існуючу систему - людський мозок, що складається з безлічі структурних одиниць, а саме нейронів. Мозок дозволяє використовувати нейрони для певних цілей і робить це набагато швидше обчислювальних машин, оскільки він має властивість довгострокового збереження станів сигналів, що розповсюджуються по цій самій нейронній мережі. Цю проблему і намагаються вирішити нейронні мережі шляхом найчастіше процесу навчання.

На сьогодні нейронні мережі представляють лише спрощений варіант роботи мозку, проте вони вже використовуються в багатьох галузях: кібернетика, штучний інтелект, програмування, математика та машинне навчання – всі ці фундаментальні галузі науки мають власну проблематику щодо використання нейромереж та вирішення конкретних завдань. У цій роботі буде розглянуто їх застосування для розпізнавання обличчя з використанням вже навчених реалізацій.

Ще у 2012 році майбутнє нейронних мереж описували як підставу для створення повністю автономного людиноподібного мозку[1]:

«*…The future of neural networks is bright and current research seems to be moving in the right direction towards the ultimate goal of all artificial intelligence, namely, the development of a humanoid robot that can work and think like a human being…*»

Але насьогодні ми можемо побачити, що їх використання йде більше убік автоматизації систем де безпосередньо необхідне прийняття рішень людини або для її ідентифікації.

Серед найбільш поширених реалізацій із використанням описаних мереж можна вважати задачу розпізнавання обличчя. Актуальність даної роботи можна простежити у збільшенні попиту щодо цієї позиції: все більше техніки, систем та пристроїв передбачають у собі вбудоване програмне забезпечення (ПЗ) оснащене нейронною мережею, яка виконує обчислення на основі вхідних даних з датчиків і видає найбільш сприятливий результат. Наприклад, системи відеоспостереження на державних й приватних об’єктах чи у містах в цілому зчаста встановлюються вже у промислових масштабах.

Таким чином головною метою дипломного проекту є аналіз існуючих реалізації нейронних мереж для розпізнавання обличчя, їх порівняння та висновок щодо критеріїв швидкодії, точності виконання та кінцевого результату.

# ПОСТАНОВКА ЗАДАЧІ

## Актуальність завдання розпізнавання обличчя

Розпізнавання обличчя – це один з найбільш перспективних методів біометричної безконтактної ідентифікації людини. У наш час технологія розпізнавання обличчя найчастіше використовується в системах відеоспостереження, контролю доступу, на різноманітних мобільних та хмарних платформах. Ці методи зазвичай складаються із нейронної мережі, яку навмисно «навчають» на деяких тестових даних, саме як людина упродовж життя знайомиться, наприклад, із іншими людьми та запам’ятовує, відразу чи пізніше, їх обличчя, характерні їм звички та інше, – аналогічно навчається й нейронна мережа, що пізніше буде використовуватися для розпізнавання обличчя за деякою базою даних (БД) із усіма обличчями до порівняння. Їх специфіка застосування різниться залежно від критичності до помилок у сфері якій вони використовуються.

Останні досягнення в аналізі обличчя, розпізнаванні образів і машинному навчанні зробили можливим розробити автоматичні системи для вирішення цих задач. З одного боку, розпізнавання обличчя є природним процесом, тому що люди зазвичай роблять це легко, без особливої ​​свідомості. З іншого боку, застосування цього процесу в області комп'ютерного зору залишається складною проблемою. Будучи частиною біометричної технології, автоматичне розпізнавання обличчя має безліч бажаних властивостей. Вони засновані на важливій перевазі — неінвазивності, тобто характеристиці пристрою, що не контактує із тілом яким-небудь прямим способом. Різні біометричні методи можна розділити на фізіологічні (відбитки пальців, ДНК, обличчя) і поведінкові (натискання клавіш, відбиток голосу). Фізіологічні підходи більш стабільні та незмінні, за винятком серйозних травм. Але слід відзначити, що зазвичай людині тяжко ідентифікувати іншу особу за шаблоном (фото з паспорту, або інше джерело), а її обсяг пам’яті невеликий, коли для системи це набагато простіший процес, якщо остання навчалась правильно та на достатньої кількості даних.

Згідно із звітом ресурсу MarketsandMarkets[2], очікується, що обсяг світового ринку розпізнавання обличчя на 2020 рік з 3,8 мільярда доларів зросте до 8,5 мільярдів до 2025 року. Головними галузями та трендами, які є найбільш значущими у зростанні популярності, відзначають зростання значення індустрії спостереження, збільшення інвестицій у технології розпізнавання обличчя з боку державного та оборонного секторів та впровадження хмарної технології розпізнавання обличчя. Однією з причин зростання інтересу та необхідності в нейромережах став також коронавірус: враховуючи розмах і актуальність проблеми пандемії, подібні технології стають невід'ємною частиною систем безпеки, де насамперед важлива соціальна дистанція та ідентифікація особи. Порівняно з біометричними підтвердженням особистості, авторизацією в медичних сервісах та іншому, розпізнавання обличчя стане чудовою заміною, оскільки дозволять виключити проблему сканерів, які потребують фізичної взаємодії, як, наприклад, сканери відбитків пальців.

Уряди по всьому світу вкладають значні кошти у впровадження цих технологій. Головним прикладом тут можна вважати США та Китай. Китай є одним з найбільшим виробником та експортером систем пов'язаних з розпізнаванням обличчя, і в самому Китаї наразі діє близько 200 мільйонів камер спеціально створених для цього, а до наступного року їх кількість має зрости втричі – до 626 мільйонів. Крім проблем охорони здоров'я та збереження соціальної дистанції, основною проблемою, яку вирішують подібні камери, є мінімізація злочинності, однак варто враховувати, що рішення, які дають ці системи, не є цілком успішними. Наприклад, посилаючись на ту саму статтю, можна побачити, що похибка у вирішенні проблеми ідентифікації людини по обличчю ще досить висока[2]:

«*…Throughout the testing period, the systems correctly identified the volunteers 153 times and failed to identify the volunteers 96 times. The high number of false alarms can result in the arrest of innocent people that do not have any criminal records against them. The arrest of innocent individuals could further lead to them taking legal action against the concerned authorities. Hence, this also remains a major restraint in the implementation of facial recognition technology…*»

І в той же час трапляються випадки, коли використання систем розпізнавання осіб вважають незаконними на законодавчому рівні, коли йдеться розмова про збереження приватних даних та про вторгнення в особисте життя людей, що може порушувати Європейську конвенцію з прав людини[3].

Іншими цікавими областями застосування системи розпізнавання обличчя можуть стати медицина та освіта. За допомогою комп'ютерного аналізу медики зможуть відстежувати стан пацієнта, оцінювати перебіг лікування, виявляти ознаки хвороби тощо. В освітній сфері, яка все більше переходить в онлайн, системи допоможуть аналізувати поведінку учнів, сприяти більшій залученості у процесі навчання. Крім того, міські камери, інтегровані в систему Internet of Things (IoT), здатні шукати дітей, старих або хворих людей, що втратилися без допомоги.

Звичайно, існує, і досі використовується, безліч інших способів ідентифікації: за голосом, через відбиток пальця або сканування райдужної оболонки ока. Але у біометрії по обличчю є низка переваг: вона легко впроваджується, дає швидкий результат та працює дистанційно, що особливо актуально під час пандемії.

Для порівняння із ресурсом MarketsandMarkets, я відзначив для себе звіт ресурсу Technavio[4], де дані були зібрані із первинних та вторинних джерел: галузеве інтерв'ю з учасниками ринку та через річні звіти, прес-релізи, презентації компаній відповідно. В цьому звіті також відзначається великий запит на збільшення використання технології нейронних мереж для розпізнавання обличчя, та згадуються такі країни як США та Китай в ролі лідируючих у цій галузі. Важливо також зауважити, що головним фактором щодо перешкод для розповсюдження цих технологій є зростання у вартості розгортання подібних систем[4]:

«*…One of the key factors impeding the facial recognition market growth is the high cost of deployment…*»

На рисунку 1.1 зображено ріст об’єму ринку розпізнавання обличчя за ресурсом Technavio.

Рисунок 1.1 – Об'єм ринку розпізнавання обличчя

Як можна помітити із даних ресурсів MarketsandMarkets та Technavio, їх значення сильно відрізняються, але тенденція залишається незмінною: ринок технології розпізнавання обличчя з роками буде рости та розширюватися.

Тобто незважаючи на те, що в деяких країнах ставляться негативно до використання нейронних мереж для розпізнавання обличчя із метою пошуку злочинців та в цілому автоматизації процесів верифікації особистості людини [3] можна сказати, що попит на використання технології пов'язаних з розпізнаванням обличчя на основі нейромереж буде тільки зростати і мати всі підстави для розвитку, оскільки найбільш впливовіші країни, як США та Китай, витрачають на це достатньо велику кількість кошт, а сама технологія передбачає у собі всі передумови для охоплення надто великої кількості сфер для користування.

## Аналіз існуючих способів розпізнавання обличчя

Серед способів розпізнавання обличчя слід відзначити насамперед головні проблеми, які вони мають вирішувати, щоб структуровано описати ці методи. Головними підрозділами щодо розпізнавання обличчя є проблеми двомірного паттерну (2D) та тримірного (3D)[5]. Відповідно до них будуємо слідуючи вимоги[5]:

1. Загальна швидкість системи від виявлення до визнання має бути приємлимою;
2. Точність повинна бути високою (>95%);
3. Система повинна легко оновлюватися і розширюватися, тобто легко збільшувати кількість об’єктів, які можуть бути визначені.

Також слід зазначити помилки якими керуються при порівняні алгоритмів розпізнання обличчя. Існує два види помилок помилки першого роду та другого, FAR та FFR відповідно.

Помилки першого роду – це хибно негативний результат, тобто хибне прийняття різних людей за одну й ту саму людину. В таблицях відзначають цей параметр або як False Acceptable Rate (FAR), або як False Match Rate (FMR). Розраховуються наступним чином:

(1.1)

Де FA – кількість хибних розпізнань (Іванов розпізнаний як Петров);

– кількість еталонних зображень в базі даних.

Помилки другого роду – це хибно позитивний результат, що навпаки не розпізнає одну людину на різних фото, вважаючи, що це інша людина. В таблицях відзначають цей параметр як False Rejection Rate (FRR) або як False Not-Match Rate (FNMR). Розраховуються таким чином:

(1.2)

Де FR – кількість хибних не розпізнань (Іванов не розпізнаний як Іванов);

– кількість еталонних зображень в базі даних.

Обидві помилки є відносними величинами, тобто залежать від алгоритму, де вони використовуються, а також вони взаємозв’язані: «…Чим менше FAR тим більше FRR[6]», приблизний зв'язок описано в таблиці 1.1, а більш детальні приклади, де вони застосовуються, у таблиці 1.2.

Таблиця 1.1 – Приблизний взаємозв’язок між FAR та FRR[6].

|  |  |
| --- | --- |
| FAR | FRR |
| 0.1% | 2.5% |
| 0.01% | 7% |
| 0.001% | 10% |

Таблиця 1.2 – Порівняння FAR та FRR різними методами біометричної ідентифікації [6].

|  |  |  |
| --- | --- | --- |
| Метод біометричної ідентифікації | FAR | FRR |
| Відбиток пальця | 0.001% | 0.6% |
| Розпізнавання обличчя 2D | 0.1% | 2.5% |
| Розпізнавання обличчя 3D | 0.0005% | 0.1% |
| Райдужна оболонка очей | 0.00001% | 0.016% |
| Сітківка ока | 0.0001% | 0.4% |
| Малюнок вен | 0.0008% | 0.01% |

Важливо розуміти, що алгоритмів щодо реалізації нейромереж для розпізнавання обличчя безліч, є велика кількість ресурсів щодо порівняння останніх за параметрами швидкодії та точності, але будуть відокремлені найбільш популярні й великі з них: NIST (National Institute of Standards and Technology) [7], MegaFace [8] та LFW (Labeled Faces in Wild) [9] – усі вони мають власну статистику щодо тестів алгоритмів розпізнавання обличчя на предмет відсотків помилок та інших характеристик. Так, наприклад, ресурс FGLFW(Fine-grained LFW)[10] відзначає велику різницю у точності верифікації (приблизно 10-20%) при застосуванні бази даних, що має більш «ідеальні» зображення та бази даних, коли використовуються зображення із різними кутами знімку, розширенням та цілком якістю. Далі приведена таблиця 1.3 порівняння алгоритмів розпізнання зображення на різних базах даних зображень.

Таблиця 1.3 – Точність верифікації (%) для набору даних LFW і FGLFW для різних алгоритмів розпізнання обличчя

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Кількість зображень (млн.) | LFW | FGLFW |
| DeepFace | 0,5 | 92,87% | 78,78% |
| DeepID2 | 0,2 | 95,00% | 78,25% |
| VGG-Face | 2,6 | 96,70% | 85,78% |
| DCMN | 0,5 | 98,03% | 91,00% |
| Noisy Softmax | 0,5 | 99,18% | 94,50% |
| Людина | n/a | 99,85% | 92,00% |

Як бачимо із таблиці, оскільки в БД LWF зображення є більш привабливими для алгоритму, та не стільки різняться між собою, як у FGLFW, то й відсоток верифікації у використані перших вище. Також відмічається, що людина при розпізнані подібних зображень мала достатньо високий відсоток, оскільки в якості шаблонів для зображень були обрані популярні та медійні люди, яких частіш за все можуть впізнати в обличчя.

На початку 1970-х років розпізнавання обличчя розглядалося як проблема розпізнавання 2D (двомірних) образів [11]. В основі технології 2D розпізнавання обличчя лежать плоскі двовимірні зображення. Алгоритми розпізнавання осіб використовують антропометричні параметри, графи, моделі обличчя або еластичні 2D-моделі, а також зображення з особами, представлені деяким набором фізичних чи математичних ознак, наприклад, як на рисунку 1.2.

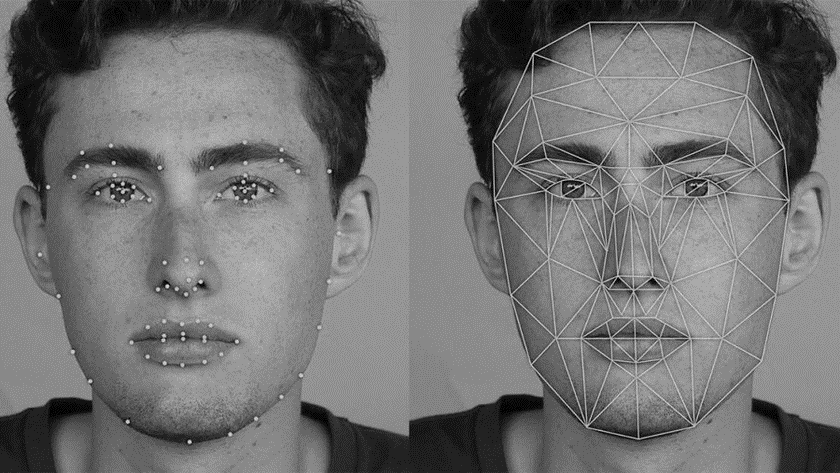


Рисунок 1.2 – Приклад 2D зображення обличчя із побудованими точками

Розпізнавання 2D зображень є однією з найбільш поширених технологій на даний момент. Оскільки основні бази даних ідентифікованих людей, накопичені у світі, - саме двомірні. І основне обладнання, вже встановлене у всьому світі, також 2D за даними на 2016 рік[12], а точніше – біля 350 мільйонів камер відеоспостереження.

