

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Інформаційна технологія розпізнавання медичних масок на
обличчі на основі згорткової нейромережі»**

Виконав: студент 2-го курсу, групи 2КН-22м
спеціальності 122 «Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Сорока В. А.

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

Паночишин Ю.М.

(прізвище та ініціали)

« 07 » 12 2023 р.

Опонент: к.т.н., професор каф. КСУ

Биков М. М.

(прізвище та ініціали)

« 07 » 12 2023 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

«08» 12 2023 р.

Вінниця ВНТУ - 2023 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та
автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

**Завідувач кафедри КН
Д.т.н., проф. Яровий А.А.**

29.08 2023 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Сороці В'ячеславу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі

керівник роботи к.т.н., доцент кафедри КН Паночишин Ю.М

затверджені наказом вищого навчального закладу від "18" Серпня 2023 року № 247

2. Строк подання студентом роботи 13.11 2023 року

3. Вихідні дані до роботи:

Вхідні дані – формат вхідних зображень - jpg, кількість вхідних нейронів – 2, обсяг навчальної вибірки – не менше 1000, обсяг тестової вибірки – не менше 200, середовище програмування – об'єктно-орієнтоване.



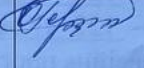
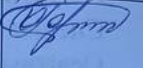
4. Зміст текстової частини:

Вступ, аналіз предметної області розпізнавання медичних масок на обличчі, проектування інформаційної технології розпізнавання медичних масок на обличчі на основі згорткової нейромережі, програмна реалізація інформаційної технології розпізнавання медичних масок на обличчі, тестування та аналіз результатів роботи програми розпізнавання медичних масок на обличчі, висновки, список використаних джерел, додатки

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

Загальний алгоритм роботи програми розпізнавання медичних масок на обличчі, структура інформаційної технології розпізнавання медичних масок на обличчі, UML діаграма послідовностей, результати роботи програми, порівняння показників якості запропонованого програмного забезпечення із показниками програми-аналога.

6. Консультанти розділів роботи

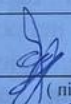
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-4	Паночишин Ю.М., к.т.н., доц. каф. КН		
5	Ратушняк О. Г., к. т. н., доц. каф. ЕПВМ		

7. Дата видачі завдання 29.08 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Пр міт
1	Аналіз сучасного рівня інформаційних технологій розпізнавання об'єктів на зображеннях. Постановка задач дослідження	01.09.23-07.09.23	
2	Побудова моделей розпізнавання медичних масок на обличчі на основі нейронної мережі та функціонування нейронної мережі	08.09.23-20.09.23	
3	Практичне застосування та оцінка ефективності розроблених моделей	21.09.23-29.10.23	
4	Підготовка економічної частини	21.10.23-29.10.23	
5	Апробація результатів дослідження	30.10.23-05.11.23	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	06.11.23-10.11.23	

Студент


(підпис)

Сорока В. А.

Керівник роботи


(підпис)

Паночишин Ю.М.

АНОТАЦІЯ

УДК 004.8

Сорока В. А. Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі. Магістерська кваліфікаційна робота зі спеціальності 122 – Комп'ютерні науки, освітня програма – Системи штучного інтелекту. Вінниця: ВНТУ, 2023. 99 с.

На укр. мові. Бібліогр.: 24 назв; рис.: 25; табл. 10.

У даній магістерській кваліфікаційній роботі розглянуто розробку програмного забезпечення для розпізнавання медичних масок на обличчі на основі згорткової нейромережі. Було розглянуто та проаналізовано відомі методи розпізнавання об'єктів на зображеннях та як найбільш перспективний, було обрано нейромережевий метод. Були проаналізовані різні парадигми штучних нейронних мереж та обґрунтовано вибір для даної задачі згорткової нейронної мережі архітектури SSD. Проаналізовано процес навчання згорткових нейронних мереж та удосконалено структуру згорткової нейронної мережі шляхом доповнення її шістьма новими шарами. Спроектовано програму розпізнавання медичних масок на обличчі на мові програмування Python у середовищі PyCharm з використанням бібліотек Keras, Tensorflow, Tkinter та OPENCV. Розроблена програма має достовірність розпізнавання медичних масок на обличчі на 4% кращу за аналог.

Графічна частина складається з 6 плакатів.

У економічному розділі визначено, що згідно узагальненого коефіцієнту конкурентоспроможності, науково-технічна розробка переважає існуючі аналоги. Термін окупності становить 9 місяців, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Ключові слова: розпізнавання, медична маска, згорткова нейронна мережа.

ABSTRACT

Soroka V. A. Information technology for recognition of medical masks on the face based on a convolutional neural network. Master's thesis in the specialty 122 – «Computer sciences», educational program – «Artificial intelligence systems». Vinnytsia: VNTU, 2023. 99 p.

In Ukrainian language. Bibliogr. : 24 titles; fig . 25; table 10.

In this master's thesis, the development of software for recognizing medical masks on the face based on a convolutional neural network is considered. Known methods of object recognition in images were considered and analyzed, and the neural network method was chosen as the most promising. Different paradigms of artificial neural networks were analyzed and the choice for this task of a convolutional neural network of the SSD architecture was justified. The learning process of convolutional neural networks was analyzed and the structure of the convolutional neural network was improved by supplementing it with six new layers. A program for recognizing medical masks on the face was designed in the Python programming language in the PyCharm environment using the Keras, Tensorflow, Tkinter and OPENCV libraries. The developed program has the reliability of recognizing medical masks on the face by 4% better than the analogue.

The graphic part consists of 6 posters.

