

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

ПРОГРАМА МОНІТОРИНГУ ДОТРИМАННЯ МАСКОВОГО РЕЖИМУ
ІЗ ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

THE PROGRAM FOR MONITORING COMPLIANCE WITH THE MASK
MODE USING NEURAL NETWORKS

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ІПЗ-41

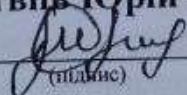
Краснопольський Павло
Олександрович


(підпис)

Керівник:

д.т.н., професор

Матвійв Юрій Ярославович


(підпис)

Кваліфікаційну роботу
допущено до захисту

«8» 06 2023р.

Гарант освітньої програми:

к.т.н., доцент

Ліщина Наталія Миколаївна


(підпис)

Луцьк – 2023 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 121 Інженерія програмного забезпечення

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

« 2 » 02 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Краснопольському Павлу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Програма моніторингу дотримання маскового режиму із використанням нейронних мереж

Керівник роботи: д.т.н., професор, Матвіїв Юрій Ярославович
затверджені наказом закладу вищої освіти від «23» грудня 2022 р. № 961-01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи «8» червня 2023 р.

3. Вихідні дані до роботи: технічне та програмне забезпечення ЕОМ, вимоги до розробки програмного забезпечення, ергономічні вимоги до функціонування програмного засобу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):

Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз і вибір засобів проектування, опис функціонального наповнення об'єкта проектування, розробка й обґрунтування системного наповнення, оцінка ергономічних та надійнісних параметрів проекрованої системи, функціонально-структурна схема роботи об'єкта проектування.

5. Перелік графічного матеріалу: 15 рисунків, 3 лістинги.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Матвій Ю.Я.		
Специфікація вимог до розробленої системи	Матвій Ю.Я.		
Розробка об'єкта проектування	Матвій Ю.Я.		
Нормоконтроль	Матвій Ю.Я.		
Гарант ОП	Ліщина Н.М.		
Показник запозичень тексту		37%	
Академічна доброчесність	Яциук А.А.		

7. Дата видачі завдання «2» лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ З/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	05.02.2023 р.	вик
2	Аналіз проблеми розробки та впровадження об'єкту проектування	27.02.2023 р.	вик
3	Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання	10.03.2023 р.	вик
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	17.04.2023 р.	вик
5	Практична реалізація об'єкта проектування та розробка бази даних	18.05.2023 р.	вик
6	Тестування та налагодження об'єкта проектування	01.06.2023	вик
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру	08.06.2023 р.	вик

Здобувач вищої освіти

(підпис)

Краснополаский П. П.
(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис) Матвій Ю.Я.
(прізвище, ініціали)

**Рецензія
на кваліфікаційну роботу**

Здобувач вищої освіти:	Краснопольський Павло Олександрович
Спеціальність і група:	121 Інженерія програмного забезпечення, ІПЗ-41
Обсяг кваліфікаційної роботи бакалавра:	49 стор., 13 рис., 3 лістинги, 15 літературних джерел
Кількість сторінок записки:	49 сторінок

Тема: Програма моніторингу дотримання маскового режиму із використанням нейронних мереж

Коротка характеристика кваліфікаційної роботи та прийнятих рішень:

Кваліфікаційна робота бакалавра складається з трьох розділів, в яких проаналізовані проблематика та поставлені завдання за темою роботи, здійснено теоретичне дослідження та практичну реалізацію програми моніторингу дотримання маскового режиму із використанням нейронних мереж.

Самостійні розробки і пропозиції автора: Кваліфікаційна робота бакалавра присвячена розробці додатку, який в режимі реального часу розпізнає лицьові маски на обличчі людей. Для навчання нейронної мережі було знайдено декілька ресурсів із великими обсягами даних осіб людей в масках та без, а також обрано бібліотеки для подальшої реалізації алгоритму.

Недоліки: Істотних недоліків в роботі не виявлено.

Загальний висновок: У кваліфікаційній роботі бакалавра було спроектовано та створено розробити програмний продукт який буде призначений для онлайн моніторингу людей із використанням нейронних мереж. Робота є результатом самостійного дослідження. Істотних недоліків не виявлено. Кваліфікаційна робота здобувача освіти Краснопольського Павла рекомендується до захисту екзаменаційній комісії та заслуговує відмінної оцінки.

Рецензент: К. М. Н., доцент, факультет інженерії програмного забезпечення
(прізвище, ім'я, по-батькові, посада)

Міхненко В.О.
Рецензент кваліфікаційної роботи

(підпис)

« 13 » 06 2023 р.

АНОТАЦІЯ

Краснопольський П.О. Програма моніторингу дотримання маскового режиму із використанням нейронних мереж. Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності Інженерія програмного забезпечення. Луцький національний технічний університет. Луцьк, 2023.

Кваліфікаційна робота бакалавра складається з вступу, трьох розділів, висновків, списку використаних джерел, додатків.

Мета роботи – розробити програмний продукт який буде призначений для онлайн моніторингу людей із використанням нейронних мереж. Він ілюструватиме детектування із використанням комп’ютерного зору.

У першому розділі описується проблемна галузь, проводиться аналіз аналогічних програм та обґрунтовується вибір мови програмування. У другому розділі виконується опис програмного засобу, описуються особливості обраної мови програмування, вибір середовища для розробки та описується структура програмного засобу. Третій розділ містить у собі блок схему роботи проекту, опис програми та інструкцію з інсталяції та роботи додатку.

Ключові слова: COVID-19, коронавірус, пандемія, машинне навчання, громадське здоров’я, автоматизація, нейронні мережі, штучний інтелект, Deep learning, Tensorflow, Python, OpenCV, комп’ютерна графіка, машинне навчання.

ANNOTATION

Krasnopol'sky P.O. The program for monitoring compliance with the mask mode using neural networks. Manuscript.

Bachelor's qualifying work of the OP "Software Engineering" specialty Software Engineering. Lutsk National Technical University. Lutsk, 2023.

The bachelor's qualification work consists of an introduction, three sections, conclusions, and a list of used sources, applications.

The goal of the work is to develop a software product that will be designed for online monitoring of people using neural networks. It will illustrate detection using computer vision.

The first chapter describes the problem industry, analyzes similar programs, and justifies the choice of programming language. The second section describes the software tool, describes the features of the selected programming language, the choice of development environment, and describes the structure of the software tool. The third section contains a block scheme of the project, a description of the program and instructions for installing and operating the application.

Keywords: COVID-19, coronavirus, pandemic, machine learning, public health, automation. neural networks, artificial intelligence, Deep learning, Tensorflow, Python, OpenCV, computer graphics, machine learning.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра.....	16
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЕНОЇ СИСТЕМИ ...	17
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення.....	17
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання.....	29
РОЗДІЛ 3 РОЗРОБКА ОБ'ЄКТА ПРОЕКТУВАННЯ.....	32
3.1 Особливості використання мов програмування, баз даних; особливості технологій, програмного забезпечення	32
3.2 Структура автоматизованої системи (підсистеми), бази даних, програми.....	36
3.3 Структурно-функціональна схема системи	37
3.4 Опис розробленого програмного (інформаційного) забезпечення обчислювальних систем	38
3.5 Інструкція користувача з інсталяції та використання програмного забезпечення	44
ВИСНОВКИ.....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48

ВСТУП

Актуальність теми. Робота присвячена проблемі розпізнавання людей в масках, а також створенню інформаційної системи розпізнавання осіб в масках та без них в режимі реального часу.

Епідемії продовжують бути присутніми в житті суспільства, і хвороби, що поширюються повітряно-крапельним шляхом, все ще представляють серйозну загрозу. Якщо причиною захворювання є вірус, існує ризик його мутації, що може зробити розроблені вакцини менш ефективними.

Найкращим прикладом цього є світова пандемія COVID-19, яка спричинила повний локдаун у багатьох країнах через високу смертність. Це призвело до глобальної економічної кризи, зупинки комунікацій та банкрутства багатьох підприємств. Виявилося, що повний карантин не є універсальним рішенням, оскільки може завдати більше шкоди, ніж користі.

Найбільш ефективним заходом у боротьбі з COVID-19 виявився режим носіння масок. Досвід показав, що це є більш прийнятними витратами, а ті, хто дотримується цього режиму, мають менше ризику заразити оточуючих. Крім того, носіння масок допомагає знизити ризик зараження, оскільки закриває вразливі ділянки носу та рота.

Недостатньо відповідальне ставлення пересічних громадян стало єдиною перешкодою на шляху до досягнення ідеального маскового режиму. Багато установ усвідомлюють це і забезпечують карантинний контроль при вході. Однак такі заходи не можуть гарантувати, що окремі особи не зніматимуть маску відразу після перевірки і не створюватимуть серйозної загрози здоров'ю присутніх у приміщенні. Крім безпосередньої небезпеки для відвідувачів, порушення карантинних норм може призвести до повного закриття установи на карантин та спричинити фінансові збитки.

Мета дослідження полягає в розробці ПЗ для полегшення спостереження маскового режиму в людних місцях за допомогою використання сучасних технологій.

Об'єкт дослідження – технології проектування та реалізації із розпізнання образів.

Предмет дослідження – послідовний процес розробки, засоби та технології реалізації ПЗ для полегшення спостереження маскового режиму в людних місцях.

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести аналіз наукових джерел, що стосуються даної предметної області;
- вивчити існуючі рішення з розпізнавання облич за допомогою методів машинного навчання;
- проаналізувати наявні інструменти, що використовуються для розробки таких систем;
- дослідити алгоритми і методи навчання нейронних мереж;
- розробити інформаційну систему класифікації облич з використанням методів машинного навчання;
- вивчити вплив різних факторів на ефективність роботи розробленої системи через проведення досліджень.

Новизна. Невисока вартість розгортання даної системи та високий рівень автоматизації.

Практичне значення. Цінність цієї роботи виявляється у можливості практичного застосування отриманих результатів для ефективного використання нейронних мереж у завданнях розпізнавання облич. Варто відмітити, що запропонована модель може бути змодифікована з метою досягнення кращої ефективності розпізнавання.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз сучасного стану проблеми

На сьогоднішній день нейронні мережі є альтернативою існуючим алгоритмам для різних завдань, таких як машинний переклад, розпізнавання мови та музики, обробка зображень, визначення об'єктів на фото та відео. Основою глибокого навчання є нейронні мережі, хоча можуть використовуватись й інші методи. В сучасному світі нейронні мережі використовуються майже у всіх сферах, пов'язаних з глибоким навчанням.