Головною перевагою 2D-паттерну є наявність готових баз даних обличь, шаблонів та готової інфраструктури. Максимальний попит доведеться саме на цей сегмент, а він стимулюватиме розробників удосконалювати технології. Серед недоліків можливо відзначити вищі коефіцієнти помилок FAR та FRR порівняно з 3D розпізнаванням осіб.

Для розпізнавання обличчя використовуються частіш за все наступні групи методів:

1. Цілісні методи порівняння (Holistic Matching Methods) – вся область обличчя враховується як вхідні дані в системі фіксації обличчя. Перша успішна демонстрація машинного розпізнавання обличчя була зроблена у 1991 році за допомогою методу Eigenfaces (Власне обличчя) [13]. Спочатку створюється база даних для зображень обличчя, що потім використовуватимуся для порівняння та створення «Власного обличчя», вони створюються за допомогою вилучення характерних рис обличчя (ніс, рот, очі та інше). Потім ці вхідні дані нормалізуються, тобто вирівнюються відносно осей координат, а також змінюється розширення, щоб у зображення для порівняння та вхідного воно було однакове й ми змогли витягти дані зображення за допомогою математичного методу головних компонентів (PCA). Після обробки даних зображень методом PCA, ми отримуємо вектор ваги будемо порівнювати між собою. Пошук відбувається шляхом знаходження найближчого по вагам зображення у БД. Рисунок 1.3 показує блок-схему застосування методу;
2. Функціональні (структурні) методи (Feature-Based Methods) – У цьому методі насамперед характерні риси обличчя, такі як очі, ніс і рот видобуті та їх розташування й положення (геометричне, відносно осей координат) подаються у вигляді структурного класифікатора. Великою проблемою для методів вилучення ознак є функція "відновлення", це коли система намагається отримати ознаки які «невидимі» через великі розмежування між у зображеннях, наприклад, поза голови, коли ми зіставляємо фронтальне зображення із зображенням профілю [14]. Тому розрізняють три різних методи вилучення:
   1. Загальні методи, засновані на ребрах, лініях і кривих;
   2. Методи на основі функцій-шаблону;
   3. Методи структурної відповідності, які враховують геометричні обмеження на особливості.
3. Гібридні методи (Hybrid Methods) – використовують як поєднання цілісних та функціональних методів, тобто методів виділення ознак. Як правило, використовуються при виявленні 3D зображень.

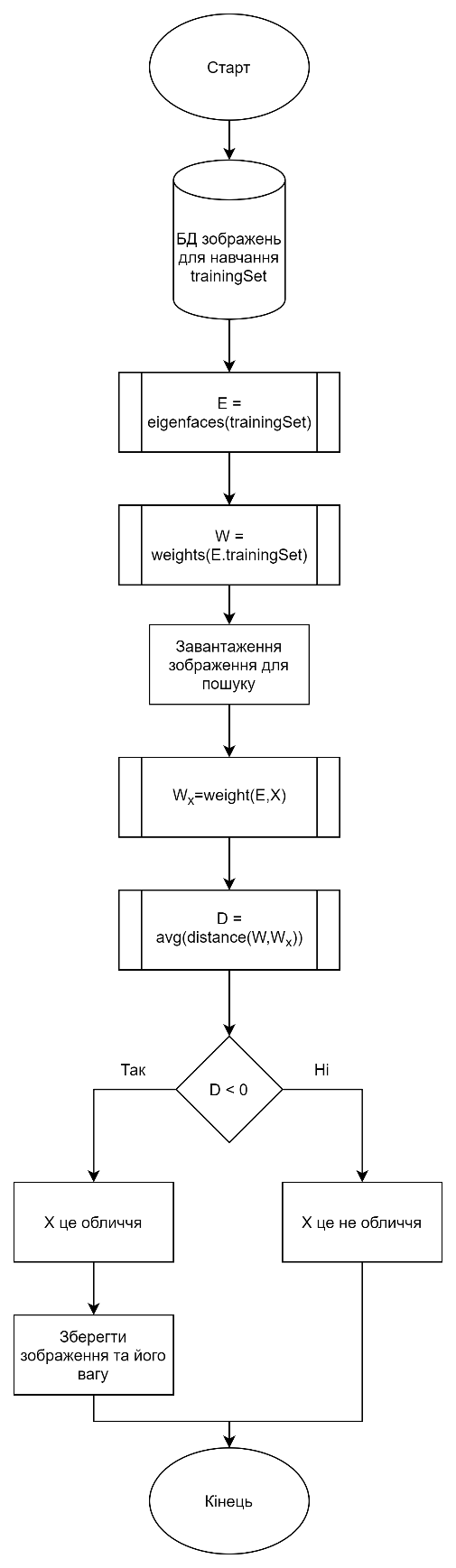


Рисунок 1.3 – Блок-схема застосування методу Eigenfaces

3D розпізнавання виробляється зазвичай по реконструйованим тривимірним образам, тобто на вершинах, на відміну від векторів точок у 2D. Технологія 3D розпізнавання обличчя має більш високі якісні характеристики. Хоча, звісно, і вона не є ідеальною.

Існує кілька різноманітних технологій 3D сканування: лазерні сканери з оцінкою дальності від сканера до елементів поверхні об'єкта, спеціальні сканери зі структурованою підсвічуванням поверхні об'єкта та математичною обробкою згинів смуг, або це можуть бути сканери, що обробляють фотограмметричним методом синхронні стереопари зображень осіб – всі вони досі достатньо складні у попиті та коштують набагато більше 2D родичів.

Серед методів для яких використовується 3D розпізнавання обличчя відзначаємо вже описаний вище гібридний метод – зображення обличчя отримується у вигляді 3D-координат, дозволяючи системі більш детально охоплювати вигини очних западин або форми підборіддя, лобу та іншого. Навіть обличчя шаблон в якості 2D-зображення може стати 3D при перетворені за допомогою спеціальних технології[15]. Приклад 3D обробки обличчя зображено на рисунку 1.4.

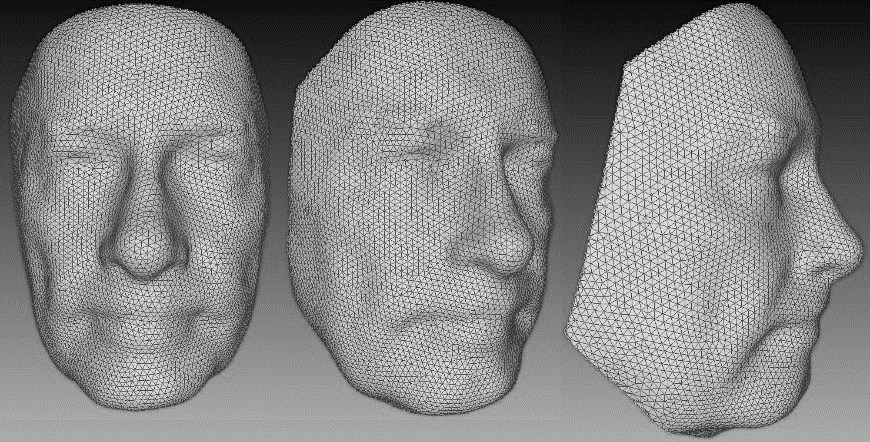


Рисунок 1.4 – Приклад побудованого 3D зображення у результаті розпізнавання обличчя

Зазвичай 3D розпізнавання працює наступним чином: виявлення, позиція, вимірювання, представлення та відповідність. Опишемо ці етапи більш детально:

1. Виявлення – зйомка обличчя, сканування фотографії або фотографування обличчя людини в режимі реального часу;
2. Позиція – визначення розташування, розміру та кута нахилу голови;
3. Вимірювання – призначення вимірювань для кожної кривої обличчя, щоб створити шаблон із конкретним фокусом на зовнішній та внутрішній сторонах ока та куті носа;
4. Представлення - перетворення шаблону в код - числове представлення обличчя;
5. Збіг – порівняння отриманих даних з базою даних.

Якщо 3D-зображення потрібно порівнювати із існуючим 3D-зображенням, потрібно, щоб у них не було ніяких кореляцій між собою. Однак зазвичай фотографії порівнюють у 2D, тому з’являється причина перетворення 3D зображень у близьку до двомірного, тобто тривимірне зображення потребує кількох змін. Це складно і є однією з найбільших проблем у цій галузі сьогодні.

Серед переваг 3D розпізнавання обличчя відзначаємо велику точність і меншу кількість помилок у порівнянні із 2D системами. Для порівняння приведемо таблицю 1.3 щодо значень FAR та FRR[6].

Таблиця 1.3– Порівняння 3D та 2D за показниками помилок FAR та FRR

|  |  |  |
| --- | --- | --- |
| Метод ідентифікації | Коефіцієнт хибних пропусків, FAR | Коефіцієнт хибних відказів, FRR |
| 2D | 0.0005% | 0.1% |
| 3D | 0.1% | 2.5% |

Але є також й причини, чому досі 3D технологія не обігнала 2D та не заваблює до себе: не дивлячись на те, що точність при ідентифікації обличчя вище, аніж у 2D, зробити підробку також досить легко й для 3D систем. Так, наприклад, один із найпопулярніших додатків власників iPhone – FaceID – достатньо легко обійти зробивши 3D маску обличчя, зі слів статті[16]:

«*…In this new experiment, Bkav used a 3D mask (which costs ~200 USD), made of stone powder, with glued 2D images of the eyes. Bkav experts found out that stone powder can replace paper tape (used in previous mask) to trick Face ID' AI at higher scores. The eyes are printed infrared images – the same technology that Face ID itself uses to detect facial image. These materials and tools are casual for anyone. An iPhone X has its highest security options enabled, then has the owner's face enrolled to set up Face ID, then is immediately put in front of the mask, iPhone X is unlocked immediately. There is absolutely no learning of Face ID with the new mask in this experiment…*»

Також однією із причин є те, що 3D технології потребують спеціального обладнання, коли увесь світ використовую 2D, тому й баз даних для цієї задачі також недостатньо, усе це коштує занадто багато навіть для державного сектору. Однак, це не означає, що з часом технології рухатимуться вперед та 3D не набуде попиту, тому що навіть у офіційному стандарті щодо біометричної ідентифікації[17] вже містить достатню кількість згадок про це.

Зображення з високою роздільною здатністю ще один фактор у вдосконаленні технології розпізнавання обличчя, завдяки високому дозволу став можливий дуже складний аналіз текстури шкіри. При такому аналізі певна область шкіри обличчя може бути захоплена як зображення, а потім розбита на більш дрібні блоки, які перетворюються на математичні вимірні простори, в яких записуються лінії, пори і фактична текстура шкіри. Технологія може ідентифікувати різницю між близнюками, що поки що неможливо використовувати за допомогою програмного забезпечення із вищеописаними 2D та 3D технологіями у побуті для розпізнавання осіб. У разі поєднання розпізнавання особи з аналізом поверхневої текстури, точність ідентифікації може сильно збільшитися, але ці технології коштують ще досить забагато, а тому не знайшли свого призначення, найбільш перспективним вважається об’єднання їх із іншими способами розпізнавання обличчя.

Використання тепловізійних камер для цілей розпізнавання осіб на даний момент вважається перспективним напрямком для розробки, оскільки надає також високих значень точності та ідентифікації, також воно є допоміжним у ситуації, наприклад, із пандемією, як наприклад на рисунку 1.5, коли необхідне стеження за станом великої кількості людей, але готових для впровадження комерційних рішень поки немає.

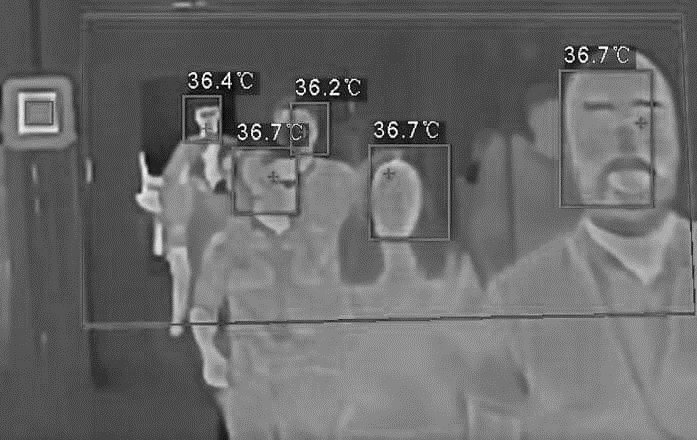


Рисунок 1.5 – Приклад розпізнавання обличчя та виміру температури тіла водночас

Технологія досить перспективна, оскільки дозволяє нівелювати больові точки 2D-розпізнавання:

1. Розпізнавання осіб у повній темряві та в умовах недостатнього освітлення;
2. Макіяж, зачіска, борода, капелюх, окуляри — не є проблемою для тепловізійних камер;
3. Дозволяють розпізнавати близнюків.

Оскільки технологія не є ще вдосконаленою, то головна розробка, опираючись на ресурс [18], ведеться у два напрямки:

1. Ідентифікація за заздалегідь створеними термограмами ідентифікованих обличь. Тут проблеми ті ж, що і з 3D-розпізнавання, готових баз даних шаблонів немає та дороге обладнання;
2. Ідентифікація людини за зображеннями отриманими з тепловізійної камери, а як обличчя шаблон застосовуються бази даних стандартних двовимірних зображень. Вирішується це завдання із використанням глибоких нейронних мереж о яких буде йти речі і далі.

## Аналіз існуючих бібліотек для розпізнавання обличчя

Коли ми говоримо о технологіях розпізнання обличчя, то частіш за все маємо на увазі нейронні мережі, які згадувалися раніше. Але що ж саме таке – нейронні мережі?