In the economic section, it is determined that according to the generalized coefficient of competitiveness, scientific and technical development prevails over existing analogues. The payback period is 9 months, which indicates the commercial attractiveness of the scientific and technical development and may encourage a potential investor to finance the implementation of this development and its introduction to the market.

Keywords: recognition, medical mask, convolutional neural network.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗПІЗНАВАННЯ МЕДИЧНИХ МАСОК НА ОБЛИЧЧІ.....	8
1.1 Постановка задачі	8
1.2 Аналітичний огляд відомих методів класифікації зображень	9
1.3 Обґрунтування вибору аналогу до програми розпізнавання медичних масок на обличчі	17
1.4 Висновок до розділу 1	18
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ МЕДИЧНИХ МАСОК НА ОБЛИЧЧІ НА ОСНОВІ ЗГОРТКОВОЇ НЕЙРОМЕРЕЖІ	19
2.1 Структура інформаційної технології розпізнавання медичних масок на обличчі на основі згорткової нейромережі	19
2.2 Обґрунтування вибору архітектури згорткової нейромережі	20
2.3 Процес навчання згорткових нейромереж	32
2.4 Розробка структури згорткової нейронної мережі.....	38
2.5 Розробка алгоритму розпізнавання медичних масок на обличчі	40
2.6 Висновок до розділу 2	42
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ МЕДИЧНИХ МАСОК НА ОБЛИЧЧІ	43
3.1 Обґрунтування вибору мови та програмних інструментів розробки	43
3.2 Обґрунтування вибору фреймворків та бібліотек	44
3.3 Формування набору зображень для навчання та тестування нейромережі.....	48
3.4 Програмна реалізація розпізнавання медичних масок на обличчі	51
3.5 Висновок до розділу 3	55
4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМИ РОЗПІЗНАВАННЯ МЕДИЧНИХ МАСОК НА ОБЛИЧЧІ	56
4.1 Тестування програми розпізнавання медичних масок на обличчі	56
4.2 Аналіз результатів роботи програми розпізнавання медичних масок на обличчі	58

4.3 Висновок до розділу 4	61
5 ЕКОНОМІЧНА ЧАСТИНА	62
5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки.....	63
5.2 Розрахунок витрат на проведення науково-дослідної роботи	66
5.3 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором	74
5.4 Висновки до розділу 5.....	78
ВИСНОВКИ	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	81
Додаток А (обов'язковий) ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ	84
Додаток Б (обов'язковий) Лістинг програми.....	85
Додаток В (обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА	90
Додаток Г (довідниковий) Інструкція користувача	97

ВСТУП

Актуальність. У наш час у зв'язку із бурхливим розвитком інформаційних технологій та суттєвим збільшенням продуктивності комп'ютерних систем, завдання автоматизації процесів інтелектуальної обробки даних і створення систем штучного інтелекту (ШІ) стають актуальними.

Штучні нейронні мережі (ШНМ) є одним з напрямків ШІ, що зараз активно розвиваються. Вони працюють відповідно до принципів біологічних нейромереж і є системою взаємопов'язаних між собою штучних нейронів. Основними галузями застосування ШНМ є: розпізнавання зображень, обробка аудіо та текстової інформації, прийняття рішень, прогнозування, аналіз даних, оптимізація.

Коронавірус став серйозною проблемою для людей в усьому світі. Він з'явився в Китаї і швидко поширився на інші країни. На сьогоднішній день виявлено понад 40 різновидів таких вірусів. Вперше новий тип, який загрожував людині, був виявлений в 1965 році. У 2019 році був виявлений новий та особливо небезпечний тип SARS-CoV-2. Назва виникла завдяки специфічній структурі вірусу, частинки якого, як правило, прикрашені великими виступами поверхні у формі шипів, які схожі корону.

Щоб запобігти поширенню коронавірусу приймаються різні заходи: скасовуються масові заходи, багато співробітників компаній працюють з дому, школярі переходять на дистанційне навчання. Багато людей знаходяться на карантині. Деякі країни навіть тимчасово закрили кордони.

Через епідемію коронавірусу COVID-19 у всьому світі зростає тенденція носіння масок для обличчя в громадських місцях. Вчені довели, що носіння масок для обличчя працює на перешкоджання передачі COVID-19. У 2020 році швидке поширення COVID-19 змусило Світову організацію охорони здоров'я оголосити COVID-19 глобальною пандемією. Епідемія вірусу корони породила надзвичайний ступінь світового наукового співробітництва.

Штучний інтелект (ШІ), заснований на машинному навчанні та глибокому навчанні, може допомогти в боротьбі з Covid-19 різними способами.

Методи машинного навчання дозволяють оцінювати величезні обсяги даних з метою прогнозування розповсюдження COVID-19, слугувати для раннього попередження про можливі пандемії та прогнозувати вразливі групи населення. Законодавчо у багатьох країнах люди зобов'язані носити медичні маски на обличчі у громадських місцях. Однак процес моніторингу в реальному часі великих груп людей є дуже складним завданням. У процесі моніторингу потрібно виявити тих людей, хто не носить медичну маску на обличчі. Саме через це задача автоматизованого розпізнавання медичної маски на обличчі є вельми актуальною.

У цій роботі розглядається програмна реалізація згорткової нейромережі, яка призначена для розпізнавання медичних масок на обличчі людей. Ця програма призначена для застосування у різних установах та організаціях, таких як залізничні вокзали, аеропорти, розважальні та торговельні центри та інші людні місця як автоматизований засіб моніторингу, що має сенс в умовах пандемії COVID-19.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета і завдання досліджень. Метою магістерської кваліфікаційної роботи є підвищення достовірності розпізнавання медичних масок на обличчі за рахунок використання попередньо натренованої згорткової нейронної мережі.