Успішність глибокого навчання безпосередньо залежить від потужності обчислювальної техніки. Початково, коли нейронні мережі з'явилися, потужності комп'ютерів були недостатніми, що обмежувало їхню ефективність. Тому було неможливо створити мережі з великою кількістю шарів, а це безпосередньо впливає на їхні можливості. Але з появою графічних процесорів (GPU) і спеціалізованих процесорів для глибокого навчання (TPU) ситуація змінилась. Сучасні методи глибокого навчання здатні працювати з великими мережами завдяки збільшеній потужності обчислювальної техніки.

Застосування згорткових нейронних мереж (CNN) є однією з найвпливовіших інновацій в області комп'ютерного зору. Вони використовуються для розпізнавання та класифікації зображень та обличь. Вони застосовуються в різних сферах, таких як забезпечення безпеки в аеропортах і вокзалах, системи допуску до корпорацій з високим рівнем секретності, а також для зручних покупок в повсякденному житті.

Генеративно-суперечливі мережі (GAN) використовуються в криміналістиці для створення фотороботів злочинців за описом, в дизайні для створення одягу або інтер'єру з урахуванням їх призначення та в кіновиробництві для зміни освітлення або вікових характеристик героїв.

Застосування пристроїв, здатних розпізнавати об'єкти, дозволяє замінити людину спеціалізованими автоматами у багатьох складних системах. Це

розширює можливості систем, що виконують інформаційні, логічні та аналітичні завдання. Важливо відзначити, що якість роботи людини на робочому місці залежить від багатьох факторів, тоді як спеціалізований автомат завжди забезпечує стабільну якість. Автоматичний контроль складних систем дозволяє моніторити їх стан та забезпечувати своєчасне обслуговування, ідентифікувати перешкоди та автоматично застосовувати методи шумозаглушення, що покращує якість передачі інформації. Використання автоматичних систем у деяких завданнях дозволяє досягти неможливої для людини швидкодії.

Вирішальні правила – це методика віднесення елементів до певних образів. Метрика – це метод, який використовується для вимірювання відстані між елементами універсальної множини. Чим менша ця відстань, тим більш схожі об'єкти, які ми розпізнаємо. Зазвичай об'єкти представляються у вигляді числових наборів, а метрика – у вигляді функції. Вибір представлення об'єктів і реалізації метрики впливає на ефективність програми. Різні алгоритми розпізнавання з різними метриками можуть мати різну частоту помилок.

Адаптація – це процес зміни структури системи та параметрів на основі поточної інформації з метою досягнення деякого стану системи при змінних умовах роботи та початковій невизначеності. Навчання – це процес, під час якого система набуває здатності реагувати на зовнішні впливи, а адаптація – це налаштування структури системи і параметрів з метою досягнення потрібної якості управління в умовах постійних змін зовнішніх умов.

Системи виявлення порушників за зображенням з відеокамери ґрунтуються на комп'ютерному зорі. При створенні такої системи необхідно досягти високої точності класифікації зображень, що може бути досягнуто шляхом оптимального вибору моделі машинного навчання та тренувального набору даних.

Оптимальним набором даних для даної кваліфікаційної роботи щодо поставленої задачі є той, що включає в себе велику кількість зображень з чітко вираженими рисами, які використовуються для класифікації на наявність або відсутність маски на обличчі людини. Крім того, набір даних повинен бути

повним і містити всі можливі варіації зображень (зокрема профілі обличь), які слід розпізнавати як різні класи. Також важливо, щоб набір даних був однорідним та включав мінімальну кількість некоректних прикладів.

Для цих потреб можна використовувати класичний набір даних, який розділяє зображення на дві категорії: обличчя з масками та без масок. Однак, є кілька способів отримання необхідної кількості таких екземплярів:

- генерація обличь з масками шляхом накладання зображення маски на обличчя без масок;
- використання доступних відкритих джерел зображень, що містять екземпляри обох категорій класифікації без додаткової обробки.

Враховуючи вимогу до повноти набору даних, було прийнято рішення вибрати другий варіант.

Оптимальною моделлю машинного навчання для цієї задачі може бути модель, яка ефективно розрізняє риси обличь людей з медичними масками, здатна виконувати класифікацію з низькою частотою помилкових розпізнавань і нерозпізнавань. Тобто, модель має мінімізувати перенавчання або недонавчання. З цими вимогами була створена система розпізнавання, що складається з двох моделей: модель грубого розпізнавання облич на зображеннях і модель класифікації цих обличь на дві категорії.

У деяких школах вже наявні недорогі камери спостереження, які є необхідними засобами для соціального моніторингу. Однак, головною перевагою моніторингу соціального дистанціювання за допомогою систем машинного зору є їх безконтактність та автоматизованість. В працях науковців та практиків наведено наступні можливості моніторингу соціального дистанціювання з використанням систем машинного зору:

- а) автоматичне виявлення та відстеження декількох динамічних об'єктів (осіб) у режимі реального часу;
- б) аналіз траєкторій руху людей та рівня порушення соціального дистанціювання;

в) визначення зон підвищеного ризику з найбільшою небезпекою можливого поширення коронавірусу.

Оскільки поширення високозаразних штамів COVID-19, таких як «омікрон», стає все більш актуальним, останній аспект набуває додаткової важливості.

Розпізнавання обличчя в масці є одним із способів контролювати дотримання маскового режиму в людних місцях, таких як університети чи школи. Алгоритми глибокого навчання автоматично виявляють людей без масок та відстежують порушення. Модель розпізнавання обличчя в масках на основі комп'ютерного зору є неінвазивною, масштабованою та порівняно легкою у реалізації, оскільки можна використовувати відеозапис будь-якої камери (рис. 1.1) [1].



Рисунок 1.1 – Розпізнавання обличчя в масці за допомогою [1]

Визначимо наступні переваги моніторингу маскового режиму за допомогою інтелектуальних систем машинного зору:

а) автоматичний моніторинг дотримання маскового режиму є більш послідовним і точним, ніж перевірка людиною;

б) виявлення людей без масок підвищує безпеку, знижуючи ризик передачі COVID-19 та інших інфекційних захворювань;

в) використання респіраторів або масок для обличчя на робочому місці може запобігти спалахам заражень в школі та університеті, особливо при поширенні штамів високої заразності.

Вже існують відеосистеми, які забезпечують карантинні заходи, проте їх застосування є досить обмеженим. Однією з основних причин такої ситуації є висока вартість доступних на ринку рішень. Ці витрати можуть бути обґрунтовані лише для великих компаній з великою кількістю відвідувачів. Для середнього та малого бізнесу подібні системи фінансово не доступні. Існує значний попит на продукти зі схожими можливостями, але за меншу ціну.

Існують відомі програмні рішення, які можна використовувати для підтримки функціональності маскового режиму.

Одним з них є SecurOS, розроблений компанією ISS. Цей продукт є системою програмної ідентифікації людей в приміщенні. SecurOS має можливість підключення до звукової сирени та температурних камер, а також надає мобільний додаток для працівників охорони. Проте, вартість цього продукту досить висока, оскільки він пропонує обширну систему, а не спеціалізоване програмне рішення для розпізнавання масок [2].

Іншим продуктом є Glen, розроблений компанією Humanizing Technologies. Цей продукт представляє собою статичного робота зі штучним інтелектом, який нагадує людям надягати маску при переході між приміщеннями. Glen надає голосового асистента і не використовує камери спостереження. Він також інтегрується з мобільним додатком.

Однак, продукт Glen має свої недоліки, такі як відсутність інтеграції з системою охорони, обмежена можливість контролювати лише певні ділянки між кімнатами, а також низьке співвідношення між його функціональністю і ціною.

Що стосується продукту Neuromation, це система розпізнавання масок, яка не вимагає окремих пристроїв для роботи, крім встановлених відеокамер. Вона має кілька переваг, таких як розумні сповіщення, необхідність лише базового

комп'ютерного обладнання та незалежна працездатність від моделі відеокамери. Однак, недоліком Neuromation є відсутність інтеграції з мобільним додатком.

М. Резай (Mahdi Rezaei) та М. Азармі (Mohsen Azarmi) розроблено DeepSOCIAL – гібридну систему машинного зору на основі глибокого навчання для автоматизованого виявлення людей у натовпі в приміщеннях та на вулиці за допомогою звичайних камер відеоспостереження (рис. 1.2).

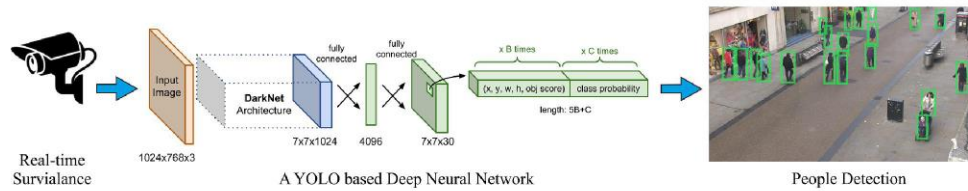
Запропонована авторами модель у поєднанні з адаптованою технікою відображення зворотної перспективи та алгоритмом відстеження SORT веде до надійного виявлення людей та моніторингу соціального дистанціювання.

Модель була навчена на основі двох наборів даних – Microsoft Common Objects in Context (MS COCO) і Google Open Image. Оцінку моделі було проведено в складних умовах, включаючи часткову видимість та при змінах освітлення із середньою точністю 99.8% і швидкість в реальному часі 24,1 кадр/с.

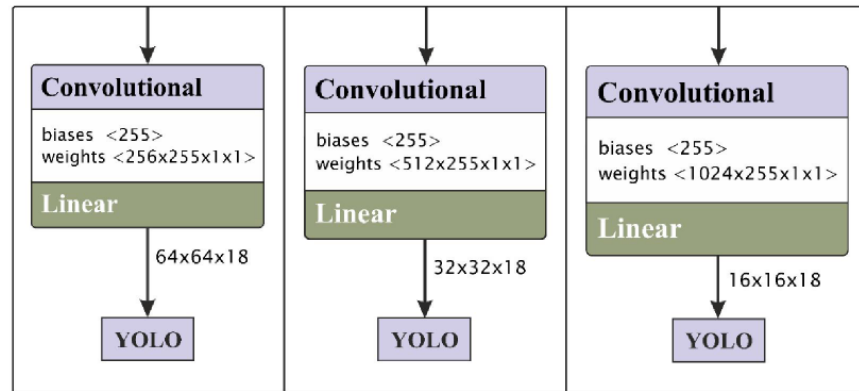
DeepSOCIAL також надає онлайн-схему оцінки ризику зараження шляхом статистичного аналізу просторово-часових даних про траєкторії пересування людей і частоти порушень соціального дистанціювання, визначаючи зони високого ризику з найбільшою ймовірністю поширення вірусу та інфікування.