Нейронні мережі працюють за прикладом з нейронами живої людини: існує нейрон, який має ядро, у нього по дендритах передається вхідний сигнал (Це є з'єднанням нейронів, тобто синапсом). Ядро, залежно від синапсу, заряджається або розряджається і на виході дає сигнал для аксона, який слідує до синопсису наступного нейрона. І так у циклі по всій системі. На рисунку 1.6 зображена більш детальна схема роботи цього алгоритму:

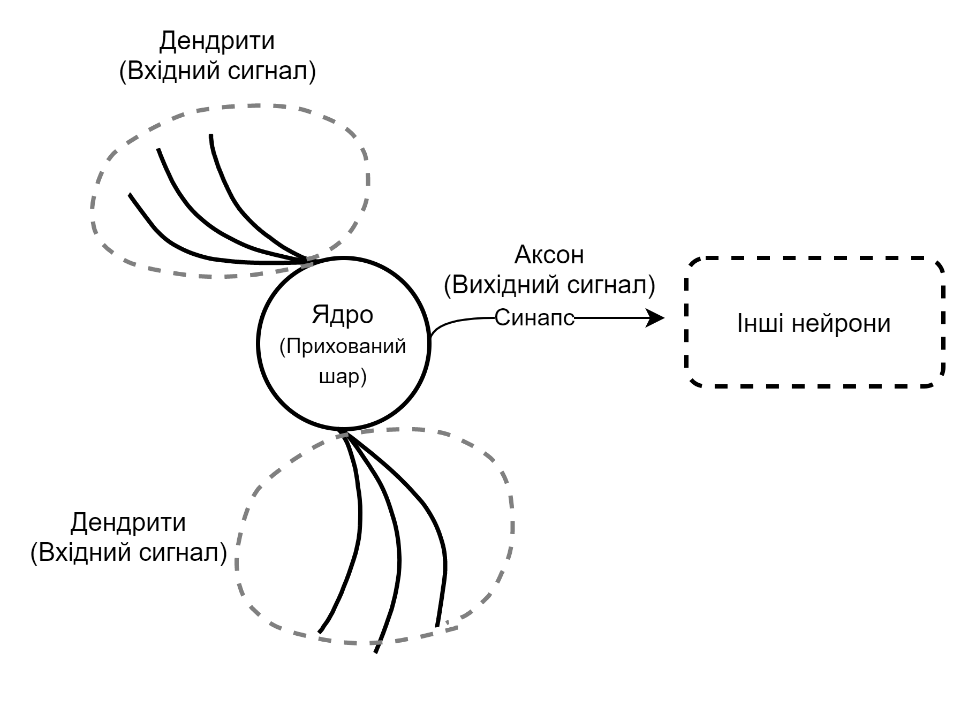


Рисунок 1.6 – Схематичне зображення роботи нейрону у нейронній мережі

Якщо уявляти це завдання у математичному вигляді та представити як блок-схему, то отримуємо рисунок 1.7.

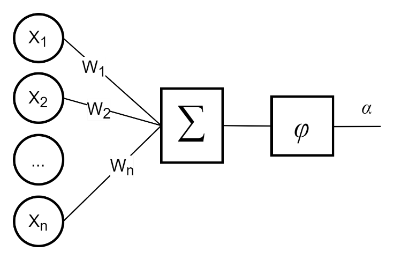


Рисунок 1.7 – Блок-схема штучного нейрона

Нелінійну функцію вирішуємо формулою 1.3.

(1.3)

Де – вхідний сигнал нейрона;

– вага вхідного сигналу;

– нелінійна функція активації, що визначає активований нейрон чи ні.

У роботі авторів Уолтера Піттса та Уоррена Мак-Каллока[19] були описані штучні нейроні та об’єднання їх у мережі в цілому, що стало штовхачем для створення такої науки як штучний інтелект взагалі. Вони пропонували представляти нейрону мережу так, як зображено на рисунку 1.8.

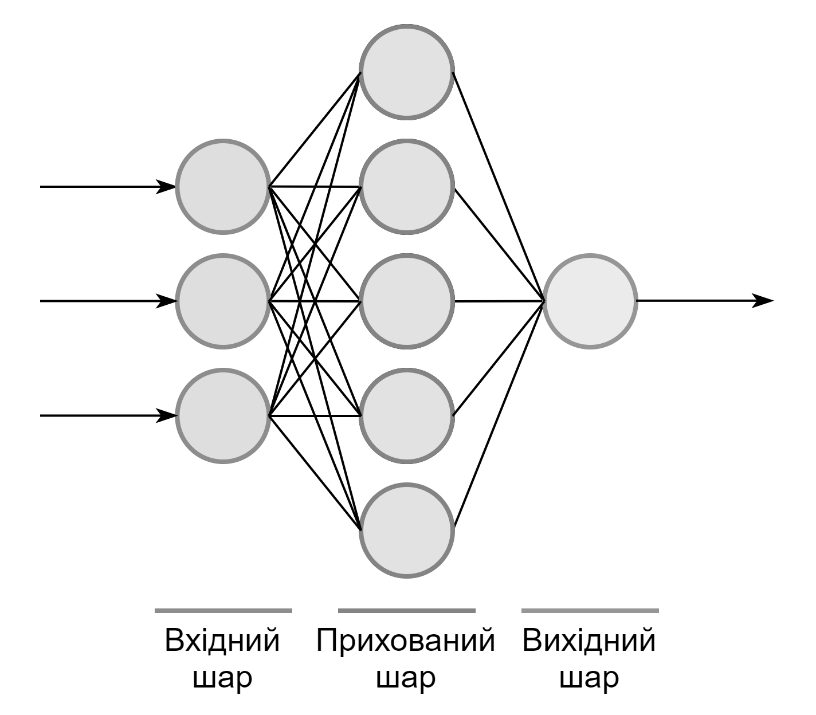


Рисунок 1.8 – Нейронна мережа, загальний вид

Принцип роботи цієї мережі полягає в тому, що ми маємо зовнішнє середовище від якого отримуємо вхідні сигнали на вхідний шар, оброблюємо їх у прихованому шарі (прихований від зовнішнього середовища), та відправляємо результуючий сигнал назад у зовнішнє середовище через вихідний шар. Серед різновиду нейронних мереж відзначають:

1. Мережу із прямим розповсюдженням сигналів – тобто сигнали вхідного шару та прихованого не змішуються між собою, а відразу об’єднуються у вихідному шарі. Представлена на рисунку 1.9;

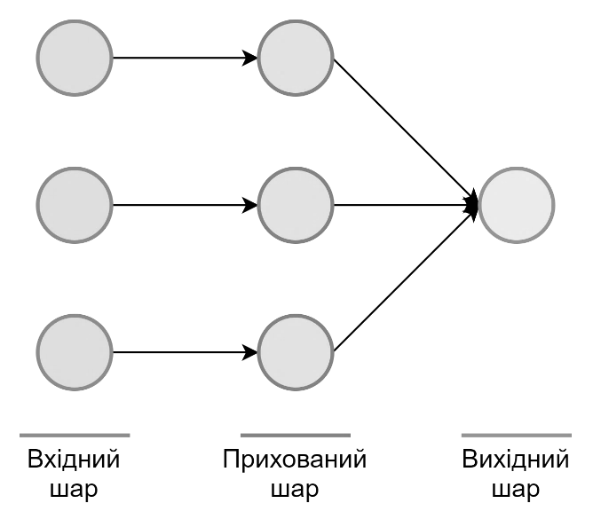


Рисунок 1.9 – Мережа із прямим розповсюдженням сигналів

1. Рекурентна мережа – мережа, що може бути зациклена сама у себе у прихованому шарі. Представлена на рисунку 1.10;

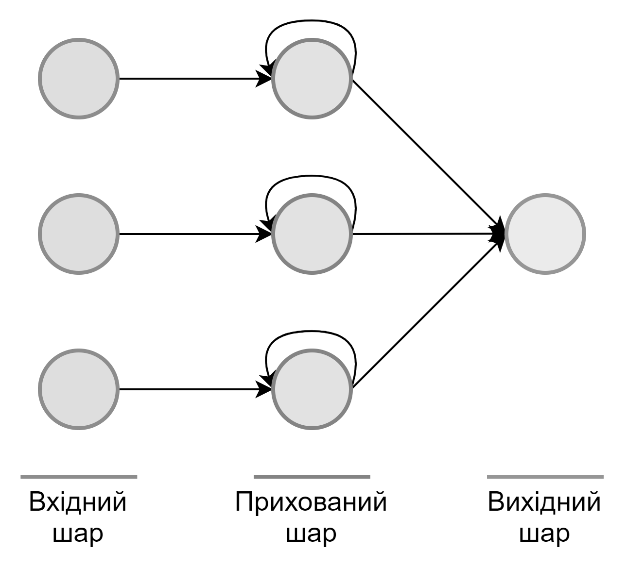


Рисунок 1.10 – Рекурентна мережа

1. Глибока нейронна мережа – нейронна мережа, що містить у собі 2 та більше прихованих шарів. Є найбільш розповсюдженою серед нейронних мереж. Представлена на рисунку 1.11.

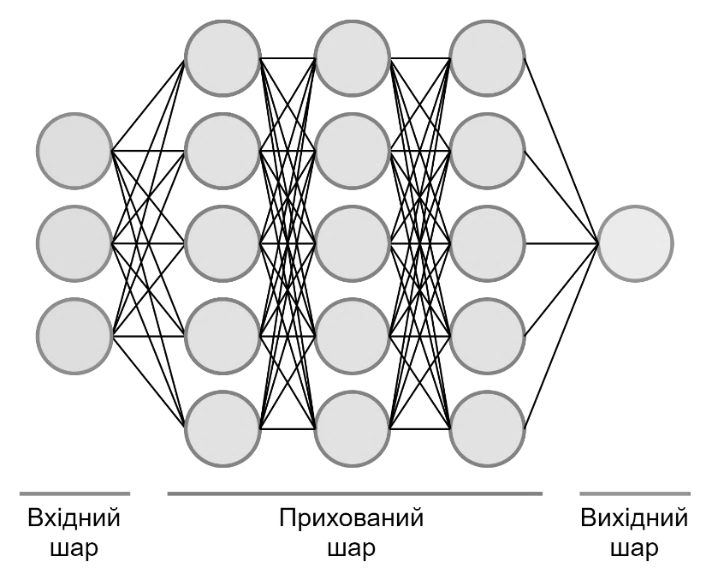


Рисунок 1.11 – Глибока нейрона мережа

Але навчанням мереж Піттс та Уоррен не займались. Вперше це запропонував Дональд Хебб у своїй роботі [19], де ввів свої правила щодо навчання нейронних мереж, засновуючись на тому, що якщо нейрони відпрацьовують разом – то їх зв'язок збільшуються, а якщо ні, то навпаки.

Головними завданнями, які вирішують нейронні мережі, є класифікація та регресія. Класифікація передбачає віднесення об’єкту до певного класу за його основними рисам, наприклад, відрізняти літак від стільця та відносити їх до різних класів. Регресія ж передбачає, наприклад, аналіз майбутніх значень якось події на основі поданих вхідних значень.

Взагалі усі нейронні мережі навчаються за наступними умовами навчання:

* + Із вчителем – передбачає, що вхідні дані поступають з заздалегідь відомими правильними відповідями;
  + Без вчителя – передбачає, що вхідні дані поступають без заздалегідь відомими правильними відповідями та має у собі завдання виявлення структурних відмінностей між даними, тобто завдання класифікації;
  + Навчання із підкріпленням – немає правильних відповідей, мережа виступає у ролі «Агенту» у зовнішньому середовищі та отримує сигнали також з цього середовища, що у ролі деякого вчителя підказує – правильно вирішує завдання мережа, чи ні.

Частіш за все використовується умова навчання із вчителем, оскільки вона дає найкращий результат навчання і більшість бібліотек орієнтується саме на цю умову.

Існує безліч бібліотек для навчання та використання нейронних мереж, для порівняння були обрані наступні:

1. Keras

Keras – це бібліотека, написана на мові програмування (МП) Python. Спочатку розроблювалася як Application Programming Interface (API), тобто інтерфейс, але потім зросла до окремої бібліотеки. Станом на 2018 рік, мала кількість користувачів у розмірі 250000 осіб [20]. Має можливість запускатися поверх таких платформ як TensorFlow, CNTK та Theano.

Переваги:

* Портативність;
* низький поріг входження для користувача;
* достатня кількість інструментів для візуалізації та моделювання результатів.

Недоліки:

* Низька швидкість розрахунків;
* Необхідність перед виконанням операції створення обчислювального графу;
* Неможливість працювати на низькому рівні, від чого виникає велика кількість помилок пов’язаних із цим.

1. Caffe

Caffe («Convolution Architecture For Feature Extraction» – Згорткова архітектура для отримання ознак) – це система глибокого навчання нейронних мереж, створена з урахуванням виразності, швидкості та модульності. Написана на МП С++, але має також API для Python. Підтримує велику кількість видів машинного навчання для вирішення задач класифікації та сегментації зображень. Головною відзнакою відмічають те, що вона використовує «блоби» – об’єктні файли, що завантажуються у ядро операційної системи (ОС) у вигляді багатомірних масивів даних та які можливо використовувати у паралельних обчисленнях.

Переваги:

* Швидкодія;
* Зручність використання;
* Відкритий код;
* Активно розвивається.

Недоліки:

* В залежності від задачі, не є достатньо гнучким та створений для спеціалізованих завдань;
* Якщо завдання виходить за рамки спеціалізованих для бібліотеки, то вона стає дуже тяжкою для використання;
* Не дивлячись на відкритість коду, немає достатньої кількості документації для ознайомлення.

1. Torch

Torch – це відкрита бібліотека для машинного навчання та наукових обчислень, вона також є мовою сценаріїв МП скриптів Lua. Реалізована на МП C. Не є активним у розробці з 2018 року через те, що розробка перейшла на до нової бібліотеки – PyTorch, тобто до мови Python. Головною різницею є те, що PyTorch забезпечує тензорні обчислення та підтримую прискорення із графічними процесорами (ГП).

Переваги:

* Зручність використання та налагодження;
* Велика кількість допоміжних бібліотек для розширення;
* Підтримка використання графічних процесорів.

Недоліки:

* Невелика кількість допоміжних матеріалів для навчання;
* Оскільки бібліотека PyTorch є оновленою Torch, то в ньому присутня невелика кількість спільноти;
* Прив’язана к однієї МП.

1. Dlib

Dlib – багатоплатформова, відкрита бібліотека загального призначення написана на мові C++. Дуже різноманітна бібліотека, що має велику кількість інструментарію: комп’ютерні мережі, потоки, графічні інтерфейси, структури даних, машинне навчання та інше. Бібліотека дуже гучна та зручна в використанні за рахунок того, що вона випускається з 2002-го року, постійно оновлюється и знаходиться в попиті: документація, та різні API, наявність досконалого налагодження та інструментарію для тестування робить її відмінною бібліотекою для навчання у сфері нейронних мереж та суміжних з нею, серед алгоритмів до останнього на офіційному сайті бібліотеки[21] відзначають:

* Структурний метод опорних векторів;
* Напіввизначена метрика навчання;
* Ядровий метод головних компонент;
* Многошаровий персептрон.