Для досягнення мети розробки необхідно виконати такі задачі:

- провести аналіз предметної області розпізнавання медичних масок на обличчі;

- розглянути існуючі методи розпізнавання медичних масок на обличчі та обрати й обґрунтувати вибір методу, який задовольняє мету даної магістерської кваліфікаційної роботи;
- розробити структуру інформаційної технології розпізнавання медичних масок на обличчі на основі згорткової нейромережі;
- розробити структуру згорткової нейронної мережі;
- розробити алгоритм розпізнавання медичних масок на обличчі;
- виконати програмну реалізацію запропонованої інформаційної технології;
- провести тестування програмного продукту та виконати аналіз отриманих результатів.

Об’єкт дослідження – процес комп’ютеризованого розпізнавання медичних масок на обличчі з використанням нейронних мереж.

Предмет дослідження – інформаційна технологія та програмні засоби розпізнавання медичних масок на обличчі комп’ютерними засобами з використанням нейронних мереж та достовірність їх роботи.

Методи дослідження. У роботі використані наступні методи наукових досліджень: системного аналізу, теорії штучних нейронних мереж для реалізації інформаційної технології, методи математичної статистики для розробки процесу розв’язання задачі нейромережевого розпізнавання медичних масок на обличчі та обрахунків результатів експериментів із програмним засобом, об’єктно-орієнтованого програмування.

Наукова новизна одержаних результатів.

1. Набула подальшого розвитку інформаційна технологія нейромережевого розпізнавання медичних масок на обличчі, яка відрізняється використанням вдосконаленої згорткової нейронної мережі, що дозволило підвищити достовірність програмних засобів розпізнавання медичних масок на обличчі.

Практичне значення одержаних результатів полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення розпізнавання медичних масок на обличчі.

Запропонована інформаційна технологія сприяє підвищенню достовірності програмних засобів розпізнавання медичних масок на обличчі, зокрема:

- розроблено алгоритм роботи програмного забезпечення розпізнавання медичних масок на обличчі;
- розроблено програмні засоби для розпізнавання медичних масок на обличчі.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології розпізнавання медичних масок на обличчі. Адекватність розроблених математичних моделей підтверджується результатами експериментальних досліджень.

Особистий внесок здобувача. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, написаних у співавторстві, здобувачу належать: аналіз процесу розпізнавання медичних масок на обличчі та методів підвищення достовірності програмних засобів розпізнавання медичних масок на обличчі [1].

Апробація результатів роботи. Результати роботи були апробовані на Міжнародній науково-практичній Інтернет-конференції студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2024)» (м. Вінниця, Україна, 2023-2024 рр.) [1].

Публікації. За результатами досліджень опубліковано одні тези доповіді на науково-технічній конференції [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОЗПІЗНАВАННЯ МЕДИЧНИХ МАСОК НА ОБЛИЧЧІ

1.1 Постановка задачі

Вхідними даними інформаційної технології є електронні зображення у форматі .jpg. Необхідно, використовуючи штучний інтелект, розпізнати на зображенні обличчя людини і встановити чи вдягнуто медичну маску [2]. Таку задачу варто розглядати як завдання класифікації зображень по їх вмісту, тобто по тих об'єктах, що знаходяться на зображенні.

Завданням роботи є огляд існуючих методів та алгоритмів вирішення задачі розпізнавання медичних масок на обличчі за допомогою нейромереж. А метою работ є підвищення точності та швидкодії розпізнавання зображень за рахунок застосування згорткової нейромережі.

У даній роботі планується виконати такі завдання:

- Підготувати набори навчальних і тестових даних для розпізнавання медичних масок на обличчі;
- Провести ефективне навчання нейронної мережі на основі якісного набору навчальних зображень;
- Здійснити порівняння результатів функціонування розробленої програми та аналогів;

Загальна постановка завдання: одержання вихідних векторів, що залежать від ознак вхідного зображення. Таку задачу вирішує класифікація. Кінцева множина класів відома. Класомі є множина об'єктів, для кінцевої підмножини яких відомо до якого із класів вони належать. Ця підмножина носить назву навчальної вибірки. Приналежність до класів інших об'єктів невідома. Потрібно спроектувати алгоритм, здатний прокласифікувати довільний об'єкт із заданої початкової множини [3].

Класифікувати об'єкт – це означає вказати номер (або найменування) класу, до якого даний об'єкт відноситься. Задачі класифікації бувають двокласові та багатокласові. У нашому випадку розпізнавання медичних масок задача класифікації є двокласовою (треба визначити - є маска чи немає маски).

Вхідними даними для нашої задачі класифікації є множина зображень, що містить більше 1000 штук. Кожне зображення є кольоровим і має мітку одного з 2 класів (є маска чи немає маски).

Таким чином, буде розроблено інформаційну технологію та здійснено її програмну реалізацію, яка здатспроможнана класифікувати зображення на 2 класи: 1) обличчя людини з медичною маскою, 2) обличчя людини без медичної маски.

1.2 Аналітичний огляд відомих методів класифікації зображень

При класичній постановці задачі класифікації зображень прийнято застосовувати математичну термінологію, що ґрунтується на логічних міркуваннях і математичних обчисленнях. У контрасті до цього підходу, відомі методи класифікації зображень на основі машинного навчання та штучних нейромереж, що формуються не на математично формалізованих методах класифікації, і демонструють в багатьох випадках не гірший, а значно кращий результат.

Основні відомі методи класифікації:

- статистичні методи;
- байєсівська (наївна) класифікація;
- метод дерев рішень;
- метод опорних векторів;
- метод найближчого сусіда;
- метод нейронних мереж;

Розглянемо загальні терміни та елементи процесу класифікації.