Це може допомогти органам влади змінити планування громадських місць або вжити запобіжних заходів, щоб пом'якшити зони високого ризику. Розроблену модель можна застосувати в багатьох інших областях, таких як автономні транспортні засоби, розпізнавання людських дій, виявлення аномалій.

У своєму огляді К. Шортен, Т. М. Хошгофтаар та Б. Фурхт розглядають широкий спектр застосувань глибокого навчання, зокрема велике значення приділяється застосуванням машинного зору (рис. 1.3) [3].



(a)



(b)

Рисунок 1.2 – Загальна структура системи машинного зору DeepSOCIAL (a) та архітектура нейронної мережі (б)

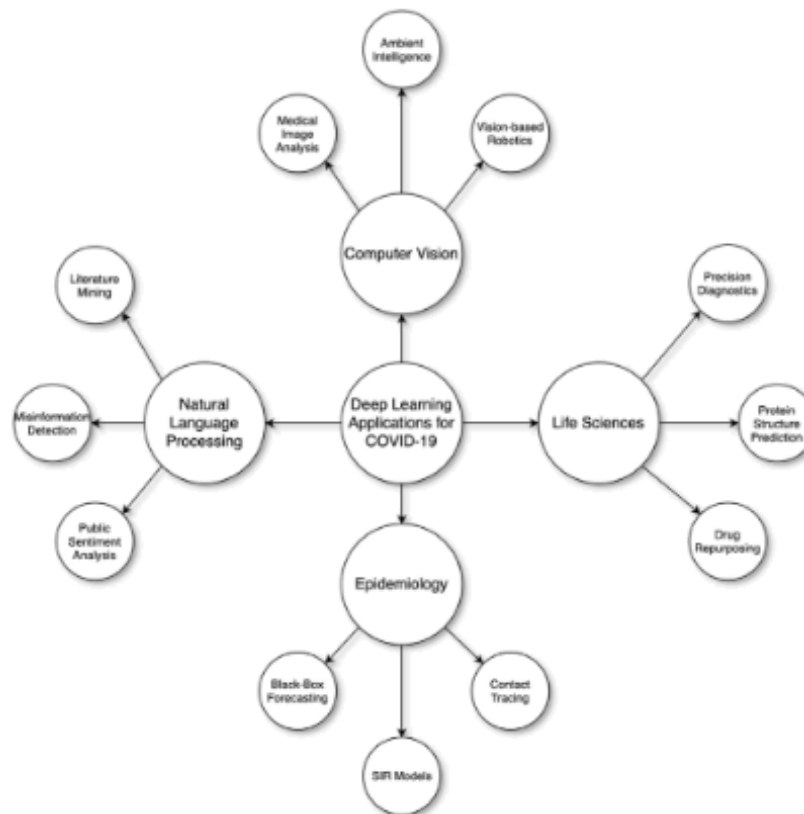


Рисунок 1.3 – Застосування глибокого навчання для подолання наслідків COVID-19

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Мета роботи – розробити програмний продукт який буде призначений для онлайн моніторингу людей із використанням нейронних мереж.

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести аналіз наукових джерел, що стосуються даної предметної області;
- вивчити існуючі рішення з розпізнавання облич за допомогою методів машинного навчання;
- проаналізувати наявні інструменти, що використовуються для розробки таких систем;
- дослідити алгоритми і методи навчання нейронних мереж;
- розробити інформаційну систему класифікації облич з використанням методів машинного навчання;
- вивчити вплив різних факторів на ефективність роботи розробленої системи через проведення досліджень.

Висновки до розділу 1

В даному розділі проведений огляд літературних джерел з предметної області; здійснений огляд підходів виявлення і відстеження людей на відео, аналіз додатків машинного навчання, а також розглянуті деякі архітектури нейронних мереж.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЕНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Протягом останніх двох десятиліть було розроблено значну кількість фреймворків, які допомагають вирішувати проблеми в галузі машинного навчання. Більшість з цих фреймворків доступні у відкритому вигляді та можуть бути використані як для наукових досліджень, так і для комерційних проектів. Ці бібліотеки для машинного навчання розроблені таким чином, щоб розробники могли вирішувати завдання, не заглиблюючись в найскладніші математичні та структурні аспекти моделей, а використовувати й розширювати готові рішення, надані фреймворками.

В машинному навчанні розрізняють два типи підходів. Перший – навчання на основі прецедентів або індуктивне навчання, засноване на виявленні загальних закономірностей на основі конкретних емпіричних даних. Другий тип – дедуктивне навчання, яке передбачає формалізацію знань експертів та їх перенесення до комп'ютера у вигляді бази знань. Дедуктивне навчання вважається складовою експертних систем.

Метод навчання на основі прецедентів, або індуктивне навчання, поділяється на три основних типи: контрольоване навчання (supervised learning), неконтрольоване навчання (unsupervised learning) та навчання з підкріпленням (reinforcement learning). Крім цих підходів, розробляються інші методи навчання, такі як активне, багатозадачне, різноманітне, трансферне тощо. Особливо успішно розвивається глибоке навчання.

Система навчання алгоритму машинного навчання складається з трьох основних компонентів: процесу прийняття рішень, функції помилки та процесу оптимізації моделі. Алгоритми машинного навчання використовуються для прогнозування або класифікації на основі вхідних даних, а функція помилки оцінює точність моделі. Процес оптимізації моделі полягає в коригуванні ваг

моделі, щоб зменшити розбіжність між відомими прикладами та оцінками моделі. Цей процес повторюється досягнення порогової точності [4].

Для ідентифікації динамічних об'єктів часто використовуються різні автономні бібліотеки з такими особливостями.

Бібліотека `fastai` – це бібліотека глибокого навчання з функціями комп'ютерного зору, написана на Python.

Бібліотека `IPSDK` – це бібліотека для обробки 2D і 3D зображень, доступна для мов програмування C++ і Python.

Бібліотека `Imutils` – це надбудова над `OpenCV`, яка надає розширені функції комп'ютерного зору. Вона також доступна для використання в C++ і Python [5].

Бібліотека `Keras` – це високорівнева бібліотека нейронних мереж, яка має підтримку згорткових та рекурентних мереж для розпізнавання зображень. Вона реалізована на мові програмування Python [6].

Бібліотека `OpenCV` – це бібліотека комп'ютерного зору, яка спрямована на реальні часи програми і аналіз відео. Вона написана на мові програмування C++ та інші.

`PyTorchCV` – це фреймворк, заснований на `PyTorch`, призначений для розв'язання завдань машинного зору, таких як класифікація зображень, сегментація, виявлення та оцінка пози. Він містить класичні моделі, такі як `AlexNet`, `ResNet`, `ResNeXt`, `PyramidNet`, `SparseNet`, `DRN-C/DRN-D` і багато інших. Цей фреймворк доступний на мові програмування Python.

Бібліотека `Scikit-Image` – це бібліотека для обробки зображень, яка є розширенням для бібліотеки `SciPy`. Вона реалізована на мові програмування Python.

Бібліотека `SimpleCV` – це бібліотека машинного зору, яка надає зручний інтерфейс для взаємодії з камерою, маніпулювання зображеннями та відеопотоками. Вона також написана на мові програмування Python.

Всі ці бібліотеки мають спільні риси, такі як вільні ліцензії, використання мови програмування Python для швидкого прототипування, підтримку нейронних мереж для розпізнавання та можливість використання веб-платформ

для розгортання. Проте використання веб-платформи та опрацювання даних у хмарі можуть бути платними опціями, які залежать від розробника (рис. 2.1).

Бібліотека	Мова
Tensorflow	Python/Java/JS
Keras	Python
DL4J	Java
Torch	Python
Theano	Python

Рисунок 2.1 – Розподіл алгоритмів навчання мереж

Інформаційна (математична) модель або алгоритм роботи

Розпізнавання обличч – це практичне використання теорії розпізнавання образів, яке включає автоматичне визначення особи на фотографії та, за необхідності, ідентифікацію особи за допомогою обличчя.

Задача ідентифікації та розпізнавання осіб є однією з перших практичних завдань, яка сприяла розвитку теорії розпізнавання та ідентифікації об'єктів. Існує дев'ять категорій об'єктів, які відповідають розпізнавальним областям та викликають зорові уявлення.

Цікавість до процедур, що лежать в основі процесу впізнавання та розпізнавання осіб, завжди була значною, особливо в контексті зростаючих практичних потреб, таких як системи безпеки, верифікація, криміналістична експертиза, телеконференції та інші. Незважаючи на очевидність факту, що людина здатна добре ідентифікувати обличчя, процес навчання комп'ютерів здійснювати цю процедуру, включаючи декодування та збереження цифрових зображень осіб, є неочевидним. Ще складніше є оцінити схожість між особами,

включаючи їх комплексну обробку. У проблемі розпізнавання осіб можна виділити кілька напрямків дослідження:

- інформаційно-процесуальні моделі;
- комп'ютерні моделі розпізнавання;
- нейрофізіологічні моделі;
- нейропсихологічні моделі.

Задача розпізнавання осіб вивчалася ще на ранніх етапах розвитку комп'ютерного зору. Багато компаній протягом понад 40 років активно розробляють автоматизовані, а нині й автоматичні системи розпізнавання облич: ImageWare (система FaceID), Spillman, Miros (система Trueface), Imagis, Epic Solutions, Visionics (система FaceIt), Vissage Technology (система Vissage Gallery), Smith & Wesson (ASID – Automated Suspect Identification System).

Технології розпізнавання осіб дозволяють автоматично шукати та розпізнавати особи у графічних файлах та відеопотоках.

Розглянемо, які завдання вирішує технологія розпізнавання облич («Face recognition») і як вона працює.