Також, як говориться на офіційному ресурсі бібліотеки, головною метою бібліотеки є направленість у портативність та легкість використання[21]:

«…Core to the development philosophy of dlib is a dedication to portability and ease of use. Therefore, all code in dlib is designed to be as portable as possible and similarly to not require a user to configure or install anything…»

Найбільш розгорнутою та налаштованою вважається API для МП Python для якої необхідно заздалегідь скомпонувати вихідні файли за допомогою системи CMake, вона налічує дуже велику кількість функцій, констант та класів. Для мови Python є актуальними наступні приклади вирішення завдань за допомогою бібліотеки Dlib:

* Бінарна класифікація;
* Вирівнювання, кластеризація, виявлення обличчя, рис обличчя;
* Знаходження кандидатів;
* Глобальна оптимізація;
* Відстеження обличчя на відео.

Частіш за все Dlib розрізняють за двома версіями, що цілком відрізняються за архітектурою: Histogram Of Oriented Gradients (HOG) та Convolutional Neural Network (CNN), перша являє собою гістограму спрямованих градієнтів, що виконується на центральному процесорі (ЦП), а інша згорткова нейронна мережа, що виконується на ГП.

В залежності від обраної версії, Dlib може мати певні переваги та недоліки. Наприклад, якщо ми обираємо HOG, то в нас з’являється переваги у першому виклику бібліотеки и взагалі у загрузці усіх модулів та навіть у обробці зображень, якщо їх розмір достатньо малий. Але в той же час, ми маємо на увазі те, що цілком система буде працювати повільніше та не буде такою точною. CNN же напроти, перекриває усі ці недоліки, оскільки ГП порівняно із ЦП [22] має більше потоків та ядер – ЦП працює набагато швидше, але частіш за все обмежено своєю пам’яттю, а в ГП усе навпаки. Тобто у сфері машинного навчання частіш за все використовують графічні процесори через їх багато поточність.

1. OpenCV

OpenCV – це набір типів даних, функцій та класів для обробки зображень алгоритмами комп'ютерного зору. Реалізована на C/C++ компанією Intel. Вважається дуже популярною через те, що вона має відкритий код та розповсюджується безкоштовно. Була офіційно запущена у 1999 році, тобто аналогічно до бібліотеки DLib – є довгожителем у своїй сфері використання. Покриває не тільки проблему комп’ютерного зору для розпізнавання обличчя та машинного навчання взагалі, а також у багаті інших випадках. Головні алгоритми якими відзначається ця бібліотека є:

* Інтерпретація зображень;
* Калібрування камери за еталоном;
* Усунення оптичних спотворень;
* Виявлення подібності на зображеннях;
* Виявлення форми об’єкту;
* Відстеження за об’єктом;
* Сегментація об’єкту.

Основні модулі бібліотеки, якими вона оперує [23]:

1. Cxcore – ядро, що містить у собі базові структури даних та алгоритми;
2. CV – обробка зображень та комп’ютерного зору (фільтрація, геометричні та йому подібні перетворення, інше);
3. Highgui – модуль введення\виводу (I\O) зображень та результатів виконання та для створення інтерфейсу користувача;
4. Cvaux – експериментальні та неактуальні функції по типу просторового зору, стерео калібрації та іншого;
5. CvCam – модуль захоплення відео потоку.

Такі модулі, як CV та HighGUI є неслідкуючими від батьківського модулю Cxcore. Також існує додатковий модуль ML (Machine Learning) для машинного навчання, що є більш додатковим, аніж основним. Ієрархія модулів зображення на рисунок 1.12.

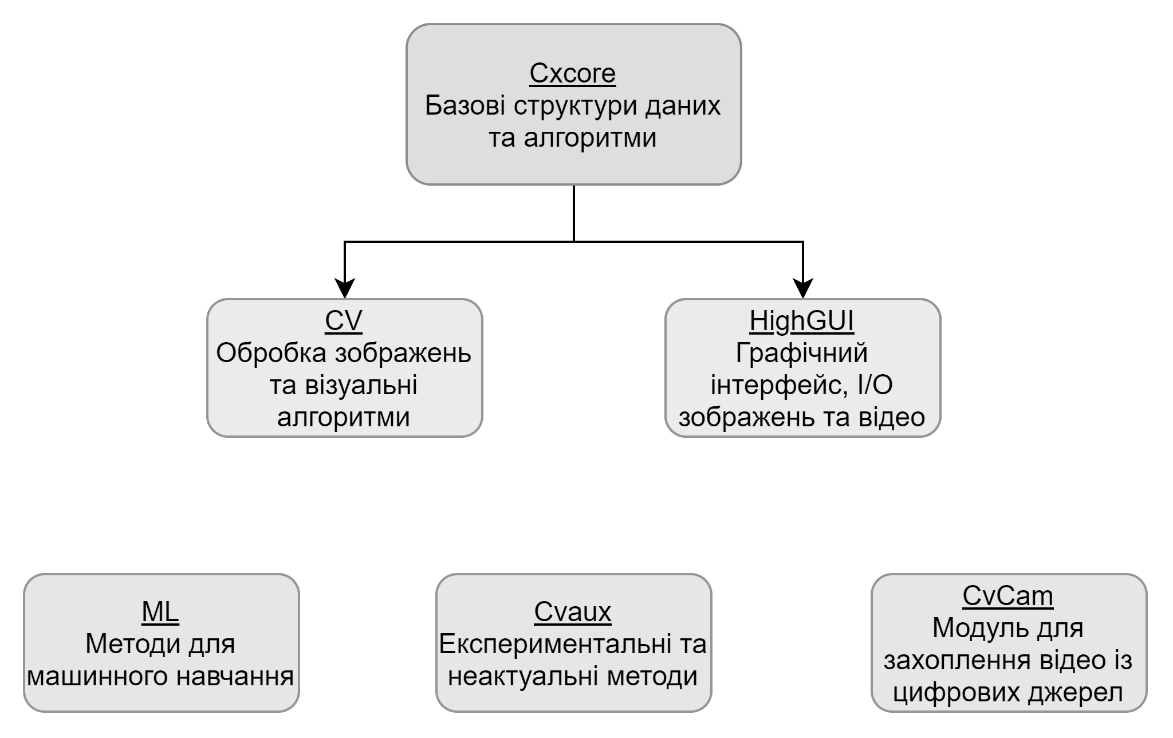


Рисунок 1.12 – Ієрархія та порядок модулів бібліотеки OpenCV

Також, аналогічно до бібліотеки Dlib, оскільки бібліотека OpenCV є дуже популярною и користується у попиті, вона має достатню кількість різноманітних API, що дозволяють використовувати її не тільки на МП C++. А також, наприклад, на Python.

Виходячи із аналізу описаних вище бібліотек, в якості порівнювальних будуть обрані бібліотека Dlib та OpenCV, оскільки матеріалу щодо їх порівняння недолік, вони обидві не є новими й мають достатню кількість документації та матеріалів до ознайомлення.

## Постановка задачі

Метою цієї роботи є створення веб-додатку для порівняння результатів бібліотек для розпізнавання обличчя за показниками точності, швидкодії та зручності використання. Обраними бібліотеками будуть Dlib та OpenCV, що використовуватимуся виключно для 2D-паттерну. Головна задача, яку необхідно вирішити, – реалізація із використанням обраних бібліотек та виконання наступних функцій:

* 1. Розпізнання обличчя та обличь на зображені;
  2. Ідентифікація обличчя на зображені, порівнюючи його із заздалегідь завантаженим шаблоном;
  3. Виявлення самого факту знаходження обличчя на зображені.

Описані функції є актуальними проблемами у сфері комп’ютерного зору та набуватимуть попиту. Не дивлячись на те, що с першого погляду, це тривіальні задачі, вони є достатньо трудомісткими залежно від умов виконання: кут зйомки, рівень освітлення, проблема «Близнюків» – все це є частими випадками розпізнавання обличчя і актуальними до вирішення, або до обрання найбільш оптимального та ефективного інструментарію. Також слід відзначити, що порівнюватися будуть результати бібліотек та використання також сіамських нейронних мереж (Siamese neural network - SNN).

Веб-додаток планується виконати за допомогою фреймворку на МП Python Django, що використовує шаблон проектування Model-View-Controller (MVC). Планується реалізація наступного функціоналу:

* + Завантаження зображення-шаблону до порівняння в залежності від обраної операції;
  + Завантаження двох зображень для порівняння;
  + Обрання методу до кожного зображення (Dlib, OpenCV чи сіамська нейронна мережа);
  + Збереження результату тестування.

Планується також порівняння результатів із використанням архітектури Compute Unified Device Architecture (CUDA), тобто із використанням ГП, щоб порівнювати швидкодію із ЦП.

## Висновок

В ході виконання аналізу актуальності теми, бібліотек та способів розпізнавання обличчя було вирішено, що через їх складність, розроблювальний веб-додаток має бути зручним й по-перше мати User-friendly інтерфейс, мати можливість завантаження зображень, обрання способу порівняння та збереження їх результатів. Має підтримувати використання ресурсів центрального та графічного процесорів за допомогою спеціальних технологій.

# ПРОЕКТУВАННЯ



## Аналіз мови програмування

Існує велика кількість мов програмування, що використовуються для нейронних мереж та суміжних із ними завдань, наприклад, комп’ютерного зору. Зазвичай до них відносять вимоги щодо швидкодії, обробки складних алгоритмів, поєднані їх із необхідними бібліотеками та фреймворками машинного навчання та у розробці скриптів-сценаріїв до останніх.

Для аналізу найбільш популярних та тих, що набирають у попиті, мов програмування використовується ресурс щодо індексу TIOBE Quality Indicator[24]. Виходячи із результатів за березень 2021-го року маємо гістограму на рисунку 2.1.

Рисунок 2.1 – Ріст попиту мов програмування

Опираючись на аналіз бібліотек для розпізнавання обличчя та нейронних мереж взагалі у главі 1.3, обрані мови програмування це сімейство C-подібних мов, Java, Python, Matlab.

* 1. C/C++/C#

Сімейство C широко використовується для створення програм штучного інтелекту. Їхні рідні бібліотеки та специфікації, такі як EmguCV, OpenGL та OpenCV мають вбудовані інтелектуальні функції для обробки зображень і можуть бути використані для швидкої розробки додатків штучного інтелекту (ШІ). Серед переваг С та С++ можливо відзначити високу швидкість, стабільність, ефективність і доступність, а для С# використовується виключно із фреймворком .NET та використовується частіш за все для розробки відеоігор, візуалізації даних та їх збереження.

Переваги сімейства:

* Колекція бібліотек та інструментів ШІ;
* Висока швидкість виконання;
* Об'єктно-орієнтовані принципи, корисні для організації даних.

Недоліки:

* Не підходить для виконання кількох завдань водночас;
* Вважається важкою для навчання через те, що є близькою до низьких мов програмування.
  1. Java

Одна з найбільш часто використовуваних мов, об’єктно-орієнтована Java має однакову потужність для створення простих настільних додатків і складних функціональних можливостей на основі штучного інтелекту. Він підходить для алгоритмів пошуку, нейронних мереж і рішень обробки природної мови.

Найбільшою перевагою Java є власні бібліотеки машинного навчання та розпізнавання зображень, за допомогою яких ви можете створювати програми з нуля. Він також сумісний з бібліотеками з відкритим кодом, наприклад, OpenCV. Крім того, рішення Java не залежать від платформи і можуть працювати на будь-якій платформі без перекомпіляції, оскільки вона має власну віртуальну машину JVM (Java Virtual Machine).

Java походить від сімейства C, взявши частину його синтаксису. У той же час вона має менше інструментів і її легше використовувати та налагоджувати, ніж, наприклад, C++.

Переваги:

* Портативність;
* Більш простий ніж C++;
* Власні бібліотека щодо ШІ.

Недоліки:

* Повільніший за С++;
* Можливо, знадобляться кардинальні зміни на старих платформах та версіях МП;
* Незважаючи на власні бібліотеки, вважається ще незрілою мовою у сфері ШІ.
  1. Python

Сьогодні Python і розпізнавання зображень є суміжними поняттями. Python це МП високого рівня, що підтримує функціональні, процедурні та об’єктно-орієнтовані стилі програмування, але має простий синтаксис і є портативним: його можна використовувати на платформах Windows, Mac OS, Linux та UNIX.

Програмування Python дає розумні можливості для рішення проблем нейронних мереж, ідентифікації зображень і рухів. Його сумісність з суміжними бібліотеками, такими як бібліотека ML з відкритим кодом TensorFlow, дає розробникам Python розумні інструменти для створення складних алгоритмів. Також зауважимо, що опираючись на главу 1.3, у найбільш популярних бібліотек зазвичай існує API використання на Python, що робить його використання ще більш зручнішим та привабливим.

Переваги:

* Портативність
* Велика кількість інструментарію та бібліотек для ШІ та розпізнавання обличчя зокрема;
* Зручне для проведення тестування та налагодження;
* Простіше за Java та C++;
* Швидше за Java та у деяких випадках, залежно від завдання, за C++;

Недоліки:

* Не підходить для мобільних додатків.

Розробка вбудованого програмного забезпечення та проекти IoT часто включають Python у свій технологічний стек.

* 1. Matlab

Matlab – це платформа програмування з набором вбудованих інструментів і функцій, а також матричною мовою для вчених та інженерів, що займаються обчислювальною математикою. Оскільки розпізнавання зображень і обчислення матриці взаємопов’язані, Matlab виявляється чудовим середовищем для додатків глибокого навчання та машинного навчання.

Використовуючи Matlab вирішують частіш за все завдання аналізу даних, побудови алгоритмів, розробки та вдосконалення моделей до останніх та проведення тестів.

Переваги:

* Матриці як базові елементи мови;
* Функціонал завжди може бути розширений за допомогою інструментарію;
* Зручність використання.

Недоліки:

* Може бути повільним, оскільки являє собою інтерпретовану мову.

Проаналізувавши переваги та недоліки описаних вище мов програмування, можливо зробити висновок, що МП Python являє собою найбільш здатну та зручну мову для виконання поставленого завдання.

## Середа розробки

Є велика кількість інтегрованих серед розробки (далі IDE) щодо написання, компіляції та налагодження програм на мові Python та їх основною різницею між собою зазвичай є зручність для користувача, автоматизація функціоналу (наприклад, завантаження бібліотек за посиланням та іншого). Python є універсальною мовою, що означає, що її можна використовувати в багатьох програмах, починаючи від програмне забезпечення 3D-моделювання і закінчуючи веб-розробкою.

IDE — це програмне забезпечення, яке складається із звичайних інструментів розробника в єдиний зручний графічний інтерфейс користувача (графічний інтерфейс користувача). IDE в основному складається з редактора вихідного коду для написання програмного коду, локальної автоматизації збірки для створення локальної збірки програмного. Нарешті, він має налагоджувач – програму для тестування інших програм. IDE може мати набагато більше функцій, крім цих, і вони відрізняються для кожної IDE.

Якщо говорити про IDE для МП Python, то виносимо наступні вимоги:

* Зберігати та перевантажувати вихідний код;
* Виконання з довкілля;
* Підтримка налагодження;
* Підсвічування синтаксису;
* Автоматизація сбірки;
* Автоматичне форматування коду.