Клас – це множина об'єктів, які мають спільні властивості. Об'єкти одного класу мають бути «схожими». У задачі розпізнавання може бути визначено довільне число класів, але більше одного. Кількість класів позначатимемо літерою S . Кожен клас має свою ідентифікаційну мітку класу [3].

Класифікація – це процес присвоєння міток класу об'єктам відповідно до визначеного опису властивостей цих об'єктів.

Класифікатор – інструмент для привласнення міток класам, який як вхідні дані отримує перелік ознак об'єкта. До одного з найпоширеніших методів класифікації відноситься метод, який базується на описі об'єктів із використанням їхніх ознак, де кожен об'єкт характеризується певним набором чисельних або нечисельних ознак. Проте, існують такі типи даних, для котрих відкриті ознаки не надають достатньої точності класифікації. Наприклад, колір пікселів зображення або відліки цифрового звукового сигналу. Загальна класифікація зображень домашніх тварин чи автомашин є доволі простою задачею для людини і, водночас, складною задачею для комп'ютера. Причиною цього є здатність людини сприймати «приховані ознаки», які часто є недоступними для комп'ютера. Це такі ознаки як морда собаки або колеса автомашини [3].

Верифікація – це процес співставлення досліджуваного об'єкта із однією з моделей об'єкта чи описом класу [3]. Під образом розуміють найзву області у просторі ознак, у якій відображується множина об'єктів або явищ матеріального світу. А ознакою будемо називати опис тієї чи іншої властивості, яка має пряме відношення до даного предмета чи явища.

Простір ознак - N -вимірний простір, що визначається для даної задачі класифікації, де N – це фіксована кількість ознак, які були виміряні для будь-якого з об'єктів. Вектор з простору ознак X , що відповідає об'єкту задачі класифікації, це N -вимірний вектор з N компонентами (x_1, x_2, \dots, x_N) , які утворюють кількісні значення ознак даного об'єкта [4].

Інакше кажучи, класифікація об'єктів може бути визначена як віднесення вхідного об'єкту до відповідного класу за допомогою виділення його істотних ознак або властивостей, які характеризують цей об'єкт, із загальної множини несуттєвих деталей.

Прикладами класифікації є: розпізнавання символів, визначення медичного діагнозу, прогнозування погоди, класифікація документів, розпізнавання осіб, тощо.

Найчастіше вхідними даними слугує отримане з фото- чи відеокамери зображення. Якщо, наприклад, розглянути 2 класи об'єктів: діти та дорослі. Як ознаки можна обрати, скажімо, зріст і вагу (Рис. 1.1).

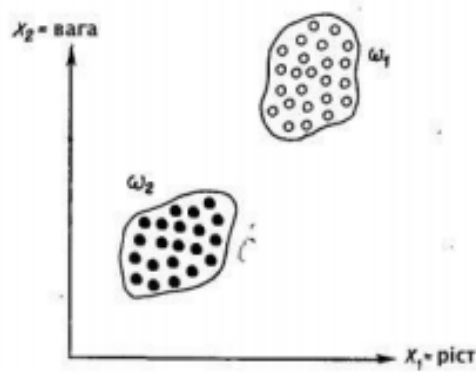


Рисунок 1.1 – Два класи, які не перетинаються

З рис. 1.1 видно, що вказані два класи утворюють дві множини («хмарки» точок), які не перетинаються, що пояснюється обраними ознаками. Проте, не завжди вдається обрати вірні вимірювані параметри як ознаки класів. Наприклад, обрані вище параметри не підійдуть, щоб створити неперетинні класи волейболістів та хокеїстів. Іншим завданням розпізнавання є виділення із вхідних зображень характерних властивостей та/або ознак. Таке завдання відноситься до попередньої обробки. Вдала ознака повинна являти собою характерну властивість класу і при цьому загальну властивість для цього класу.

Міжкласові ознаки – ознаки, які визначають відмінності між класами. Загальні ознаки, які властиві усім класам, не несуть корисну інформацію, а тому для задачі класифікації об'єктів вони не розглядаються як характерні. Обрати правильні ознаки – це одна із найважливіших задач побудови засобів класифікації.

Класичні методи комп'ютерного зору для класифікації зображень умовно можна поділити на три групи: методи логічного аналізу, методи фільтрації, методи машинного навчання.

У завданнях класифікації зображень процес фільтрації часто використовується для попередньої обробки зображень перед процесом аналізу його морфологічних ознак, але зустрічаються і завдання, в яких необхідним і достатнім буде використання тільки фільтрації (наприклад, у задачах комп'ютерного зору).

Вибір області гистограми, бінаризація по порозу. Для RGB зображень і напівтонових зображень (у градаціях сірого) порогом є значення кольору. Встановлення величини порогу, за яким відбудеться бінаризація, і визначає процес бінаризації. Зазвичай бінаризація проводиться за допомогою адаптивного алгоритму, який обирає поріг. Такий алгоритм може використовувати вибір математичного сподівання, піків або моди гистограми. Бінаризація ефективна для сегментації кольорів при роботі з гистограмою [4].

Фур'є-перетворення майже не використовується при обробці зображень у чистому вигляді, так як для аналізу зображень одновимірною перетворення зазвичай не вистачає і виникає потреба використання двовимірною перетворення Фур'є, яке є більш ресурсним. Цей метод використовується лише у разі, якщо необхідно провести аналіз спектру, оскільки використання згортки буде цікавим для областей з уже готовим фільтром. Проте, одновимірне перетворення Фур'є застосовується при стисненні зображень.