Системи розпізнавання облич дозволяють вирішувати різноманітні завдання, як прості, так і складні, включаючи:

- підтвердження та ідентифікація особи під час онлайн-іспитів студентів;
- розпізнавання людей у громадських місцях, що перебувають у «чорному списку»;
- здійснення оплати різних товарів та послуг;
- використання в сучасних парках для збереження місця в черзі до атракціонів;
- розблокування мобільного телефону.

У минулому системи розпізнавання облич використовувалися переважно правоохоронними органами для пошуку підозрілих осіб у натовпі. Деякі урядові установи також використовували ці системи для забезпечення безпеки та протидії шахрайству під час виборів.

Однак існує багато інших ситуацій, в яких така технологія стає популярною. Завдяки зниженню вартості і поширенню систем розпізнавання обличь, вони стають сумісними з камерами та комп'ютерами, які використовуються в банках та аеропортах. Туристичні агентства впроваджують програми «бувалих мандрівників», які дозволяють швидко перевіряти безпеку пасажирів, які добровільно надають свою інформацію. Це допомагає прискорити процес проходження через аеропорт, оскільки система розпізнавання обличь порівнює особи з внутрішньою базою даних.

Інші потенційні застосування включають використання систем розпізнавання обличь у банкоматах та терміналах видачі готівки. Програмне забезпечення може швидко перевірити особу клієнта, а після його підтвердження зняти знімок обличчя. Це забезпечує захист клієнтів від крадіжок особистих даних та шахрайських транзакцій, оскільки банкомат не надасть гроші неправомірній особі.

Для ефективного розпізнавання облич при використанні 2D-зображень необхідно зйомка з фронтального ракурсу та при достатньому освітленні, що зазвичай використовується у охоронних системах на підприємствах та держустановах (рис. 2.2). У випадку роботи в громадських місцях використовуються 3D-зображення. За допомогою синхронізованих камер з різних ракурсів робиться серія фотознімків, на основі яких формується тривимірна модель об'єкта. Система працює, визначаючи контрольні точки, і порівнює отримані дані зі збереженими в базі даних. При збігу параметрів, особа ідентифікується.

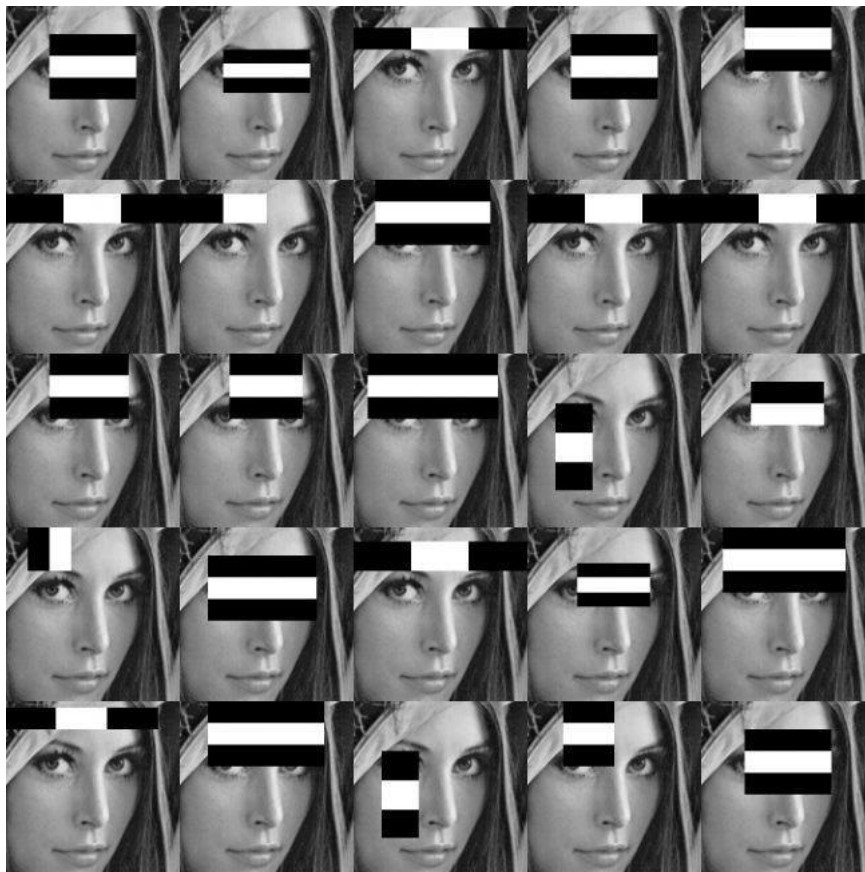


Рисунок 2.2 – Як розпізнається 2D зображення

Розпізнавання облич – це математично обґрунтований процес, який можна описати у контексті використання камери. Для алгоритму розпізнавання облич був обраний метод "Каскади Хаара", який раніше використовувався у старих фотоапаратах.

Давайте пояснимо, як розпізнати обличчя на фотографії. Для цього потрібно виділити основні компоненти обличчя, такі як ніс, лоб, очі, губи та інші. Для цієї задачі використовуються шаблони, які також називаються примітивами Хаара (рис. 2.3).

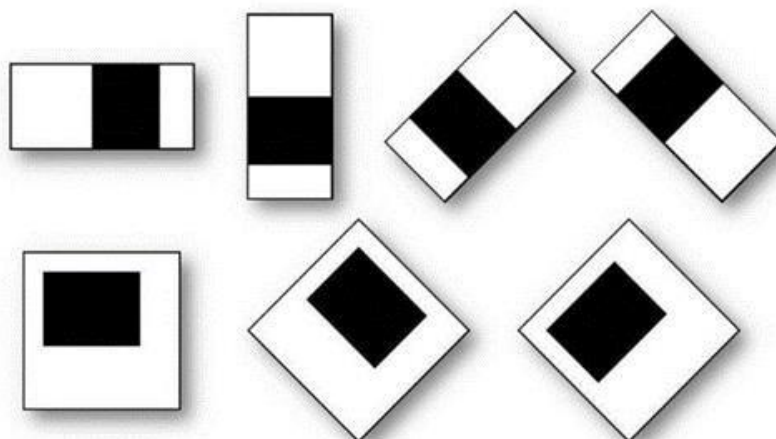


Рисунок 2.3 – Примітиви Хаара

Якщо на зображенні виявляються області, які відповідають шаблонам, ми вважаємо, що на зображенні присутнє людське обличчя. Для кожного шаблону порівнюється різниця між яскравістю білих і чорних областей. Це значення порівнюється з еталоном, і приймається рішення про наявність частини людського обличчя.

Цей метод відомий як метод Віоли-Джонса, або каскад Хаара. Припустимо, що на фотографії міститься не одне велике обличчя, а кілька менших. Якщо застосувати шаблони до всього зображення, ми не знайдемо всіх осіб, оскільки вони будуть меншого розміру, ніж шаблони. Для пошуку облич різних розмірів на всіх фотографіях використовується метод ковзного вікна. Це вікно переміщується по всьому зображенню, і для кожного проходження в ньому обчислюються примітиви. Після кожного проходження зображення вікно збільшується, щоб знайти особу більшого масштабу.

Тепер розглянемо роботу класифікатора, який визначає, яка саме особа знаходиться перед камерою. Для формування ознак моделі використовується каскад Хаара. Ці ознаки обробляються за допомогою алгоритму градієнтного бустінгу. Метод найшвидшого бустінгу використовується для класифікації людей, тоді як каскад Хаара використовується для пошуку облич на фотографіях.

Перед тим, як описувати «градієнтний бустінг», необхідно розглянути більш простий алгоритм, який буде використовуватися як базовий (градієнтний

бустінг об'єднує кілька базових алгоритмів у композицію). Зазвичай в якості базового алгоритму використовується "рішуче дерево". Рішуче дерево є бінарним деревом (у загальному випадку), де вузли містять прості предикати, а листки містять прогнози (рис. 2.4).

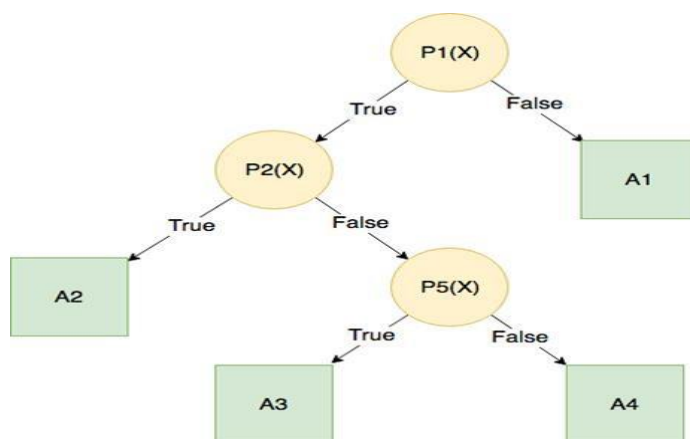


Рисунок 2.4 – Вирішальне дерево

Кожне рішуче дерево будується автоматично. На кожній ітерації алгоритм вибирає такий предикат, який максимально добре розділяє класи. Проще кажучи, якщо ніс однієї людини суттєво відрізняється від носа іншої людини, алгоритм найімовірніше використовуватиме цю ознаку для першого розбиття. Подальше виконання алгоритму полягатиме в пошуку менш очевидних залежностей.

Тепер необхідно описати один зі способів об'єднання дерев у композицію – метод градієнтного бустінгу. Цей метод спрямований на послідовну побудову композиції. Іншими словами, ми не будемо одну модель, а кілька моделей разом. Будівництво відбувається наступним чином: спочатку будується перша модель, обчислюється її помилка, потім будується друга модель таким чином, щоб вона виправляла помилки попередньої моделі. Таким чином, ми отримуємо якісну класифікацію, оскільки кожна нова модель у композиції виправляє помилки попередніх моделей.

Давайте розглянемо наочний процес виправлення помилки одним алгоритмом за допомогою спрямованої побудови композиції. Наведений

приклад, який буде розглядатися, не стосується нашої конкретної задачі, але його зручно використовувати для демонстрації роботи алгоритму.