Серед найбільш популярних IDE за наступним ресурсом[25] відзначають:

1. Visual Studio

Ця IDE має багатий функціонал, вбудовану підтримку Git, вбудований термінал, повний набір інструментів для тестування. Є маркетплейс із розширеннями. Visual Studio поставляється з Intellisense - однією з найкращих на сьогоднішній день систем автодоповнення. За допомогою плагінів можна додати підтримку інструментів веб-розробки, таких як Flask або Django.

Переваги:

* Багатий функціонал;
* Дуже різноманітна на мови програмування та платформи;
* Доступний спосіб додання плагинів;
* Автодоповнення за допомогою Intellisense.

Недоліки:

* Додатні ресурси – платні;
* Є занадто великою IDE щодо різноманіття функціоналу й можливостей для завдання цього проекту (тобто тільки МП Python).

1. IDLE

IDLE - IDE, що встановлюється з Python за замовчуванням. У ній є весь основний інструментарій: вікно з оболонкою Python, автодоповнення, підсвічування синтаксису, розумні відступи та базовий відладчик. Вона йде із обмеженим функціоналом, створена для людей, які вивчають програмування. Вона легковажна і не надто настроювана. Але через час можливо помітити помітите недоліки у продуктивності IDLE та нестачу потрібних функцій.

У IDLE також немає підтримки плагінів, а значить, ви не зможете розширити її функціонал, щоб наздогнати під свої потреби.

Переваги:

* Йде разом із Python;
* Легка у застосуванні та освоєні;
* Має усі основні інструментарії для програмування.

Недоліки:

* Низька швидкодія;
* Не може бути доданим додатковий функціонал.

1. PyDev

PyDev це IDE-плагін для Eclipse, популярної Java IDE для open-source розробки. Постачає з усіма вбудованими елементами керування Eclipse і додає підтримку автодоповнення, інтерактивної консолі та налагодження для Python. Це одна з небагатьох IDE, що мають підтримку Jython та Cython. Також є інтеграція із Django з його інструментами для веб-розробки.

Переваги:

* Легка до освоєння для користувачів IDE Eclipse;
* Має гарну оптимізацію щодо веб-розробки.

Недоліки:

* Обмежений функціонал, оскільки представляє із себе лише плагін;
* Якщо користувач незнайомий із IDE Eclipse – важка у освоєнні.

1. PyCharm

PyCharm - одна з найпопулярніших кросплатформових IDE для Python, створена і регулярно оновлюється компанією JetBrains. Ця IDE вже з коробки має всі найкращі інструменти: контроль версій, автодоповнення, підсвічування помилок, рефакторинг коду та інше. На відміну від інших середовищ розробки, PyCharm можна починати писати код відразу ж після завантаження. Має безкоштовну та платну версії, які достатньо сильно відрізняються. Також має підтримку щодо веб-додатків та автоматичного завантаження необхідних бібліотек та плагинів.

Часто відзначається особлива корисність пропонованого коду PyCharm (у порівнянні з іншими IDE).

Переваги:

* Має вісь необхідний функціонал;
* Не потребує тонкого налаштування;
* Добре підходе як для починаючих розробників, так й для розвинутих.

Недоліки:

* Висока ціна за платну версію;
* Порівняно із іншими IDE, високі вимоги до конфігурації персонального компьютеру (ПК).

Оскільки PyCharm є найбільш знайомою середою розробки для автору проекту та в цілком безкоштовна версія покриває усі необхідні вимоги до проекту, то вибором щодо середи розробки для проекту є саме PyCharm.

## Алгоритми

Опираючись на постановку задачі, робимо висновок, що основними алгоритмами щодо проекту будуть виступати завантаження, обробка та вивантаження результатів зображень у яких будуть виконуватися пошук обличь.

Також слід відзначити, що усі представлені алгоритми мають бути перероблені із вимогами та труднощами, що можуть виникнути на етапі розробки додатку.

Перед тим як розробляти алгоритми, був створений приблизний ескіз сторінки додатку, де саме будуть виконуватися усі дії. Сторінка повинна буде складатися із меню вкладок, що описуватимуть функції та інформацію щодо додатку: порівняння двох зображень за обраною бібліотекою за параметрами евклідового значення, точності та швидкодії, знаходження самому факту обличчя на зображені, можливість завантаження фото шаблону за яким будуть порівнюватися інші фото, збереження результату у текстовий файл та вкладка «Інформація», де будуть зберігатися інформація про додаток. Загальний вигляд інтерфейсу додатку виглядає наступним чином, дивитися рисунок 2.2.

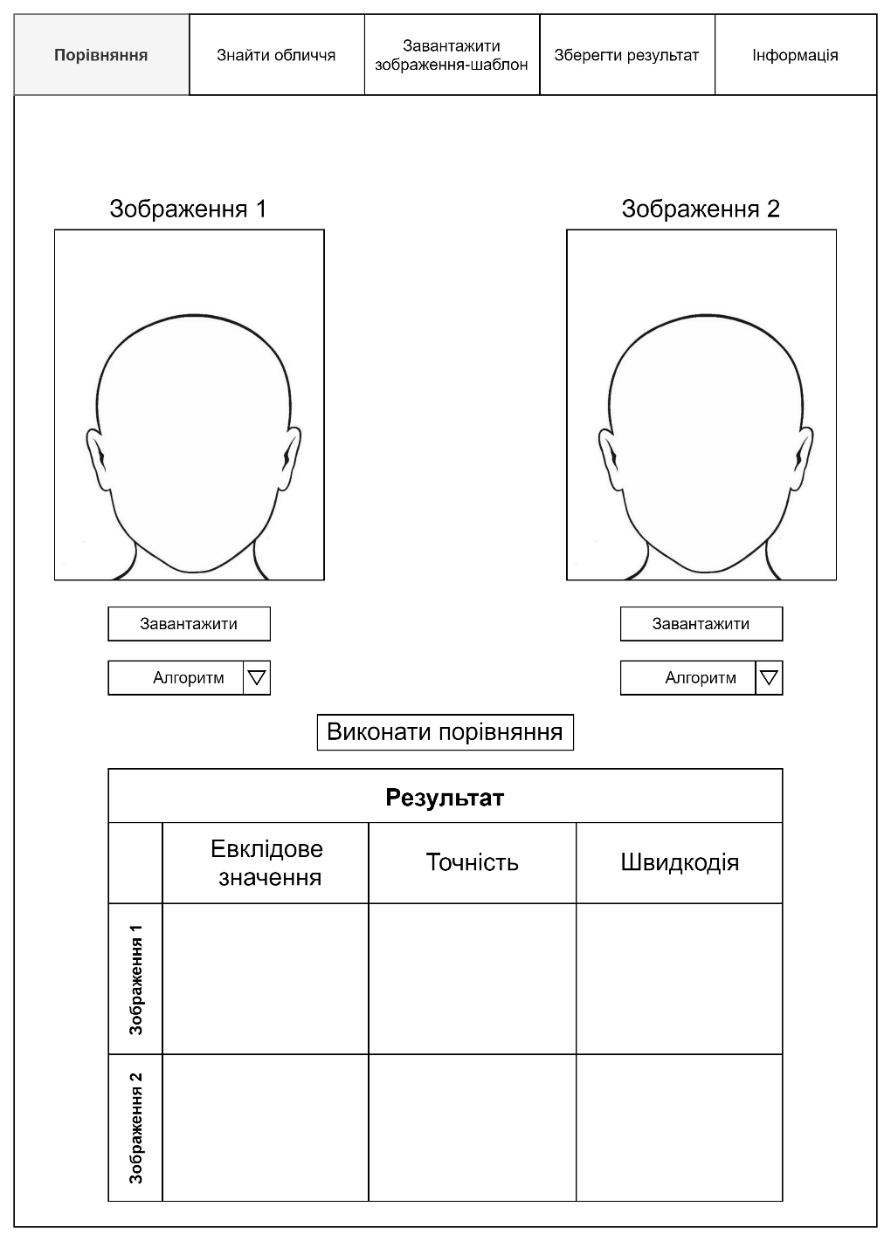


Рисунок 2.2 – Загальний ескіз інтерфейсу додатку

Відштовхуючись від ескізу інтерфейсу та описаних функцій у додатку можливо виконати проектування алгоритмів проекту.

Потрібне загальне уявлення щодо функціоналу додатку, для цього подивимось на рисунок 2.3.

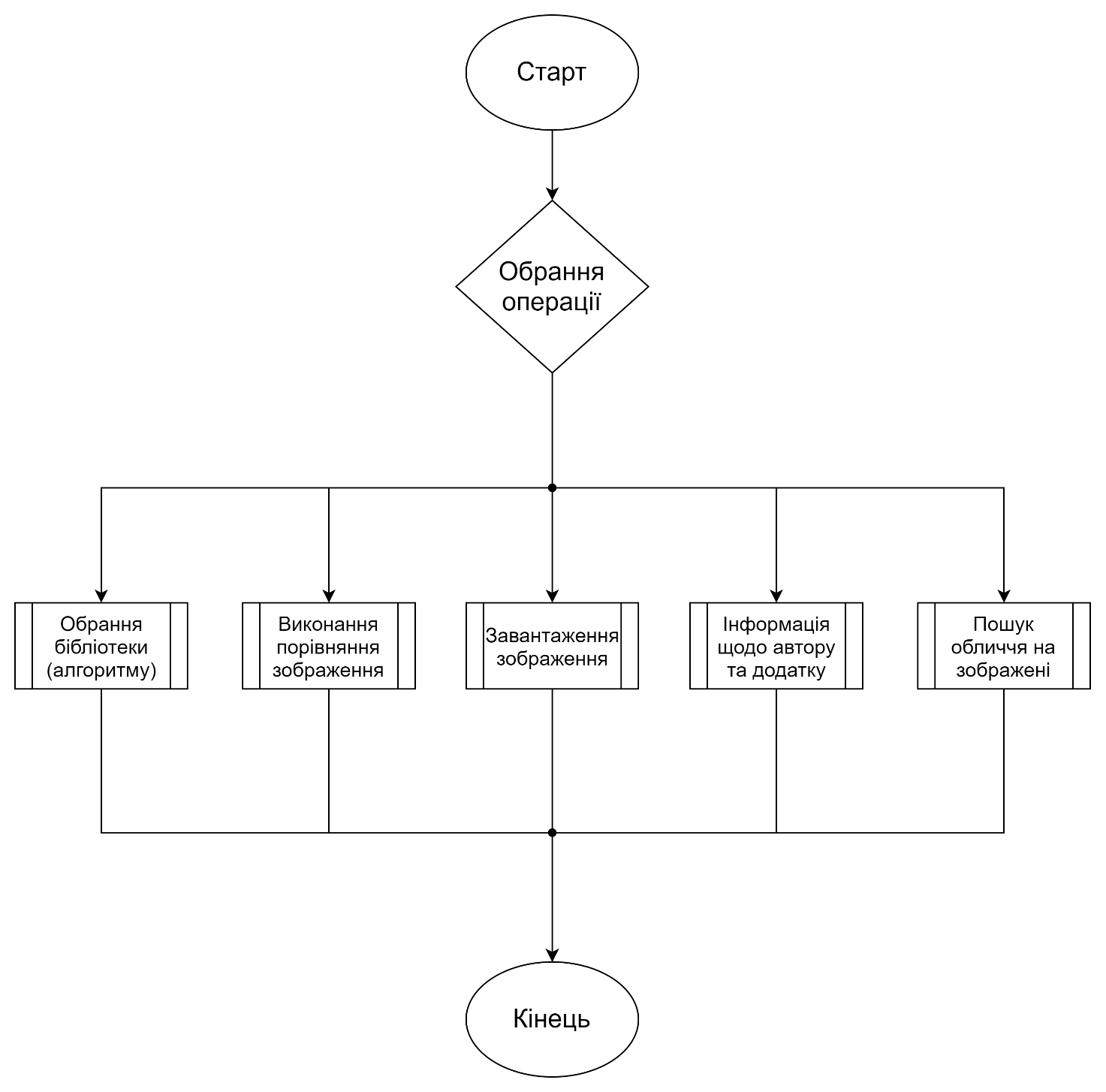


Рисунок 2.3 – Загальний алгоритм додатку

Далі виконаємо декомпозицію алгоритму на рисунку 2.3. Для початку опишемо функцію завантаження зображення, вона має бути однаковою для завантаження зображення-шаблону та зображень для виявлення обличчя, вона представлена на рисунку 2.4.

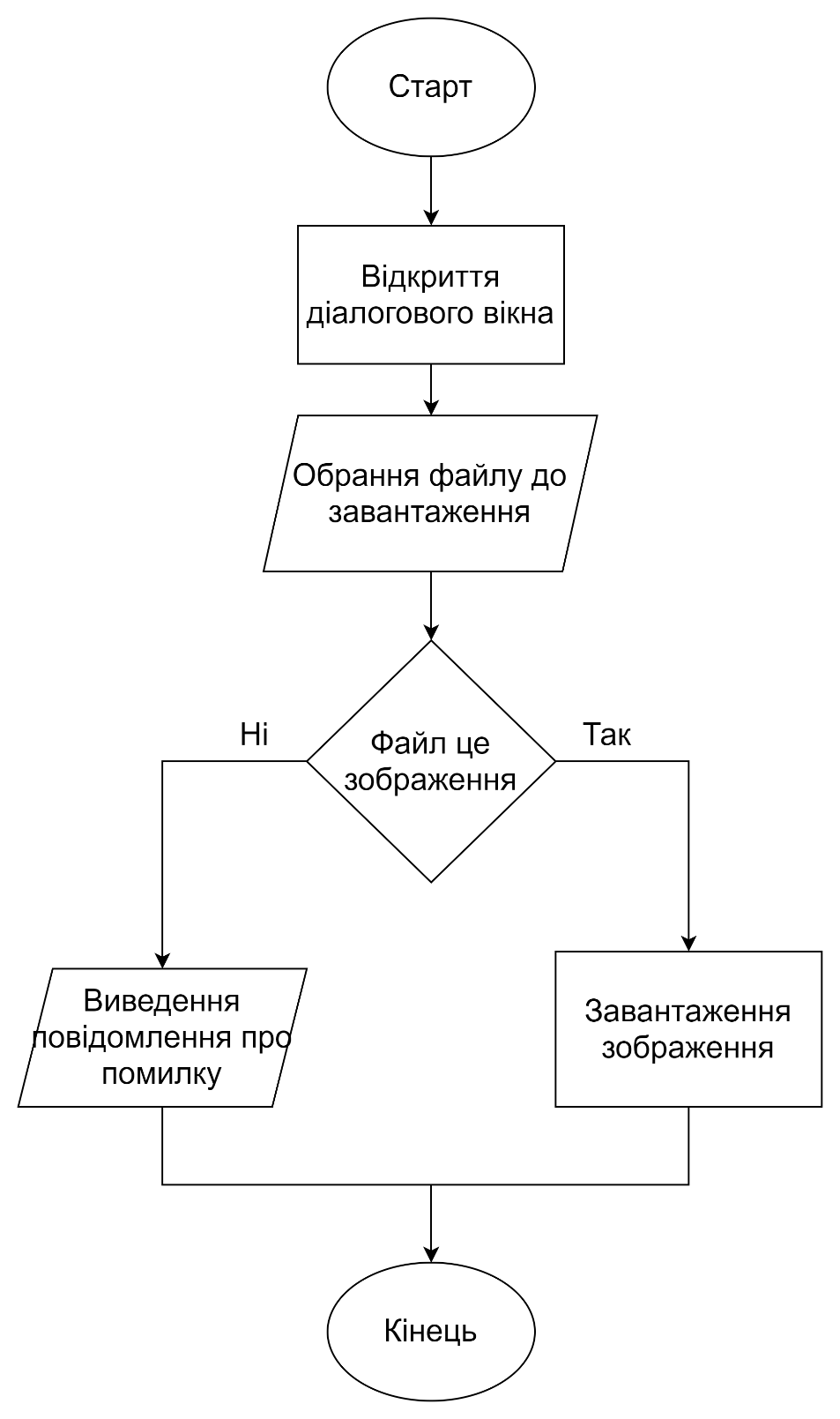


Рисунок 2.4 – Алгоритм завантаження зображення

Алгоритм для обрання бібліотеки до порівняння зображено на рисунку 2.5.