Найпростіший приклад фільтра низьких частот – це фільтр Гаусса, а фільтра високих частот – це фільтр Габора. Для кожного пікселя зображення обирається вікно, у якому знаходиться добуток вхідних даних із фільтром

токого ж розміру (згортка). Такий метод отримав доволі широке поширення на практиці, що дозволяє виділяти на зображенні потрібну інформацію та відсікати зайву інформацію [5].

Вейвлет-перетворення визначається, коли відбувається згортка сигналу (ділянки зображення) із деякою характеристичною функцією, яка називається вейвлетом. Вейвлети - це сімейства функцій, локальних за часом і за частотою, у яких всі функції отримують з однієї з функцій за допомогою змін її положення і розтягування на осі часу. Існує набір класичних функцій, які залучено у вейвлет-аналізі. До них відносяться: вейвлет Морлі, вейвлет Хаара, вейвлет МНат, вейвлет Добеши. На практиці під вейвлет-аналізом розуміють пошук довільного патерну на зображенні за допомогою згортки зображення та моделі цього патерну. Класичні вейвлет-функції використовують для класифікації або компресії зображень [5].

Метод обчислення кореляції, який покладено в основу вейвлет перетворення, сам по собі є непересічним інструментом у системах комп'ютерного зору і нерідко використовується у своєму природному вигляді. Наприклад, для визначення рухів або оптичних потоків (кореляція відеопотоків). На основі обрахованої корелятором міжкадрової різниці реалізуються найпростіші детектори руху [5].

Метод фільтрів-функцій. У цьому підході застосовуються математичні фільтри, які дозволяють знаходити прості математичні функції на зображенні, для чого формують акумулятивне зображення (накопичувальний простір), в якому для кожного пікселя вхідного зображення будується множина породжувальних її функцій. Класичним прикладом є узагальнене перетворення Хафа, яке застосовується до бінаризованих зображень, що і дозволяє шукати функції на зображенні. Модифіковане перетворення Хафа дозволяє знаходити будь-які фігури, але його використання в обробці зображень пов'язано з недостатньою стабільністю: низька швидкість роботи і висока чутливість до якості бінаризації примушують шукати ефективніші методи [5].

За допомогою фільтрації можливо отримати множину даних, яка буде придатна для подальшої обробки, але більшість завдань у галузі машинного зору потребують аналізу внутрішньої структури зображень і морфологічних ознак зображених об'єктів. Саме для цього досліджуються і впроваджуються різноманітні методи логічного аналізу зображень. Методи математичної морфології являються результатом теорії та практики аналізу та обробки геометричних структур, що засновані на теорії множин, випадкових функціях та топології.

Ці методи реалізуються за допомогою базових операцій. Операції двійкової морфології являються деякими перетвореннями впорядкованих множин або підмножин (ділянки зображення) за допомогою структурних елементів. Структурним елементом називають бінарне зображення довільного розміру та структури, але найчастіше використовують симетричні елементи: прямокутник фіксованого розміру або коло фіксованого діаметру. Результатом перетворення також є бінарне зображення [6].

За допомогою методів математичної морфології можна видаляти шуми з двійкових зображень, а також реалізувати алгоритми виділення контурів, але на практиці ці методи застосовуються у поєднанні з іншими способами.

Контурний аналіз – це потужний математичний апарат, який дозволяє описувати, знаходити і зберігати об'єкти, що знаходяться в формі зовнішніх контурів. Раніше розглядалися фільтри контурів, результатом використання яких природним чином є контури об'єктів на зображенні, без застосування додаткової попередньої бінаризації. У контурному аналізі попередні етапи додаткової бінаризації і фільтрації є обов'язковими стадіями завдання. Передбачається, що контур має потрібну інформацію про форму об'єкта, а внутрішні точки до уваги не беруться, що суттєво обмежує галузь застосування алгоритмів контурного аналізу. Проте, контури, отримані з його допомогою, дозволяють переходити від двовимірного простору образів до простору контурів. У деяких завданнях це суттєво знижує складність алгоритму.

Методи контурного аналізу є інваріантними до зсуву, повороту і масштабу зображення об'єкта. Серед методів контурного аналізу можна виділити такі, у яких контур об'єкта відстежується і векторизується; сканувальні алгоритми, які засновані на огляді всього зображення і виділенні контурних точок без відстеження; методи виділення контурів у напівтонових зображеннях, методи дослідження кривизни функцій. На практиці методи контурного аналізу є достатньо чутливими до умов середовища, що може викликати складність їх застосування в реальних умовах більшості завдань машинного зору. Проте, вони є корисними у задачах комп'ютерного зору, коли умови середовища досить суворо визначені. У таких ситуаціях методи контурного аналізу будуть неперевершеними по швидкодії, володіють зрозумілою і простою логікою, яка обумовлює зручність їх використання в багатьох спеціалізованих областях.

Знаходження особливих точок (feature detection). Для вирішення допоміжної прикладної задачі у даній роботі буде використаний один з алгоритмів feature detection [6], тому зупинимося на цій темі докладніше. Знаходження особливих точок є одним з найбільш поширених методів у комп'ютерному зорі. Особливі точки віддзеркалюють унікальні характеристики об'єкта, що дозволяє порівнювати різні зображення одного об'єкта або об'єктів одного класу між собою. Саме тому особливі точки на практиці найбільш актуальні у задачах, де можливим рішенням буде обробка послідовності зображень або відеопотоку та подальший аналіз отриманих масивів особливих точок.

Алгоритми feature detection умовно класифікують за ступенем стабільності точок при переходах від одного кадру (зображення) об'єкта або класу об'єктів до іншого. Складність алгоритмів знаходження зростає відповідно до необхідного рівня стабільності шуканих точок. Результатом роботи таких алгоритмів є множина особливих точок, зокрема, кутів, для яких потрібно виконати математичний опис.