На рисунку 2.5 показаний процес виправлення помилки за допомогою спрямованої побудови композиції. Перший графік представляє справжню залежність. Другий графік (tree 1) відображає перший алгоритм, який намагається захопити цю залежність. Третій графік показує алгоритм, який намагається виправити помилку попереднього. Четвертий графік відображає алгоритм, який намагається виправити помилку двох попередніх алгоритмів.

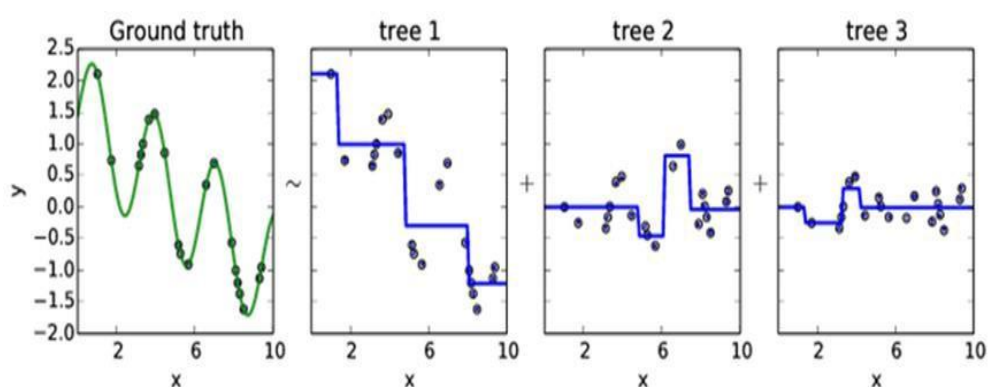


Рисунок 2.5 – Результат градієнтного бустінгу

Проведений аналіз подібних програмних рішень дає можливість зробити деякі висновки про те, які функціональні можливості повинні бути у програмному продукті.

Методи, засновані на значеннях яскравості пікселів.

Назва даної групи методів вказує на те, що для розпізнавання виявлених обличч використовується лише колір або яскравість пікселів на зображеннях. Простий метод порівняння є одним з подібних підходів, де відстань між векторами яскравості пікселів зображень служить мірою схожості. Однак цей метод не є надійним у випадку змін освітленості, положення обличчя та масштабування. Крім того, такий підхід вимагає значних обчислювальних ресурсів і не є придатним для реального часу. Тому часто використовуються методи, які перетворюють векторний опис зображень в простір з меншою розмірністю, що дозволяє більш ефективно порівнювати обличчя [7].

Один з таких методів – це алгоритм Eigenfaces, запропонований Метью Терком і Алексом Пентландом у 1991 році, який став першим успішним методом розпізнавання осіб. Основна ідея цього алгоритму полягає у застосуванні методу головних компонентів для знаходження векторів, які найкраще описують зображення облич. Використовуючи цей метод, можна виявити різні зміни в навчальній вибірці зображень облич та описати ці зміни у просторі, що складається з декількох ортогональних векторів, відомих як власні вектори (eigenfaces). Обчислення головних компонентів зводиться до обчислення власних векторів та власних значень коваріантної матриці, яка обчислюється зі зображення. Отриманий набір власних векторів, який формується один раз на навчальній вибірці зображень облич, використовується для кодування всіх інших зображень осіб, які представляються як зважена комбінація цих власних векторів. Шляхом обмеження кількості використовуваних власних векторів можна отримати компактну апроксимацію вхідного зображення особи, яку потім можна зберігати в базі даних у вигляді вектору коефіцієнтів, що використовується як ключ для пошуку у базі даних осіб. Сума головних компонентів, помножених на відповідні власні вектори, представляє реконструкцію зображення. Для кожного зображення обличчя обчислюються його головні компоненти, зазвичай від 5 до 200 власних векторів.

Процес розпізнавання ґрунтується на порівнянні головних компонентів невідомого зображення з компонентами всіх відомих зображень. При цьому припускається, що зображення облич, які належать одній людині, групуються у власному просторі. З бази даних вибираються зображення-кандидати, які мають найменшу відстань до вхідного зображення. Метод власних облич потребує ідеалізації умов, таких як однакові параметри освітленості, нейтральний вираз обличчя та відсутність завад, таких як окуляри і борода. При невиконанні цих умов головні компоненти не відображатимуть міжрасові варіації.

Наприклад, при різних умовах освітленості метод власних облич майже не застосовується, оскільки перші головні компоненти переважно відображають зміни освітленості, і порівняння видає зображення, яке має схожий рівень

освітленості. При дотриманні ідеалізованих умов точності розпізнавання з використанням методу досягає більше 90%. Обчислення набору власних векторів є трудомістким процесом. Один зі способів полегшити цей процес – розбити зображення на стрічки та стовпчики, що дозволяє представити його у меншому розмірі. Обчислення та розпізнавання зображення в такому форматі відбувається швидше, але неможливо відновити початкове зображення. Метод головних компонентів успішно використовується в практичних застосуваннях. Проте, коли на зображенні обличчя присутні значні зміни в освітленості або виразі, ефективність методу значно знижується. Ідея методу полягає в представленні зображення обличчя у вигляді набору векторів головних компонентів зображень.

Переваги:

- легкість реалізації;
- придатність для розпізнавання в реальному часі;
- можливість компактного зберігання великих обсягів даних;
- швидке розпізнавання.

Недоліки:

- висока трудомісткість обчислення набору власних векторів;
- непридатність для розпізнавання облич, які мають перешкоди, такі як окуляри або борода;
- необхідність якісної попередньої обробки зображення для приведення його до стандартних умов;
- вимога до зображень бути отриманими в близьких умовах освітленості та з однаковим ракурсом;
- потреба у навчанні один раз, а при додаванні нових облич до бази даних потрібно будувати простір з власних векторів заново.

Недоліки викликані тим, що найбільш важливі власні вектори більше описують особливості освітлення, ніж характеристики облич, оскільки метод головних компонентів спочатку прагне апроксимувати дані, а не класифікувати їх.

Fisherfaces – це алгоритм, який використовує лінійний дискримінантний аналіз, зокрема лінійний дискримінант Фішера, у відмінність від методу eigenfaces. Дія алгоритму полягає в пошуку проекції даних, яка максимально розділяє класи зображень облич. Використання методу головних компонент веде до максимізації розкиду даних по всій базі облич.

Ця відмінність допомагає вирішити проблему високої чутливості до змін освітлення. Метод головних компонент, що лежить в основі методу eigenfaces, шукає лінійну комбінацію ознак, яка максимізує загальну дисперсію даних. Це потужний спосіб представлення даних, але він не враховує класи об'єктів і може призвести до втрати значної інформації через відкидання компонент, які не враховуються у базисі.

Наприклад, коли база містить фотографії з різним освітленням, метод головних компонент може змішувати зразки різних класів, оскільки він не завжди включає всю відмінну інформацію. Це ускладнює подальшу класифікацію і робить її неможливою.

Цю проблему вирішує алгоритм лінійного дискримінантного аналізу, який є основою методу Fisherfaces. Лінійний дискримінантний аналіз – метод статистики машинного навчання, який застосовується для пошуку лінійних комбінацій ознак, які найкращим чином розділяють два чи більше класи об'єктів чи подій (тобто об'єкти одного і того ж класу мають шукатись, як можна ближче один до одного в просторі, і при цьому відбувається максимізація відстані між класами. Даний алгоритм краще розпізнає при різному освітленні, чим EigenFaces, тільки навчання має відбуватись також при різних умовах освітленості.

Переваги: завдяки використанню набору фотографій кожної особи алгоритм стає стійким до змін умов освітлення.

Недоліки: відсутність інваріантності щодо афінних перетворень.

Локальні бінарні шаблони (LBP – Local Binary Pattern) – це простий і ефективний оператор перетворення зображень, який вперше був запропонований в 1996 році для класифікації текстур, а згодом знайшов застосування в

розпізнаванні осіб. Цей оператор використовує значення яскравості околиці кожного пікселя зображення і за допомогою функції надає кожному пікселю значення, що описує його околицю. Потім отримане зображення розділяється на підобласті, для кожної з яких розраховується гістограма. Гістограми об'єднуються і порівнюються за допомогою методів машинного навчання, при класичному варіанті використовується метод найближчого сусіда.

Переваги цього методу полягають у простоті реалізації і високій швидкості роботи, яку можна збільшити за допомогою різних модифікацій алгоритму. При цьому алгоритм показує високі результати при розпізнаванні облич і є стійким до монотонних змін освітлення. Все це робить його оптимальним для розпізнавання облич у системах обробки в реальному часі.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Для вибору мови програмування для нашого проекту ми розглядаємо Python і C++. Основні характеристики, переваги та особливості цих мов такі.

Python – це мова програмування високого рівня, яка здатна працювати в різних парадигмах програмування. Вона підтримує об'єктно-орієнтовану, узагальнену та процедурну парадигми.

C++ – це мова програмування високого рівня, яка також підтримує кілька парадигм програмування, включаючи об'єктно-орієнтовану, узагальнену та процедурну парадигми.

Одна з груп, що активно працює над розвитком мови C++, це Boost. Вони надають поради щодо поліпшення мови та додають до неї особливості метапрограмування.

Стандарт C++ не визначає способи іменування об'єктів, деталі обробки винятків та інші можливості, які пов'язані з реалізацією мови. Це може створювати несумісність між об'єктним кодом, розробленим за допомогою

різних компіляторів. Проте існує безліч стандартів, розроблених третіми особами для конкретних архітектур і операційних систем.

Одним з викликів у реалізації стандарту C++ є ключове слово "export", яке використовується для розділення оголошення та визначення шаблонів.

C++ була створена з метою забезпечити сумісність з мовою C, тому більшість програм, написаних на C, можуть працювати без змін з використанням компілятора C++. C++ має синтаксис, який базується на синтаксисі C.

Стандартна бібліотека C++ включає стандартну бібліотеку C з невеликими змінами, що роблять її більш підходящою для використання в C++. Також, велика частина бібліотеки C++ ґрунтується на Стандартній Бібліотеці Шаблонів (STL), яка надає важливі інструменти, такі як контейнери (наприклад, вектори і списки) та ітератори, що дозволяють доступатися до цих контейнерів як до масивів. Крім того, STL дозволяє працювати з іншими типами контейнерів, такими як асоціативні списки, стеки та черги.

За допомогою шаблонів можна створювати узагальнені алгоритми, які працюють з будь-якими контейнерами або послідовностями, доступ до елементів яких забезпечують ітератори.