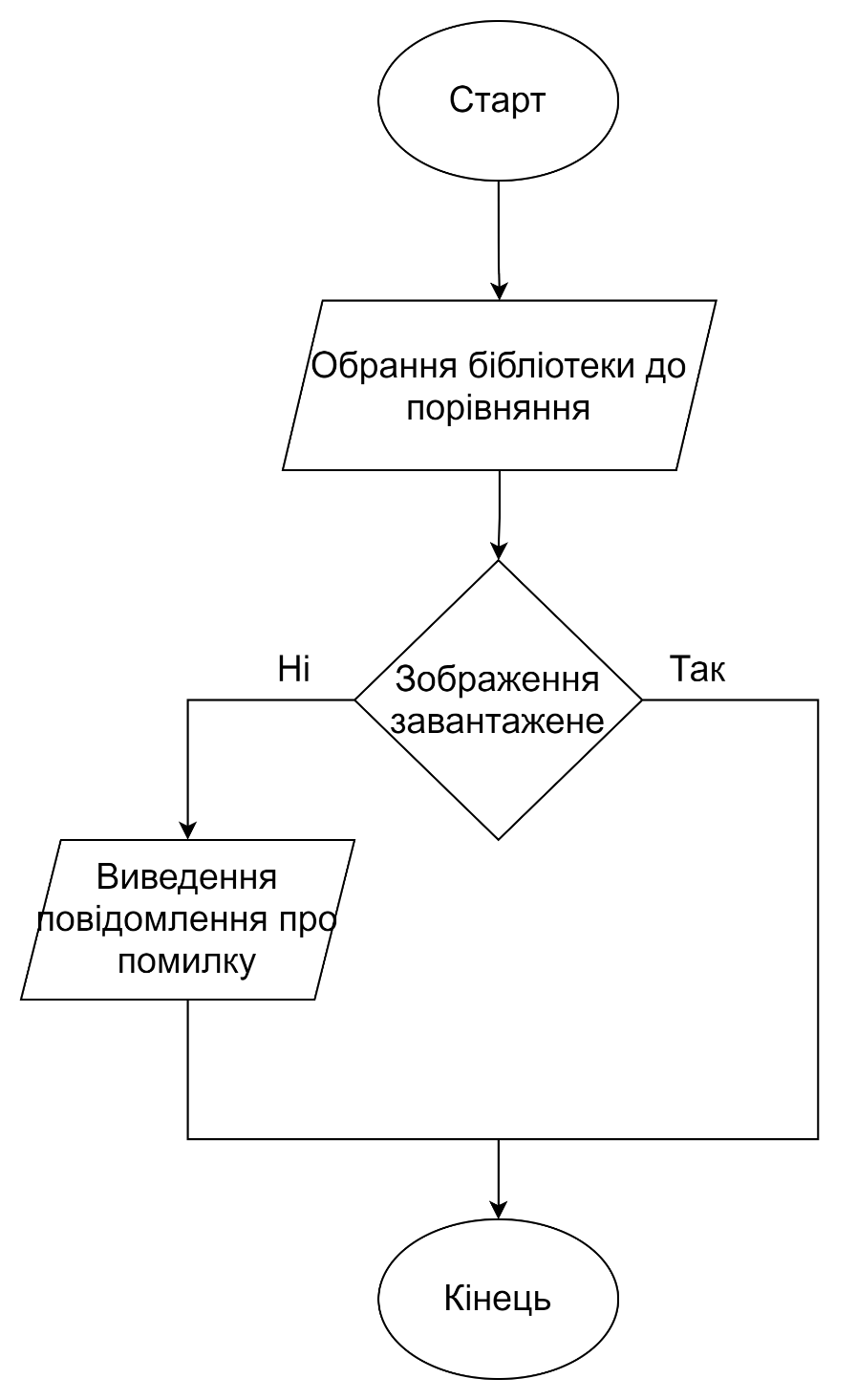


Рисунок 2.5 – Обрання бібліотеки до порівняння

Виконання порівняння зображено на рисунку 2.6 та виконується послідовно із використанням першої та другої бібліотек. Після кожної дії виконується перевірка на помилки та при успішному виконання обох – виведення результатів.

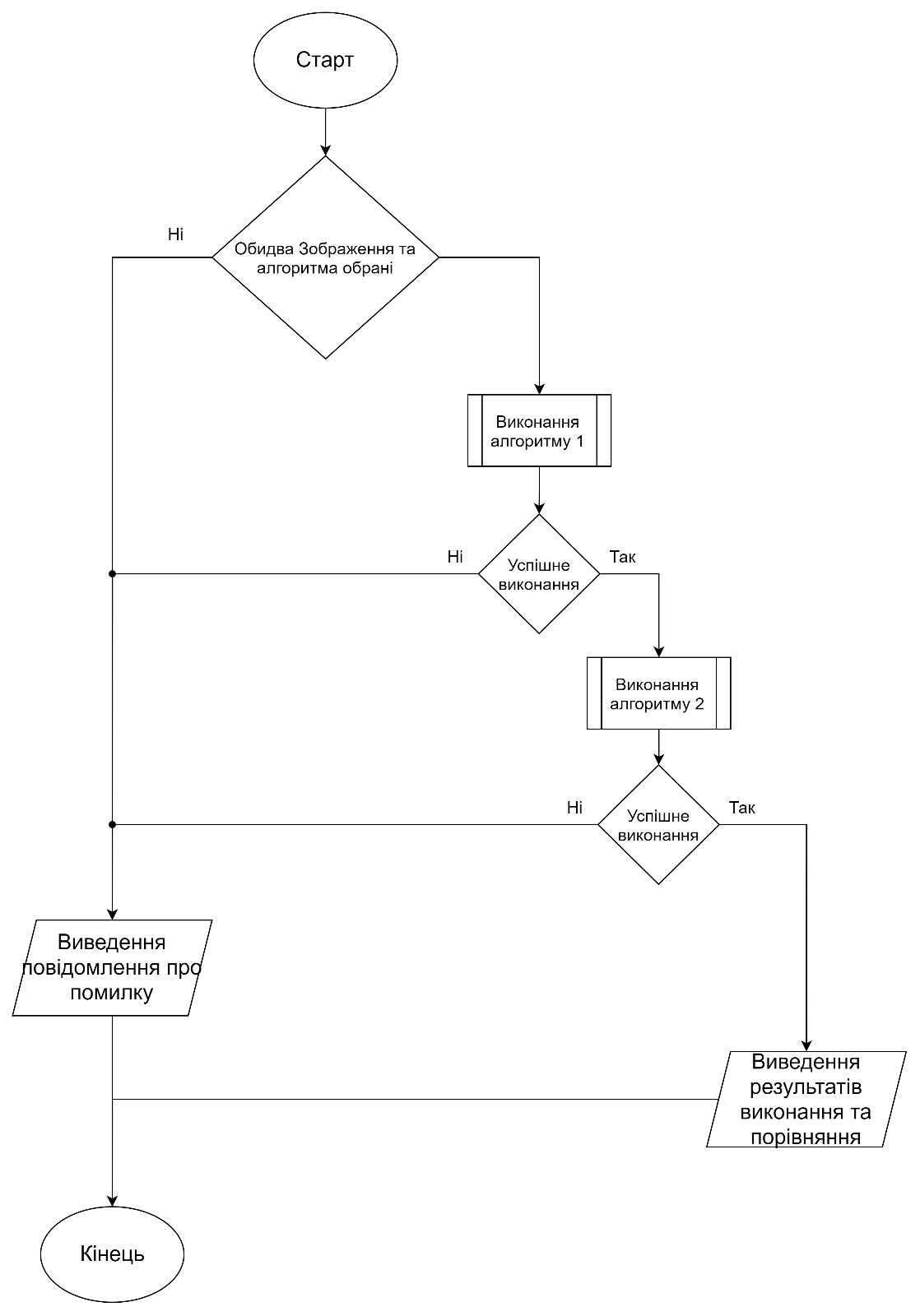


Рисунок 2.6 – Виконання порівняння

Найголовнішим є алгоритм роботи додатку щодо результатів пошуку обличчя на двох зображення (зображення-шаблон та завантажене зображення) за допомогою обраної бібліотеки. Цей алгоритм є достатньо абстрактним та віддаленим, але описує головну ідею та принцип роботи, оскільки будь яка бібліотека опирається на факт використання готової, або навченої в процесі, нейронної мережі, що за знайденими векторами ознак, тобто дескриптору обличчя, виявляла би належність обличчя на зображені до обличчя на зображені шаблону.

Алгоритм роботи зображено на рисунку 2.7.

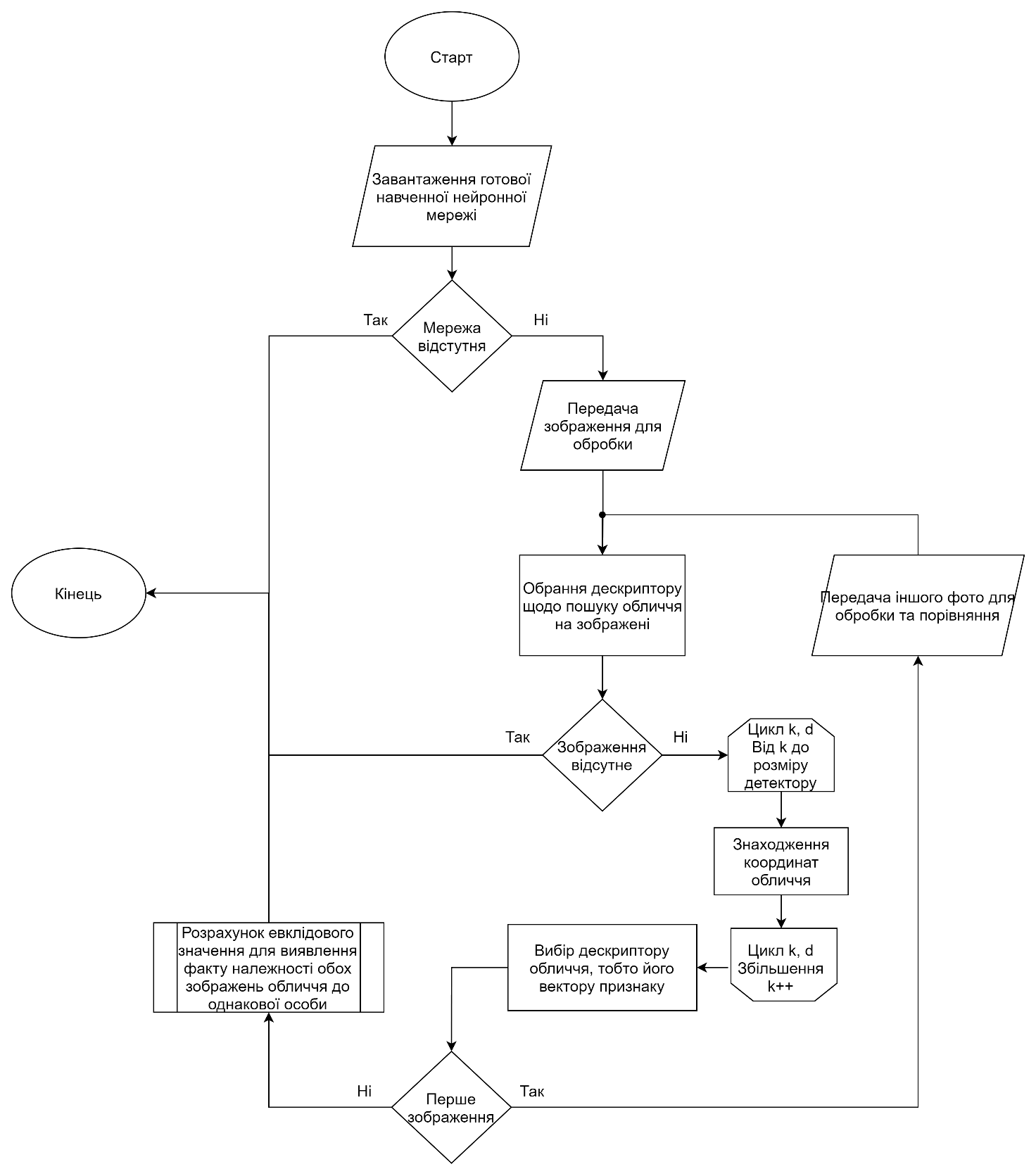


Рисунок 2.7 – Алгоритм роботи додатку із використанням бібліотеки щодо пошуку обличчя на зображеннях та порівняння результату

## UML-діаграми

У разі описання інтерфейсу та взаємодії кінцевого користувача із додатком, слід описати та зобразити UML-діаграму додатку у виді діаграми прецедентів, тобто взаємодії актора (користувача) із головними функціями додатку.

Аналогічно алгоритмам, ми повинні абстрагуватися від тонкощів та відобразити конкретно функції якими можуть користуватися та їх результат.

UML-діаграма зображена на рисунку 2.8, а більш докладне описання функціоналу у таблиці 2.1.