Формування математичного опису є завданням дескриптора. Багато різних дескрипторів одночасно вирішують задачу пошуку особливих точок і побудови описів для цих точок за допомогою вбудованих алгоритмів або за допомогою оригінальних власних методів. Описи (ознаки) будуються на основі інформації про інтенсивність, текстуру та колір особливих точок.

Метод штучних нейронних мереж. Штучна нейронна мережа є концептуальною моделлю біологічної нейромережі і складається з пов'язаних між собою певним чином шарів штучних нейронів, що організують загальну активну структуру і впливають функціонально на роботу один іншого.

У більшості архітектур штучних нейронних мереж активність нейрона визначається перетворенням сумарного зовнішнього впливу інших нейронів на цей нейрон.

Технології штучних нейронних мереж з моменту свого зародження розвивалися дещо відокремлено від класичних методів розпізнавання та зачасти докорінно змінювали уявлення про проблематику і предмет теорії машинного навчання та класифікації зображень, формуючи суттєвий вплив на теоретичний, методологічний і термінологічний апарати цих дисциплін. Через певний час після виникнення базових моделей штучних нейромереж, відбувся значний поділ нейромережевої науки відповідно до видів топологій та архітектур нейромереж та методи їх навчання.

У більшості архітектур штучних нейромереж функції активації нейронів є фіксованими, а ваги синапсів є змінюваними параметрами нейромережі. Деякі входи нейронів є зовнішніми входами всієї нейромережі, а деякі виходи нейронів - виходами всієї нейромережі.

Завдання нейронної мережі є перетворення вхідного вектора у вихідний вектор, яке здійснюється залежно від топології та ваг нейромережі.

Таким чином, серед усіх відомих методів класифікації зображень для реалізації у даній роботі оберемо метод на основі штучних нейронних мереж, тому що він характеризується адаптивністю і добре працює у

випадках нечіткої постановки завдань чи при наявності певної невизначеності чи неповноти вхідних даних.

1.3 Обґрунтування вибору аналогу до програми розпізнавання медичних масок на обличчі

На теперішній час існує кілька відомих програмних реалізацій, функціонально призначених для розпізнавання медичних масок на обличчі.

Серед таких програм можна назвати Face Mask Detector [7], що є програмним інструментом, створеним на мові Python (див. рис. 1.2). Програма Face Mask Detector має гібридний дизайн, який використовує машинне навчання на основі глибоких нейромереж для розпізнавання маски на обличчі та складається з двох частин. Як перша частина використовується глибока нейронна мережа Resnet50 для вилучення ознак. Друга частина (класифікатор) використовує машину опорних векторів та метод дерев рішень для класифікації ознак. Програма Face Mask Detector має достовірність розпізнавання на тестовій вибірці 95,5 відсотка для набору даних RMFD,

До недоліків цієї програми можна віднести невисоку достовірність її роботи та низьку швидкодію.

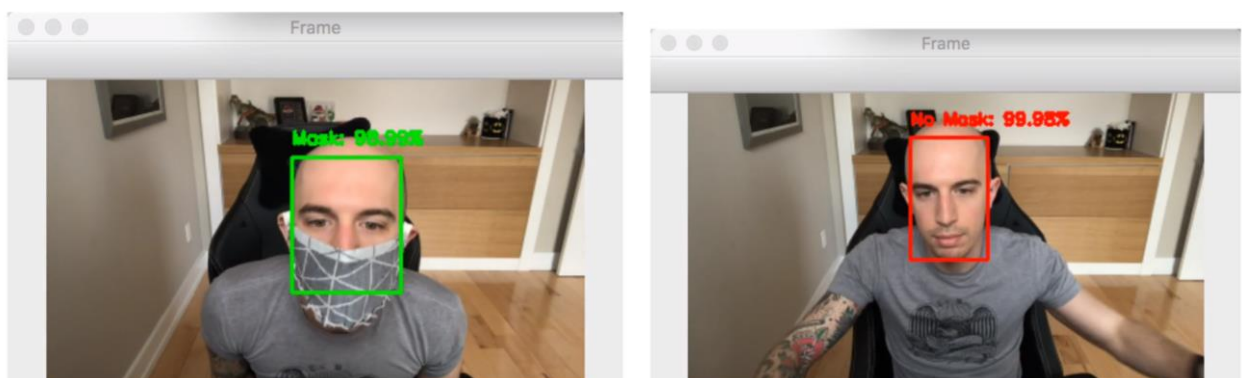


Рисунок 1.2 – Вікно програми Face Mask Detector

Інші програмні реалізації також мають такий недолік як невисока достовірність розпізнавання медичних масок на обличчі. З огляду на це, постає завдання розробки нового програмного засобу для розпізнавання медичних масок на обличчі з підвищеною достовірністю роботи.

1.4 Висновок до розділу 1

У розділі розглянуто постановку задачі розпізнавання медичних масок на обличчі, проведено аналітичний огляд відомих методів класифікації зображень, які можуть використовуватися для поставленої задачі. Обґрунтовано доцільність вибору методу нейронних мереж для розпізнавання медичних масок на обличчі. Було проаналізовано різні програми для детектування медичних масок на обличчі та обрано аналог. Головним недоліком аналога та інших подібних програм є невисока достовірність розпізнавання медичних масок на обличчі, із чого слідує мета роботи – підвищення достовірності розпізнавання медичних масок на обличчі.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ МЕДИЧНИХ МАСОК НА ОБЛИЧЧІ НА ОСНОВІ ЗГОРТКОВОЇ НЕЙРОМЕРЕЖІ

2.1 Структура інформаційної технології розпізнавання медичних масок на обличчі на основі згорткової нейромережі

Завданням цієї роботи є розробка інформаційної технології та її програмної реалізації, яка визначала би по зображенню людини чи вдягнена в неї на обличчі маска. Тому вхідними даними є файл із зображенням людини, на обличчі якої є медична маска або немає. Структура інформаційної технології розпізнавання медичних масок на обличчі на основі згорткової нейромережі представлена на рис. 2.1.