У C++, подібно до C, можна активувати функціонал бібліотек, включивши стандартні файли за допомогою директиви `#include`. В стандарті C++ визначено 50 таких файлів.

C++ додає до C об'єктно-орієнтовані можливості, введення класів, які забезпечують інкапсуляцію, успадкування і поліморфізм – три найважливіші властивості ООП.

Python – це мова програмування високого рівня зі строгим динамічним типізацією та об'єктно-орієнтованою парадигмою. Вона володіє численними перевагами, які роблять її привабливою для швидкої розробки програм та поєднання компонентів. Python має високорівневі структури даних, динамічну семантику та динамічне зв'язування, що сприяє зручності програмування.

Однією з переваг Python є його чистий синтаксис, де блоки коду виділяються за допомогою відступів. Він також переносний і доступний на

багатьох платформах. Python має широкий набір корисних модулів у своєму стандартному дистрибутиві, включаючи модуль для розробки графічних інтерфейсів. Python надає можливість використовувати його в діалоговому режимі для експериментів та простих задач. Також він має просте, але потужне середовище розробки IDLE, яке є частиною стандартного дистрибутиву і написано саме на Python. Ще одна перевага полягає у його зручності для вирішення математичних проблем, включаючи роботу з комплексними числами та можливість оперувати цілими числами довільної величини.

Python пропонує потужні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Його елегантний синтаксис, динамічна обробка типів та інтерпретована природа роблять його ідеальним вибором для написання скриптів та швидкої розробки прикладних програм у різних галузях на більшості платформ.

Інтерпретатор мови Python та його багата Стандартна бібліотека, як у вихідних текстах, так і у бінарних дистрибутивах для різних операційних систем, можуть бути завантажені з веб-сайту Python (www.python.org) та розповсюджуватися безкоштовно. Цей веб-сайт також надає дистрибутиви та посилання на численні модулі, програми, утиліти та додаткову документацію.

Зважаючи на всі особливості та потреби майбутнього програмного продукту, Python було обрано для реалізації, оскільки він відповідає всім вимогам та забезпечує необхідну функціональність.

Висновки до розділу 2

Здійснений огляд підходів виявлення і відстеження людей на відео, аналіз додатків машинного навчання, а також розглянуті деякі архітектури нейронних мереж. Проаналізовані існуючі інструменти, що використовуються для розробки такої системи та досліджені алгоритми і методи навчання нейронних мереж; розглянуті та обрані алгоритми оптимізації та функції втрат для досягнення найкращих результатів.

РОЗДІЛ 3

РОЗРОБКА ОБ'ЄКТА ПРОЕКТУВАННЯ

3.1 Особливості використання мов програмування, баз даних, особливості технологій, програмного забезпечення

Для початку треба визначити, як розпізнати обличчя на фотографії. Перш за все, ми повинні знайти місцезнаходження особи на зображенні і уникнути сплутування її з годинником на стіні або кактусом на підвіконні. Здається, що це просте завдання для людини, але виявляється складним для комп'ютера. Для досягнення цієї мети ми повинні виділити основні компоненти обличчя, такі як ніс, лоб, очі, губи. Для цього ми будемо використовувати шаблони, які зображені на рисунку 3.1.

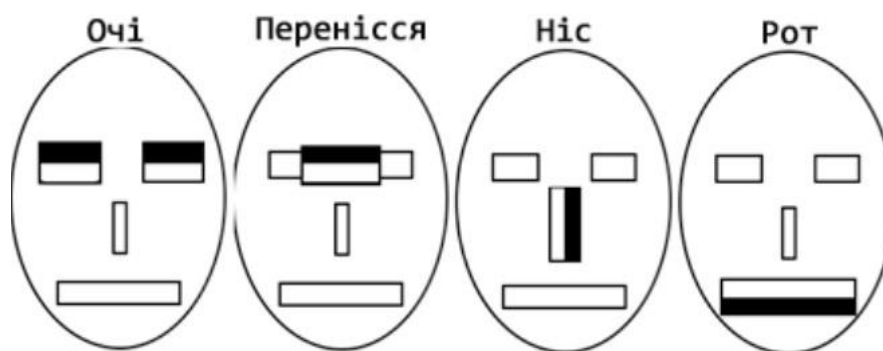


Рисунок 3.1 – Основні шаблони

Давайте розглянемо ознаки Хаара і їх різновиди. Почнемо з прямокутних ознак Хаара. Найпростіша прямокутна ознака Хаара може бути визначена як різниця суми пікселів двох суміжних областей всередині прямокутника, який може займати різні положення і масштаби на зображенні. Такий тип ознак відомий як 2-прямокутна ознака. Віола та Джонс також визначили 3-прямокутні та 4-прямокутні ознаки. Кожна ознака може виявити наявність або відсутність певної характеристики на зображенні, такої як границя або зміна текстури. Наприклад, 2-прямокутна ознака може вказати на межу між темним і світлим регіонами [8].

Також існують похилі ознаки Хаара. Лінхарт і Майд запропонували ідею використання нахилених під кутом 45 градусів ознак Хаара для збільшення розмірності простору ознак. Цей підхід виявився успішним, і деякі похилі ознаки здатні краще описувати об'єкти. Наприклад, 2-прямокутна похила ознака Хаара може показати наявність нахиленої під кутом 45 градусів краю. Мессі та Барзаков додали нові ідеї до концепції похилих ознак Хаара. Незважаючи на математичну правильність цієї ідеї, на практиці виникають проблеми при використанні ознак під різними кутами через помилки округлення, пов'язані з використанням зображень низького розширення для прискорення обчислень. Через ці обмеження похилі ознаки Хаара зазвичай не використовуються.

У методі Віоли-Джонса основою є примітиви Хаара, які є необхідною складовою цього методу і визначають його суть. Примітиви Хаара представляють собою розбиття заданої прямокутної області на набори прямокутних підобластей різних типів, як показано на рисунку 3.2. Ці примітиви є базовою структурою, від якої виходять усі інші аспекти методу.

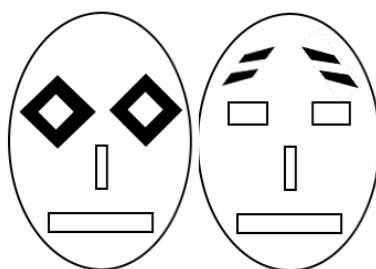


Рисунок 3.2 – Шаблони з поворотом

У початковій версії алгоритму Віоли-Джонса використовувалися лише примітиви без поворотів, і для обчислення значення ознаки використовувалася різниця між сумою яскравостей пікселів однієї підобласті та сумою яскравостей пікселів іншої підобласті. Згодом метод був удосконалений шляхом введення примітивів з нахилом на 45 градусів і несиметричних конфігурацій. Крім того, замість обчислення звичайної різниці, було запропоновано призначити кожній підобласті певне вагове значення, а значення ознаки обчислювати як зважену суму пікселів з різних типів областей, використовуючи відповідну формулу:

$$feature = \sum_{i \in I=1, \dots, N} w_i \cdot RectSum(r_i)$$

Складність обчислення ознаки, так само як і отримання значення пікселя, залишається $O(1)$. Значення кожної підобласті можна обчислити шляхом комбінування 4 значень інтегрального представлення (Summed Area Table – SAT), яке можна побудувати заздалегідь один раз для всього зображення за час $O(n)$, де n – кількість пікселів у зображенні. Це можна зробити з використанням формули:

$$SAT(-1, y) = SAT(x, -1) = SAT(-1, -1) = 0$$

$$SAT(x, y) = SAT(x, y-1) + SAT(x-1, y) + I(x, y) - SAT(x-1, y-1)$$

Це дало можливість створити швидкий алгоритм пошуку об'єктів, який успішно використовується вже протягом багатьох років. Але давайте повернемося до наших ознак. Для визначення приналежності об'єкта до певного класу в кожному каскаді, сумуються значення слабких класифікаторів, що належать до цього каскаду. Кожен слабкий класифікатор видає два значення залежно від того, чи перевищує значення ознаки заданий поріг, який встановлений для цього класифікатора. В кінці, сума значень слабких класифікаторів порівнюється з порогом каскаду, і приймається рішення, чи знайдений об'єкт з використанням цього каскаду, чи ні.

Використання методу Хаара

У використаному методі Віоли-Джонса, відомому також як каскади Хаара, шаблони або примітиви відповідають конкретним областям на зображенні, і ми припускаємо, що на зображенні присутнє людське обличчя. Для кожного обличчя вимірюється різниця між яскравістю білих і чорних областей. Це значення порівнюється з еталоном, і приймається рішення, чи є на цій ділянці обличчя чи ні.

Давайте уявимо, що на фотографії присутні не одне велике обличчя, а кілька менших. Якщо застосувати шаблони до всього зображення і не знайти на ньому жодного обличчя, то обличчя меншого розміру будуть мати менше шаблонів. Для того, щоб шукати обличчя різних розмірів на всіх фотографіях, використовується метод ковзного вікна. Вікно, так би мовити, ковзає по всьому

зображенню. Після кожного проходження зображення, вікно збільшується, щоб знайти обличчя більшого масштабу.

Якщо на фотографії знайдено обличчя, для того, щоб визначити конкретну людину, необхідно виконати кілька додаткових кроків. Для цього ми використовуємо алгоритм локальних бінарних шаблонів (Local Binary Patterns). Його суть полягає в тому, що ми розбиваємо зображення на частини, і в кожній частині кожен піксель порівнюється зі своїми восьмима сусідніми пікселями. Якщо значення центрального пікселя більше за значення сусіднього, ми записуємо 0, в іншому випадку - 1. Таким чином, для кожного пікселя отримуємо послідовність бінарних чисел. Далі, на основі цих чисел для всіх частин, на які розбито фотографію, обчислюється гістограма. Усі гістограми з областей об'єднуються в один вектор, який характеризує зображення в цілому. Якщо ми хочемо порівняти два обличчя на схожість, нам потрібно обчислити такий вектор для кожного з них і порівняти їх. Процес обчислення вектора показано на рисунку 3.3.