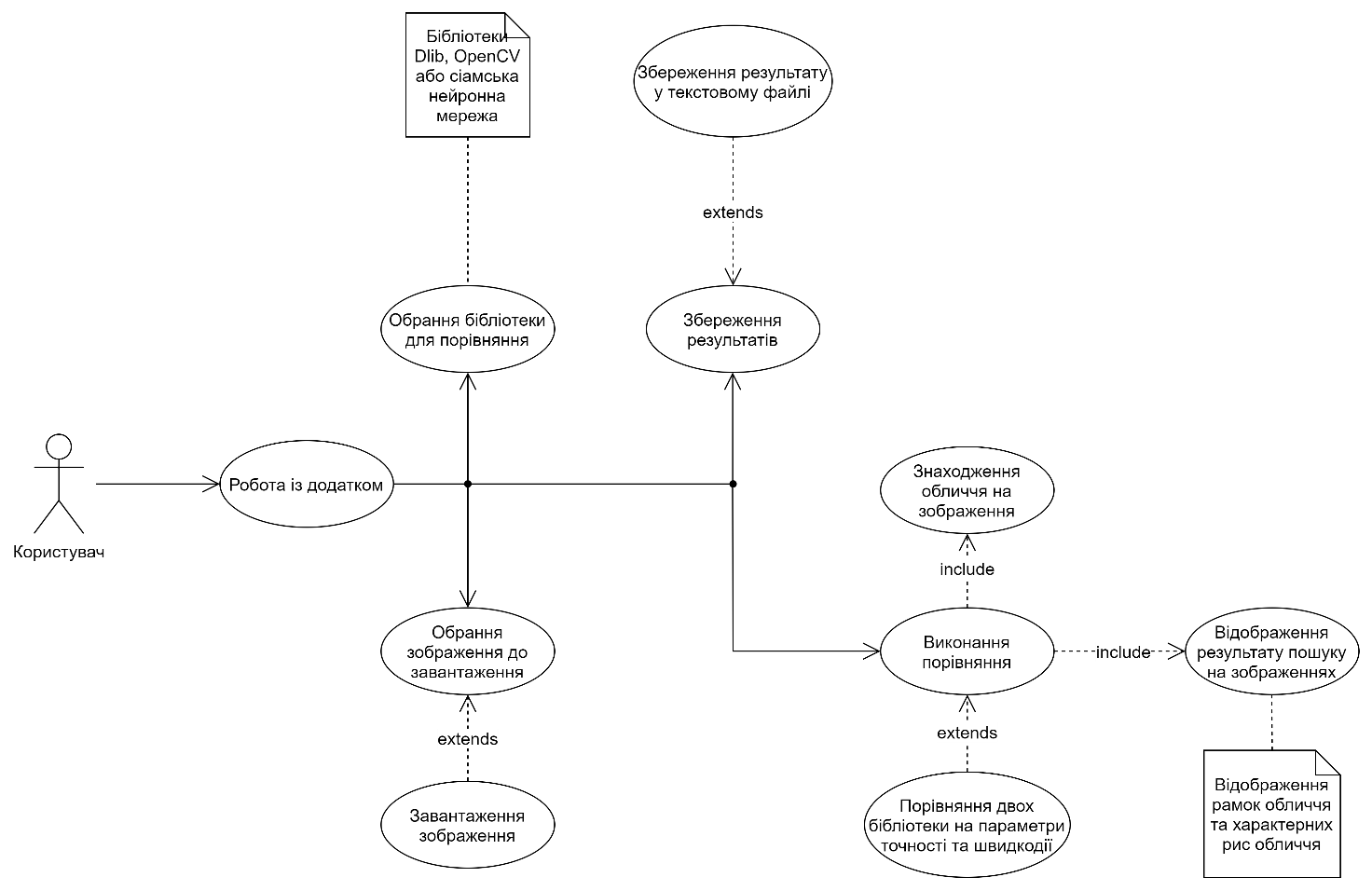


Рисунок 2.8 – UML-діаграма прецедентів

Таблиця 2.1 – Опис діаграми прецедентів

|  |  |
| --- | --- |
| Назва | Опис |
| Робота із додатком | Взаємодія із основними функціями додатку користувачем |
| Обрання бібліотеки для порівняння | Обрання бібліотек між якими буде пізніше виконуватися порівняння та виведений результат. Виконується після завантаження зображення |
| Обрання зображення до завантаження | Обрання зображення до завантаження та послідуючого порівняння знаходження обличчя за допомогою заздалегідь обраною бібліотекою |
| Збереження результатів | Збереження результатів у вигляді текстового файлу |
| Виконання порівняння | Виконання порівняння після завантаження зображень та обрання відповідних їм бібліотек. Порівняння виконується за параметрами швидкодії та точності у вигляді евклідового значення. |
| Відображення результату пошуку на зображеннях | Виконується у ході виконання порівняння бібліотек. Як приклад, рисунок 1.2, де зображені точки опису обличчя та його контур |
| Знаходження обличчя на зображені | Виконується у ході виконання порівняння бібліотек. Як приклад також виступає рисунок 1.2, де ми можемо бачити рамку із виділеним зображенням. Виконується за пошуком дескрипторів обличчя |

## Архітектура додатку

У архітектурі додатку буде описана загальна структура та взаємодія різних компонентів між собою, щоб була зрозуміла їх взаємозв’язок між собою. Заздалегідь слід сказати, що загальна структура буде описана на прикладі архітектури для веб-додатків за допомогою фреймворку Django, оскільки він виступає каркасом для цього завдання. БД Django замінюють стандартні БД із SQL класами із МП Python щоб ними було набагато легше користуватися. Також слід відзначити те, що для взаємодії із класами у вигляді БД, використовується система CRUD (create, read, update, delete)[26].

Django передбачає використання MVC-патерн, шо розширюється до MTV (Model Template View), для створення та обробки сторінок. Усі виклики кінцевого користувача оброблюються у вигляді шаблонів для динамічного створення сторінок. У фреймворку Django існує для цього власна мова DTL – Dynamic Template Language, що створює шаблони у вигляді HTML-сторінок. Опишемо спочатку загальну діаграму роботи веб-додатку на рисунку 2.9.

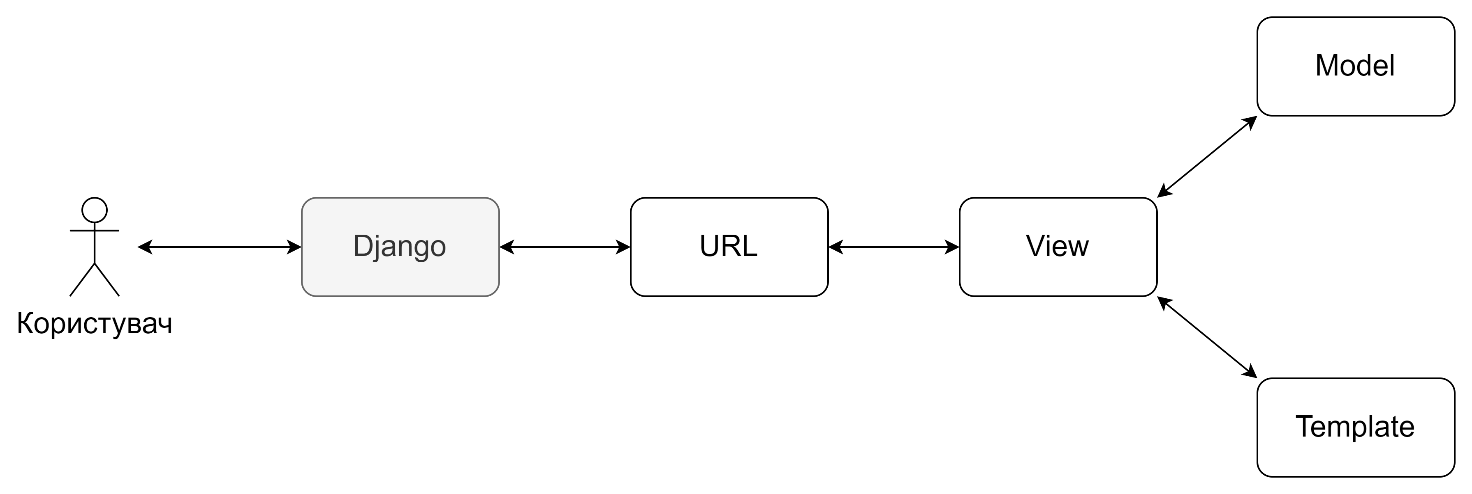


Рисунок 2.9 – Загальна діаграма роботи веб-додатку

Як ми бачимо, головним компонентом, через який виконується взаємодія, є сам Djangо, що виступає у ролі посередника між додатком та користувачем. Слід також описати головні компоненти паттерну MVT (MVC) у таблиці 2.2.

Таблиця 2.2 – Опис компонентів MVT (MVC)

|  |  |
| --- | --- |
| Компонент | Опис |
| Model | Опис даних |
| View | Керує відображенням даних для користувача |
| Template | Контролює взаємодію із користувачем |
| Controller | Виділяє шар, що буде представлений користувачу (обрання View |

Далі необхідно більш детально описати саме архітектуру додатку, для цього скористаємося рисунком 2.10.

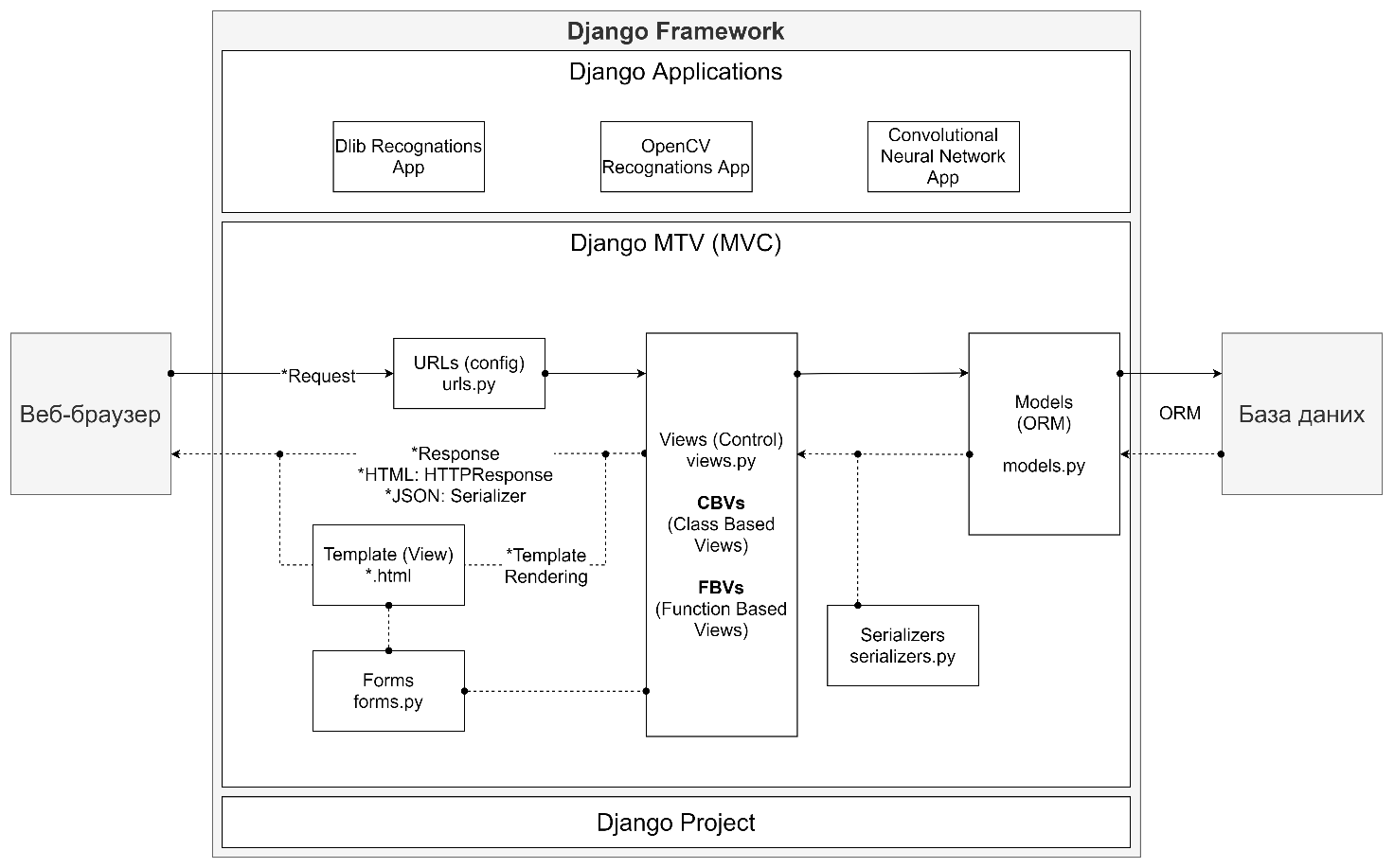


Рисунок 2.10 – Загальна архітектура веб-додатку

Увесь додаток знаходиться у просторі Django Framework. У середовищі Django Applications маємо основні додатки які використовуються: додаток бібліотек Dlib, OpenCV та сіамської нейронної мережі. Далі йде структура MTV (MVC) патерну, вона складається із основних елементів цього патерну та допоміжних: Model, View, Template, а в кінці Django Project. Зовнішніми зазвичай є сам веб-браузер через який приходить запит, та до якого йде відповідь, та база даних в якій зберігаються усі дані.

Послідовно додаток працює наступним чином: веб-браузер робить запит до додатку, той за допомогою URL визначає необхідну сторінку и у компоненті Views контролює, якою повинна бути сторінка, визначивши, чи існує вона взагалі чи ні, CBV – це вигляди зумовлені класами, а FBV – функціями. У останніх зазвичай зберігають вигляді щодо помилок, тобто помилки 404, 403, 500 за протоколом HTTP. Якщо сторінка існує, то робиться запит до компоненту моделей де створюється вже запит до БД за допомогою технології об’єктно-реляційного відображення (ORM) , яка перетворює необхідні дані у близький до об’єктно-орієнтованих мов програмування вигляду. За допомогою серіалізації ми зберігаємо ці дані та створюємо вже необхідну сторінку через компоненти форм та шаблонів (Forms, template відповідно). В кінці користувач отримує коректну сторінку або повідомлення об помилці.

Django applications описують спосіб обробки даних, що були обрані чи передані. Є більш комплексною структурою.

Django Project передбачає саме фреймворк Django, що дозволяє усім компонентам взаємодіяти між собою.

## Архітектура сіамської нейронної мережи

Оскільки нейронні мережі вже були описані, то варто було би залучити до уваги архітектуру сіамських нейронних мереж, що складає із себе дві або більше (триплети) симетричних нейронних мереж з однаковою структурою, що після виконання обчислень поєднуються за допомогою деякої функції енергії – Е. В даному описі, слово «сіамський» передбачає в своєму значені «однаковий», тобто ми беремо два різних векторів із однаковими вагами, що працюють в тандемі (разом), для розрахунку вихідних векторів значень. Один із векторів визначений як базова лінія яку ми будемо порівняти із іншими вихідними векторами. Частим прикладом подібного застосування можливо привести відбитки пальців. В літературі частіш за все використовують поняття[27] «Хешування з урахуванням розташування» - це алгоритмічний метод із високою точністю, що відносить одні дані до інших, відбираючи подібні. Його основна задача – кластеризація даних та пошук найближчого сусіда.