Рисунок 2.1 – Структура інформаційної технології розпізнавання медичних масок на обличчі на основі згорткової нейромережі

Усі зображення, які подаються на вхід нейромережевої системи, повинні піддатися попередній обробці, яка полягає у приведенні початкового зображення до певного розміру, на який налаштована нейромережа. Крім цього, над зображеннями із навчальної вибірки, на яких навчається нейромережа, повинна здійснюватися аугментація (тобто внесення незначних шумів та здійснення невеликих афінних перетворень). Це робиться з метою збільшення ентропії навчальної інформації. Після цього здійснюється навчання нейронної мережі на розпізнавання медичних масок. У першому розділі ми з'ясували, що найперспективнішим методом є нейромережевий, а із всіх нейронних мереж для цієї задачі найбільше підходять згорткові нейромережі, оскільки вони ефективно працюють із зображеннями завдяки їх здатності зменшення розмірності вхідних даних. У цьому розділі обґрунтуємо вибір конкретної архітектури згорткової нейронної мережі. Далі у інформаційній технології здійснюється збереження навченої нейронної мережі, після чого її вже можна використовувати для виявлення обличчя на вхідному зображенні та подальшої класифікації цього обличчя («У масці» чи «Без маски»). І нарешті – останнім є процес виведення результату розпізнавання.

2.2 Обґрунтування вибору архітектури згорткової нейромережі

Згорткова нейронна мережа, яка була розроблена в 2012 році і відома під назвою AlexNet, була першою моделлю згорткових нейронних мереж, яка досягла в змаганні ILSVRC 2012 (ImageNet Large Scale Visual Recognition Challenge) відсотка помилок 15,4% [8]. У моделі AlexNet були запропоновані такі новітні технології, як збільшення обсягу вхідних даних (аугментація) та дропаут (dropout) - шар, що обнуляє виходи деяких із нейронів, які обираються випадково і по-новому для кожного наступного навчального прикладу. Ці методи довели перевагу згорткових нейромереж і

застосовуються по теперішній час. Архітектура мережі AlexNet являє собою 5 згорткових шарів, 1 шар субдискретизації, 1 шар дропаутів і 3 повнозв'язаних шари.

Стандартна модель вихідного сигналу нейрона визначається як функція f від вхідного сигналу x так: $f(x)=\tanh(x)$ або $f(x)=(1+e)^{-1}$. У нейромережі AlexNet запропоновано обрати активаційну функцію лінійну з обмеженням *ReLU*. У згорткових нейромережах активаційна функція *ReLU* збільшує продуктивність мережі у декілька разів порівняно з використанням активаційної функції гіперболічний тангенс $\tanh(x)$. Чотиришарова згорткова нейронна мережа з функцією активації *ReLU* досягає величини помилки 25% у шість разів швидше, аніж така ж нейромережа з функцією активації гіперболічний тангенс $\tanh(x)$.

Архітектура нейромережі AlexNet зображена на рис. 2.2. Нейромережа містить 8 шарів з вагами; причому перші п'ять шарів є згортковими, а останні три шари є повнозв'язними. Останній шар мережі містить 1000 виходів, які визначають номер класу. Фільтри другого, четвертого та п'ятого згорткових шарів з'єднано з картами ознак попереднього шару тільки. Фільтр третього згорткового шару мережі з'єднаний з усіма картами ознак другого шару мережі. Нейрони повнозв'язних шарів мережі зв'язані з усіма нейронами попередніх шарів мережі. Шари нормалізації є наступними за першим та другим згортковими шарами мережі. Шари субдискретизації є наступними за шарами нормалізації та п'ятим згортковим шаром мережі. Функція активації обмежена лінійна *ReLU* використовується у кожному згортковому шарі та у повнозв'язному шарі мережі.

На перший згортковий шар надходить вхідне зображення розміром $224 \times 224 \times 3$. Воно обробляється 96-ма фільтрами розміром $11 \times 11 \times 3$ кожен з кроком у 4 пікселі (це відстань між центрами ділянок сприйняття сусідніх нейронів у карті ознак). Другий згортковий шар мережі приймає на вхід вихідні сигнали першого шару та обробляє їх фільтрами з розміром $5 \times 5 \times 48$ кожен. Третій, четвертий та п'ятий згорткові шари мережі з'єднані з іншими

шарами без застосування шарів субдискретизації і нормалізації. Третій згортковий шар мережі має 384 фільтра з розміром $3 \times 3 \times 256$ кожен, що з'єднані з нормалізованим та субдискретизованим виходом другого шару мережі. Четвертий шар мережі має 384 фільтра з розміром $3 \times 3 \times 192$ кожен, п'ятий шар мережі має 256 фільтрів з розміром $3 \times 3 \times 192$ кожен. Повнозв'язні шари мають по 4096 нейронів.