5	8	1		1	1	0
3	4	6		0		1
7	1	3		1	0	0

Рисунок 3.3 – Розрахунок важелів LBP

Отже, були розглянуті алгоритми розпізнавання обличчя за ознаками Хаара та методом Віолі-Джонса, який базується на пошуку незмінних ознаках обличчя, таких як позиція рота, носу, очей, тощо. та будуванні матричної сітки для задання значення важелів. За допомогою порівняння основних ознак, можна з великою точністю порівняти обличчя із двох різних фотографій, та отримати ймовірність тотожності.

Також цей спосіб не залежить від віку людини, тому порівняння однієї людини із самою собою з різницею 10 років буде давати результат аналогічної точності.

3.2 Структура автоматизованої системи (підсистеми), бази даних, програми

Ми обрали використання бібліотеки OpenCV з кількох причин. По-перше, ця бібліотека є вільно поширюваною. По-друге, окрім розпізнавання осіб, вона надає додаткові корисні функції для налагодження, такі як промальовування вікон та геометричних примітивів навколо осіб.

Важливо відзначити, що класифікатори, що вбудовані в OpenCV, добре оптимізовані для роботи з неякісними даними, такими як погане освітлення та інші шуми в даних. Тому реалізація подібної бібліотеки з нуля не є доцільною, оскільки OpenCV є довготривалим та широко налагодженим проектом. Витрачати додатковий час та ресурси на написання аналогічної бібліотеки не має сенсу, коли в OpenCV вже реалізовані всі необхідні функції.

Наведемо короткий опис бібліотеки OpenCV та функцій та класів, які були використані.

OpenCV – це бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом [5].

Нижче наведено функції та класи з бібліотеки, які були використані.

Додавання нового обличчя.

Для захоплення зображення з камери використовувався екземпляр класу VideoCapture.

Для виявлення осіб на захопленому зображенні використовувався метод detectMultiScale екземпляру класу CascadeClassifier [9].

Для тренування класифікатора використовувався метод train екземпляру класу LBPHFaceRecognizer [10].

Для промальовування геометричних примітивів навколо осіб під час налагодження використовувалась функція `rectangle`.

Для звільнення камери була використана функція `release`. Розпізнавання відомих / невідомих осіб.

Для використання класифікатора були використані методи `read` і `predict` з екземпляра класу `LBPFaceRecognizer`.

Було також необхідно вирішити проблему доступу до камери. Перед записом нового обличчя в класифікатор, потрібно звільнити камеру, оскільки вона постійно працює в режимі спостереження. Цю проблему було вирішено наступним чином: був створений окремий модуль для постійної роботи камери (який запускається паралельно з ботом), і якщо потрібен доступ до камери з іншої функції, паралельно запущений скрипт призупиняє свою роботу. Після того, як всі функції, які запитують доступ до камери, виконались, скрипт постійного спостереження запускається знову.

Під час роботи програми всі дії з камерою фіксуються у консолі. Це робиться для відстеження коректності роботи програми. Кожен лог починається з ключового префікса [INFO], за яким слідує текст повідомлення.

Ось список лог-повідомлень:

- при запуску постійної роботи камери: `Camera handler subprocess is running;`
- при призупиненні постійної роботи камери: `Camera handler subprocess terminating for a while.`

Також є лог-повідомлення, які повідомляють про додавання/видалення нового обличчя і про фіксацію камерою присутності будь-якої людини.

3.3 Структурно-функціональна схема системи

У розроблюваній програмі використано багатоварову нейронну мережу. Блок-схема її роботи зображена на рисунку 3.4.

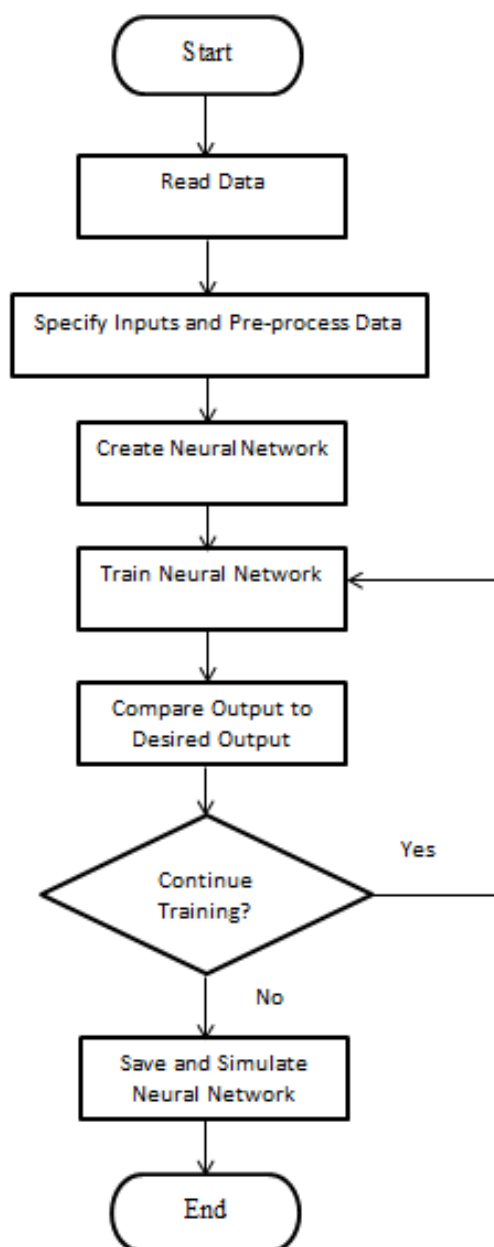


Рисунок 3.4 – Блок-схема реалізації багатошарової нейронної мережі

3.4 Опис розробленого програмного (інформаційного) забезпечення обчислювальних систем

Опис вхідних даних. Вхідними даними є 690 зображень з обличчями людей без лицьової маски, та 686 зображень з обличчями людей з лицьовою маскою, які були взяті з публічного проекту на сайті GitHub (рис. 3.5).

Для застосування алгоритмів комп'ютерного зору, в даних мають бути присутні проблеми:

- наявність зображень однієї і тієї самої людини за накладанням на нього додаткового шуму або розмиття;
- наявність зображень, які відзеркаленні по одній із осей;
- кожне зображення може мати різну роздільну здатність;
- наявність однакових зображень, які мають різні обертання.



Рисунок 3.5 – Приклади особливостей вхідних даних

Розділення даних для тренування. На цьому етапі йде розділення даних на навчальну множину, що містить зображення, на яких навчатиметься модель ЗНМ, і тестова множина із зображеннями, на яких тестуватиметься наша модель.

При цьому розроблена функція `train_test_split` буде приймати значення `split_size 0.8`, що означає, що 80% всіх зображень потраплять в навчальну множину, а 20% зображень, що залишилися – в тестову множину (лістинг 3.1).

Лістинг 3.1 – Налаштування підключення до бази даних

```

The number of images with facemask: 690
The number of images without facemask: 686
train images with mask: 552
test images without mask: 138
train images with mask: 548
test images without mask: 138
  
```

кінець лістингу 3.1

Після розбиття ми бачимо, що необхідний відсоток зображень був розподілений як на навчальну, так і на безліч тестів, як згадувалося вище.

Побудова моделі. Будується модель послідовної ЗНМ із різними шарами.

Перший шар – це шар Conv2D (цей шар створює ядро згортки, яке згортається із входом шару для отримання виходів) зі 100 фільтрами та розміром фільтра або ядра 3X3. На цьому першому етапі використовується функція активації "ReLU". Ця функція ReLU означає лінійну одиницю, яка виводить вхідні дані якщо вони позитивні, інакше виводиться нуль. Розмір вхідних даних також ініціалізується як 150X150X3 для всіх зображень, які будуть навчені та протестовані за допомогою цієї моделі.

Потім використовується MaxPooling2D, максимальне об'єднання для зменшення просторових розмірів вихідного обсягу.

Наступний шар – знову шар Conv2D з ще 100 фільтрами того ж розміру 3X3, а як функція активації використовується «ReLU». За цим шаром Conv2D слід шар MaxPooling3=2D з розміром блоку 2X2.

На наступному етапі використовується шар Flatten(), щоб перетворити всі шари в один шар 1D.

Після шару Flatten використовується шар Dropout (0.5), щоб запобігти переоцінці моделі.

Наступний буде використовуватись щільний шар (це шар нейронної мережі, де кожен нейрон щільного шару отримує вхід від усіх нейронів попереднього шару, та генерує щільний шар на вихідний сигнал, який являє собою вектор) із 50 одиницями та функцією активації «ReLU».

Останній шар моделі буде ще одним щільним шаром, з двома одиницями, а функція активації, що використовується, буде функцією «Softmax». Функція Softmax виводить вектор, який є розподілом ймовірностей кожної з вхідних одиниць. Тут використовуються дві вхідні одиниці. Функція Softmax виводить вектор із двома значеннями розподілу ймовірностей (лістинг 3.2).

Після побудови моделі, вона компілюється та визначається функція втрат та оптимізатор. У цій моделі був обраний оптимізатор Адам та бінарна перехресна ентропія як функцію втрат для навчання.

Лістинг 3.2 – Налаштування підключення до бази даних

```

model = tf.keras.models.Sequential ([
    Conv2D (100, (3,3), activation='relu', input_shape= (150,
150, 3)),
    MaxPooling2D (2, 2),
    Conv2D (100, (3,3), activation='relu'),
    MaxPooling2D (2, 2),
    Flatten () ,
    Dropout (0.5),
    Dense (50, activation='relu'),
    Dense (2, activation=' softmax') # dense layer has a shape of
2 as we
have only 2 classes
])
model.compile (optimizer= 'adam', loss= 'binary
crossentropy', metrics= ('accuracy' ))

```

кінець лістингу 3.2

Побудовану модель ЗНМ можна графічно побачити на рисунку 3.6.