Говорячи про розпізнання обличчя, сіамська мережа має приймати на вхід парні значення із результатом щодо їх стану – чи однакові вони, чи ні. Це необхідно для того, щоб заздалегідь налаштувати та навчити її. В якості підмереж можуть виступати, наприклад, об’єднання підмереж багатошарового персептрону або згорткових нейронних мереж, що описувалися раніше.

Намагаємося відобразити та описати сіамську нейронну мережу[28] у рисунку 2.11.

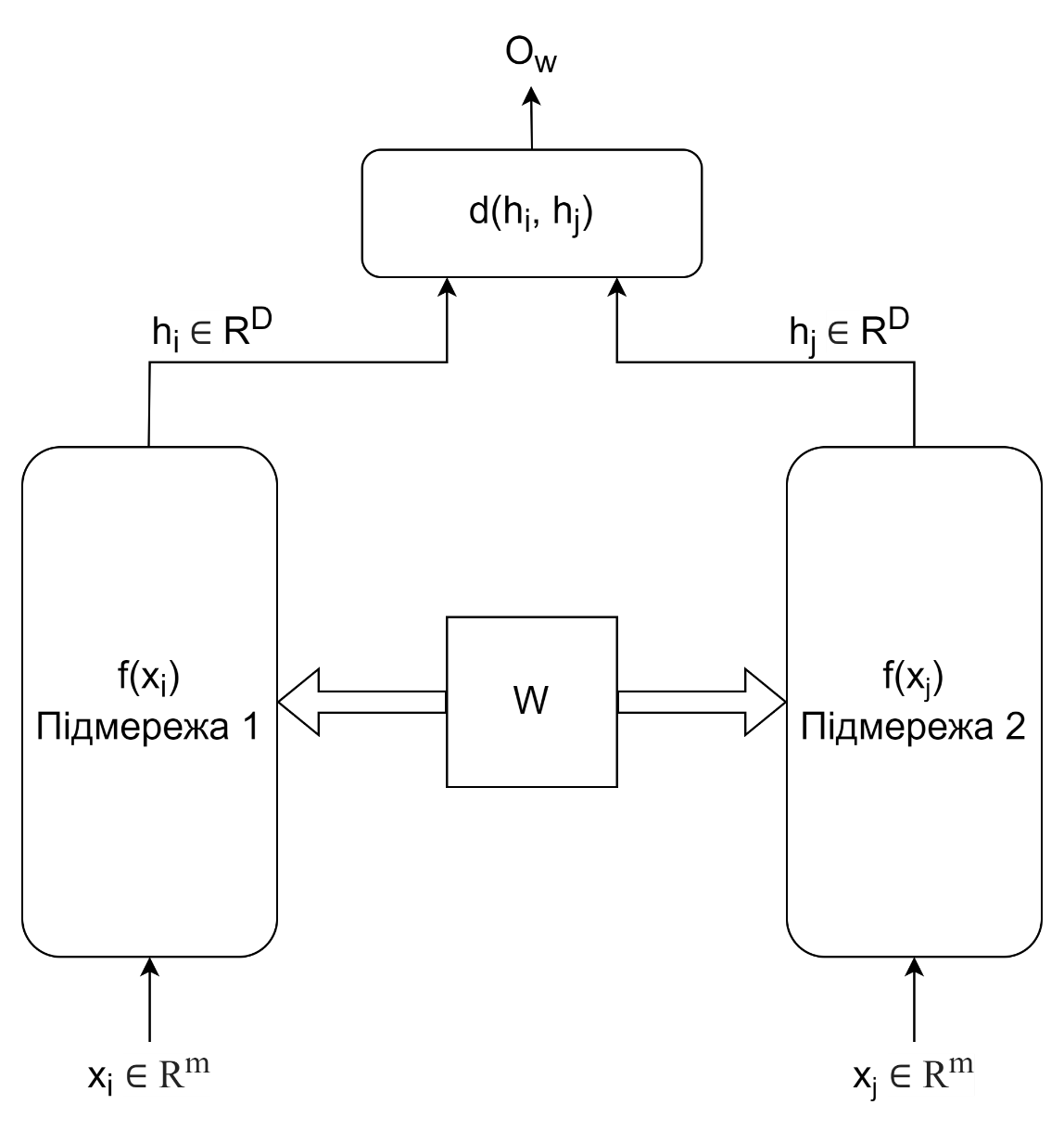


Рисунок 2.11 – Узагальнена архітектура сіамської нейронної мережі

Опишемо вхідні дані, що подаються на вхід мережі: є набір даних , що складається із n векторів ознак, де розміру m із міткою , де – число класів. Створюємо навчальний набір , що складається з пар із бінарними мітками . Якщо обидва вектори ознак відносяться до одного класу, то , інакше 1. Навчальний набір розділяється на два підгрупи: одна – зі схожими ознаками, а інша із різними.

Цілком сіамська мережа може представлятися трьома різними структурами до яких подаються вхідні дані і які входять потім до кінцевого модуля, що визначає евклідове значення (відстань) між ними й вирішує, чи відносяться вони до одного класу, чи ні.

Дивлячись на рисунок 2.11, можна зрозуміти, що та – вхідні дані, W – загальна вага параметрів, а та – виходи обох підмереж. Сіамська мережа передбачає собою відображення для якого евклідова відстань максимально низьке при умові, що та максимально велике навпаки. Мережа повертає оцінку того, наскільки різняться між собою та .

Існує три головні структури сіамських мереж, опишемо їх нижче та зобразимо на рисунку 2.12:

1. Вхідні та подаються на дві паралельні підмережі, що представляють із себе дві різні мережі із однаковими вагами та зміщеннями. Обидва виходи цих мереж використовуються для розрахунку описаної метрики відстані (), що по ознаці вимірює схожість між входами підмереж;
2. Декілька шарів підмереж об’єднуються в одну для вирішення задачі класифікації за допомогою логістичної функції softmax для багатомірних випадків: вона перетворює вектор z розмірності K у вектор , де кожна координата представлена дійсним числом та сума яких дорівнює 1. Формула розрахунки координат:

(2.1)

1. Обидва вхідних векторів обєднуються и подаються у вигляді однієї мережі, де на кінці розраховуються також за softmax-перетворенням.

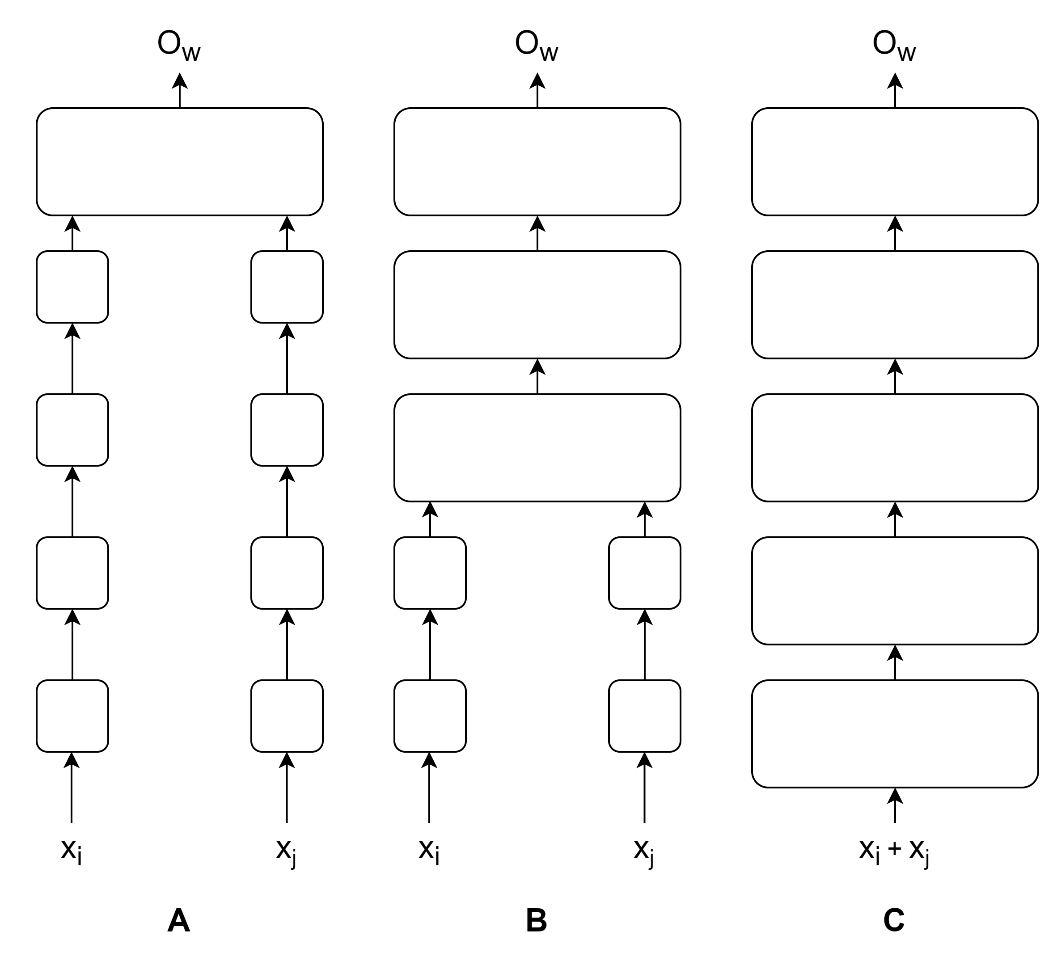


Рисунок 2.12 – Види структур сіамських мереж



## Висновок

Проаналізував актуальні мови програмування та середи розробки, їх доступність та складність до використання, опираючись на поставку задачі, було вирішено використовувати Python як МП та PyCharm як IDE. Вони повністю покривають необхідний мінімум щодо вимог для розробки веб-додатку для розпізнавання обличчя за використання декількох бібліотек та сіамських нейронних мереж, які після будуть порівнюватися. Була описана загальна структура додатку, а також у деталях описана математична модель сіамських нейронної мережи, що буде виступати тандемом при об’єднанням бібліотек.

Також був доцільно описаний сам додаток: його алгоритми роботи головних функцій та архітектура, створена UML-діаграма прецедентів для описання інтерфейсу та взаємодії кінцевого користувача із веб-додатком.

# ПЕРЕЛІК ДЖЕРЕЛ

1. The Future of Neural Networks. URL: https://www.researchgate.net/publication/230899572\_The\_Future\_of\_Neural\_Networks (дата звернення: 10.10.2021).
2. Facial Recognition Market by Component (Software Tools (3D Facial Recognition) and Services), Application (Law Enforcement, Access Control, Emotion Recognition), Vertical (BFSI, Government and Defense, Automotive), and Region - Global Forecast to 2025. URL: https://www.marketsandmarkets.com/Market-Reports/facial-recognition-market-995.html (дата звернення: 10.10.2021).
3. UK Court of Appeal finds facial recognition technology unlawful. URL: <https://www.simmons-simmons.com/en/publications/ckelg1z7p8kjt09008l0bet7c/uk-court-of-appeal-finds-facial-recognition-technology-unlawful> (дата звернення: 10.10.2021).
4. Facial Recognition Market by Technology, End-user, Application, and Geography - Forecast and Analysis 2021-2025. URL: <https://www.technavio.com/report/facial-recognition-market-industry-analysis> (дата звернення: 10.10.2021).
5. Face Recognition Methods & Applications. C.U. Shah College of Engg. & Tech
6. Технология распознавания лиц от «А» до «Я». URL: [https://securityrussia.com/blog/face-recognition.html#31](https://securityrussia.com/blog/face-recognition.html%2331) (дата звернення: 10.10.2021).
7. National Institute of Standards and Technology – Face Recognition Vendor Test. URL: <https://www.nist.gov/programs-projects/face-recognition-vendor-test-frvt-ongoing> (дата звернення: 10.10.2021).
8. MegaFace and MF2: Million-Scale Face Recognition. URL: <http://megaface.cs.washington.edu/results/facescrub.html> (дата звернення: 12.11.2021).
9. Labeled Faces in the Wild. URL: <http://vis-www.cs.umass.edu/lfw/index.html> (дата звернення: 10.11.2021).
10. Fine-grained face verification: FGLFW database, baselines, and human-DCMN partnership
11. C. A. Hansen, “Face Recognition”, Institute for Computer Science University of Tromso, Norway
12. Rise of Surveillance Camera Installed Base Slows. URL: <https://www.sdmmag.com/articles/92407-rise-of-surveillance-camera-installed-base-slows> (дата звернення: 01.11.2021).
13. M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces", 1991.
14. W. Zhao, R. Chellappa, P. J. Phillips & A. Rosenfeld, “Face recognitions literature survey”, ACM Computing Surveys, Vol. 35, No. 4, December 2003, p. 399–458.
15. Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression. URL: <https://aaronsplace.co.uk/papers/jackson2017recon/> (дата звернення: 10.10.2021).
16. Bkav’s new mask beats Face ID in "twin way": Severity level raised, do not use Face ID in business transactions. URL: <https://www.bkav.com/top-new/-/view-content/65202/bkav-s-new-mask-beats-face-id-in-twin-way-severity-level-raised-do-not-use-face-id-in-business-transactions> (дата звернення: 10.10.2021).
17. ISO/IEC DIS 39794-5
18. Deep Neural Nets Can Now Recognize Your Face in Thermal Images. URL: <https://www.technologyreview.com/2015/07/24/167019/deep-neural-nets-can-now-recognize-your-face-in-thermal-images/> (дата звернення: 09.10.2021).
19. Hebb, D. O. The organization of behavior: a neuropsychological theory
20. Why use Keras? URL: <https://keras.io/why-use-keras/> (дата звернення: 10.10.2021).
21. Dlib official site. URL: <http://dlib.net> (дата звернення: 10.11.2021).
22. 2018 6th International Conference on Control Engineering & Information Technology (CEIT): “Performance Analysis and CPU vs GPU Comparison for Deep Learning”
23. Основні модулі OpenCV. URL: <https://github.com/opencv/opencv/tree/4.x/modules> (дата звернення: 11.11.2021).
24. TIODE Index for March 2021. URL: <https://public.flourish.studio/visualisation/5509518/?utm_source=showcase&utm_campaign=visualisation/5509518> (дата звернення 07.12.2021)
25. 10 Best Python IDE & Code Editors. URL: <https://hackr.io/blog/best-python-ide> (дата звернення 10.12.2021)
26. Django Explained – IBM. URL: <https://www.ibm.com/cloud/learn/django-explained> (дата звернення 12.12.2021)
27. Siamese Neural Network: Overview. URL: <https://link.springer.com/protocol/10.1007%2F978-1-0716-0826-5_3> (дата звернення 12.12.2021)
28. Сиамская Нейронная Сеть – Вики ИТМО. URL: <http://neerc.ifmo.ru/wiki/index.php?title=Сиамская_нейронная_сеть> (дата звернення 12.12.2021)