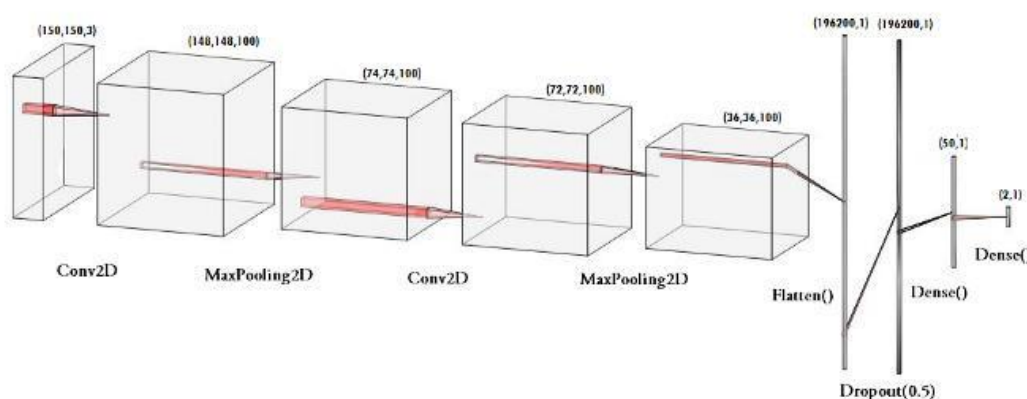


Рисунок 3.6 – Графічний приклад моделі ЗНМ для лицьових масок

Тренування моделі. Навчання моделі за допомогою наступних функцій. Спочатку відкриваються два навчальні потоки з двох каталогів навчальних та тестових (валідаційних) зображень. Потім використовується функція `ImageDataGenerator` з бібліотеки `Keras`, що дозволяє доповнювати зображення в режимі реального часу, поки модель ще вчиться. Ця функція застосовує будь-які довільні перетворення до кожного навчального зображення, коли воно передається в модель.

Доповнення зображення – це техніка застосування різних перетворень до вихідного зображення, у результаті виходить кілька перетворених копій однієї й тієї зображення. Кожна копія, однак, відрізняється від іншої у певних аспектах залежно від застосовуваних технік збільшення, таких як зсув, поворот, перевертання тощо.

Застосування цих невеликих варіацій до вихідного зображення не змінює його цільовий клас, лише дозволяє по-новому поглянути на об'єкт у реальному житті.

Ці методи доповнення зображень не тільки розширюють розмір набору даних, але й включають рівень варіативності, що дозволяє моделі краще узагальнювати дані, що не зустрічаються в полі зору. Крім того, модель стає більш стійкою під час навчання на нових, трохи змінених зображеннях.

Поворот зображення є однією з технік, що широко використовуються, доповнення і дозволяє моделі стати інваріантною до орієнтації об'єкта. В тренуванні зображення буде довільно обертатися в діапазоні від 0 до 40 градусів.

При повороті зображення деякі пікселі виходять за межі і залишають порожню область, яку необхідно заповнити. Її можна заповнити різними способами, наприклад постійним значенням, найближчими значеннями пікселів тощо. Це вказується в аргументі `fill_mode`. Було обрано використовувати значення «nearest», яке просто замінює порожню область найближчими значеннями пікселів.

Може статися так, що об'єкт не завжди знаходиться у центрі зображення. Щоб подолати цю проблему, можна посунути пікселі зображення по горизонталі

або вертикалі, це робиться шляхом додавання певного постійного значення всіх пікселів.

Останнім кроком є підстановка зображень з навчальної та тестової множини до послідовної моделі, яка була побудована за допомогою бібліотеки `keras`. Навчання моделі йде протягом 30 епох (ітерацій). Однак можна навчити модель на більшу кількість епох для досягнення вищої точності.

Під час навчання зберігаються контрольні точки в окремих каталогах кожної контрольної точки, щоб зберегти модель з найкращою точністю та найменшими втратами. Нарешті, викликається функція `fit_generator` (лістинг 3.3) моделі і навчання починається. Під час процесу відстежується точність навчання та перевірки, а також втрати (ці значення використовуватимуть пізніше для побудови кривих навчання).

Лістинг 3.3 – Налаштування підключення до бази даних

```
history = model.fit_generator (train_generator,
                              epochs=epochs,
                              validation_data=validation_generator,
                              callbacks= [checkpoint])
```

кінець лістингу 3.3

Контрольні точки фіксують точне значення всіх параметрів (об'єктів `tf.Variable`), що використовуються моделлю. Контрольні точки не містять жодного опису обчислень, визначених моделлю, і тому зазвичай корисні лише тоді, коли доступний вихідний код, який використовуватиме збережені значення параметрів.

Формат `SavedModel`, з іншого боку, включає серіалізований опис обчислень, визначених моделлю, крім значень параметрів (контрольна точка). Моделі у цьому форматі не залежать від вихідного коду, який створив модель. Таким чином, вони підходять для розгортання через `TensorFlow Serving`,

TensorFlow Lite, TensorFlow.js або програми іншими мовами програмування (API TensorFlow C, C++, Java, Go, Rust, C# і т.д.).

3.5 Інструкція користувача з інсталяції та використання програмного забезпечення

Для запуску програми необхідно встановлене програмне забезпечення, яке дає можливість запуску інтерпретатора Python. Також робота додатку передбачає наявність веб-камери для демонстрації роботи.

Особливості реалізації.

Після побудови моделі ми позначаємо дві можливості для результатів, 0 як «без маски» і 1 як «з маскою». Також встановлено колір граничного прямокутника, використовуючи значення RGB червоний для "без маски" та зелений для "з маскою".

Після цього це буде використатися для визначення того, чи надіта на людині маска, використовуючи веб-камеру комп'ютера. Для цього спочатку необхідно подати виявлення особи. Для цього буде використовуватись каскадні класифікатори з урахуванням ознак Хаара для виявлення рис обличчя.

Цей каскадний класифікатор розроблений бібліотекою OpenCV для виявлення передньої особи шляхом навчання тисяч зображень. Для цього потрібно завантажити файл .xml і використовувати його для виявлення обличчя.

На останньому етапі використовується бібліотека OpenCV для запуску нескінченного циклу для використання веб-камери за комп'ютеру, в якій визначається особа за допомогою каскадного класифікатора.

Модель передбачить можливість кожного з двох класів (без маски, з маскою). На основі того, яка ймовірність вище, буде вибрано позначку, яка відобразатиметься навколо осіб.

Показано кілька фотографій та результати тестування (рис. 3.7). Як можна побачити, виявлено два типи масок на обличчі – лицьова і фронтальна. Але під час усіх тестувань було виявлено багато збоїв. При поганому освітленні кімнати,

модель могла видавати помилкові показання. Якщо особа вдягнена в одяг, який закриває область шиї, модель у 50% випадків видає некоректні результати. Також на відстані особи від камери більш ніж на півтора метри модель зовсім не видає результатів.



Рисунок 3.7 – Результати визначення лицьової маски на обличчі

Є припущення, що набір даних для навчання не забезпечує достатньої кількості даних для усіх можливих сценаріїв, що призводить до неточності навченої моделі та прогнозу. Якщо підвищити кількість даних у декілька разів, та у якості даних брати не лише зображення, а й відео, то модель може потребувати більше ітерацій на навчання, що збільшить витраченого часу на її навчання (при тренуванні з наявними даними та 30 ітераціями знадобилось 20-30 хвилин).

Висновки до розділу 3

Отже, розроблена інформаційна система класифікації облич з використанням методів машинного навчання та досліджений вплив різних факторів на ефективність роботи системи; було використано згорткову нейронну

мережу, бо її архітектура дозволила розробити модель детектування, яка самостійно знаходить у зображенні високорівневі ознаки, використовуючи лише зображення у якості вхідних даних. Для навчання нейронної мережі було знайдено декілька ресурсів із великими обсягами даних осіб людей в масках та без, а також обрано бібліотеки для подальшої реалізації алгоритму.

ВИСНОВКИ

Метою роботи було вирішення задачі розпізнавання лицьових масок на обличчі із використанням нейронних мереж.

Для досягнення поставленої мети, були виконані наступні завдання:

- проведений огляд літературних джерел з даної предметної області. Розглянуто широкий спектр застосувань глибокого навчання, зокрема застосування машинного зору в задачах розпізнавання обличч;
 - здійснений огляд підходів виявлення і відстеження людей на відео, аналіз додатків машинного навчання, а також розглянуті деякі архітектури нейронних мереж;
 - проаналізовані існуючі інструменти, що використовуються для розробки такої системи та досліджені алгоритми і методи навчання нейронних мереж; були розглянуті та обрані алгоритми оптимізації та функції втрат для досягнення найкращих результатів;
 - розроблена інформаційна система класифікації облич з використанням методів машинного навчання та досліджений вплив різних факторів на ефективність роботи системи; було використано згорткову нейронну мережу, бо її архітектура дозволило розробити модель детектування, яка самостійно знаходить у зображенні високорівневі ознаки, використовуючи лише зображення у якості вхідних даних. Для навчання нейронної мережі було знайдено декілька ресурсів із великими обсягами даних осіб людей в масках та без, а також обрано бібліотеки для подальшої реалізації алгоритму.

У кінцевому результаті був створений додаток, який в режимі реального часу розпізнає лицьові маски на обличчі людей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Нильсен М.А. Нейронні мережі та глибоке навчання. Determination Press, 2019. URL: <http://neuralnetworksanddeeplearning.com/> (дата звернення: 13.05.2023).
2. Q. Cao, L. Shen, W. Xie, O. M. Parkhi, A. Zisserman. VGGFace2: A dataset for recognising face across pose and age. International Conference on Automatic Face and Gesture Recognition. URL: <http://www.robots.ox.ac.uk/~vgg/publications/2018/Cao18/cao18.pdf> (дата звернення: 13.05.2023).
3. Object Detection in 20 Years: A Survey. URL: <https://arxiv.org/pdf/1905.05055.pdf> (дата звернення: 13.05.2023).
4. Understanding Loss Functions in Machine Learning. URL: <https://www.section.io/engineering-education/understanding-loss-functions-in-machine-learning/> (дата звернення: 13.05.2023).
5. OpenCV About. URL: <https://opencv.org/about/> (дата звернення: 13.05.2023).
6. About Keras. URL: <https://keras.io/about/> (дата звернення: 13.05.2023).
7. Object Detection Algorithms and Libraries. URL: <https://neptune.ai/blog/object-detection-algorithms-and-libraries> (дата звернення: 13.05.2023).
8. Признаки Хаара. URL: https://ru.wikipedia.org/wiki/Признаки_Хаара (дата звернення: 13.05.2023).
9. ML Practicum: Image Classification. URL: <https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks> (дата звернення: 13.05.2023).
10. CS231n: Convolutional Neural Networks for Visual Recognition. URL: <http://cs231n.github.io/neural-networks-1/> (дата звернення: 13.05.2023).