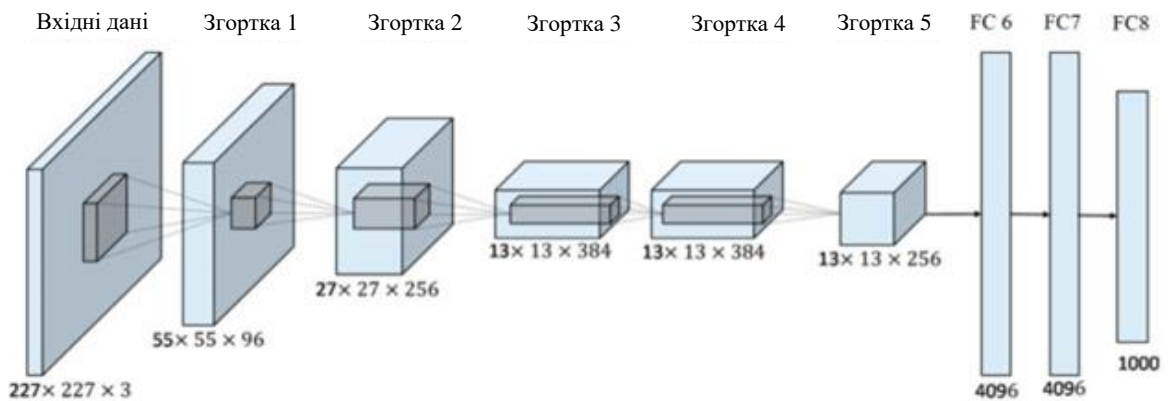


Рисунок 2.2 - Архітектура нейромережі AlexNet

Для зменшення перенавчання в нейромережі AlexNet запропоновані такі методи:

- Збільшення обсягу навчальних даних – це найпростіший і часто використовуваний метод для зменшення перенавчання. Цей метод полягає у додатковому перетворенні навчальних зображень з метою збільшення кількості вхідних тренувальних зображень. Для цього застосовується дзеркальне відображення картин по горизонталі і змінення інтенсивностей кольору пікселів.

- Дропаут – це встановлення в нуль сигналу кожного нейрона у прихованому шарі з імовірністю 0,5. Ті нейрони, до яких застосовано операцію дропаут, не беруть участі при навчанні у зворотному поширенні помилки. Тому, кожен раз, коли на вхід мережі надходить зображення, нейромережа має різну архітектуру. Дропаут еквівалентний усередненню всіх

моделей, які отримувались на кожному кроці випадковим викиданням деяких нейронів.

Після успіху нейромережі AlexNet в 2012 році, під час проведення конкурсу ILSVRC 2012 Workshop, широко обговорювалася можливість узагальнення результатів роботи згорткових нейронних мереж для задачі розпізнавання різноманітних об'єктів у PASCAL VOC Challenge. У моделі нейронної мережі R-CNN показано, що згорткові нейромережі значно перевершують за якістю детектування об'єктів в PASCAL VOC у порівнянні з методами, які засновані на методі гістограм орієнтованих градієнтів [9].

Для того, щоб досягти цей результат, було вирішено два завдання: 1) локалізація об'єктів за допомогою глибокої нейромережі і 2) навчання моделі мережі тільки на невеликій кількості анотованих даних. Задача розпізнавання передбачає локалізацію об'єкта (або кількох об'єктів) на заданому зображенні. Ранні моделі згорткових нейромереж вирішували цю задачу методом ковзного вікна. Ці нейромережі мали, зазвичай, тільки 2 згорткових шари і 1 шар субдискретизації. В нейромережі R-CNN запропоновано використовувати замість методу ковзного вікна метод «розпізнавання всередині області». Цей метод утворює майже 2000 незалежних ділянок на вхідному зображенні, витягає вектор ознак фіксованої довжини з кожної ділянки, використовуючи при цьому згорткову нейромережу, та потім класифікує кожну ділянку за допомогою SVM класифікатора. На рисунку 2.3 зображено алгоритм R-CNN. Із вхідного зображення (1) витягується майже 2000 ділянок (2). У кожній ділянці обчислюються вектори ознак фіксованої довжини за допомогою згорткової нейромережі (3). Отримані ознаки класифікуються за допомогою SVM класифікатора (4)

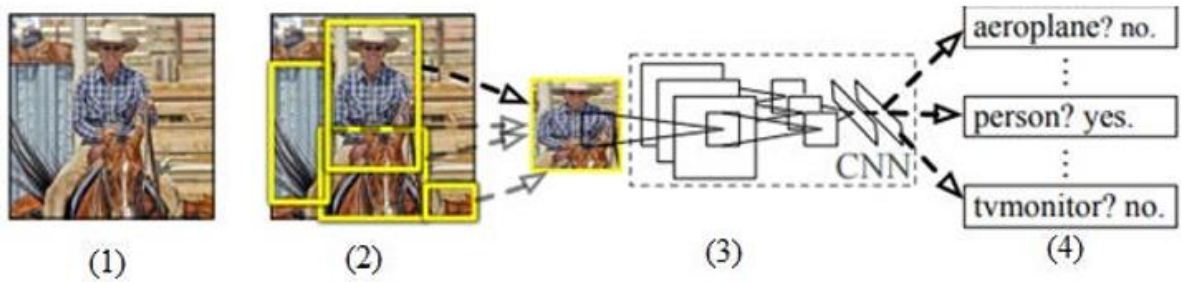


Рисунок 2.3 - Архітектура нейромережі R-CNN

В роботі [9] показано, що нейромережа R-CNN дає приріст 30% у порівнянні з найкращим результатом змагання PASCAL VOC 2012. Модель нейромережі R-CNN має такі недоліки:

- Навчання нейромережі забирає кілька стадій. Нейромережа R-CNN спочатку навчається шляхом обчислення функції втрат. Потім навчається SVM класифікатор ознак. На третьому етапі навчання нейромережі навчаються регресори прямокутників, які обрамляють об'єкти, для коригування істинної області розташування об'єкта.

- Навчання нейромережі вимагає багато часу і ресурсів комп'ютера. Для SVM класифікатора ознаки витягують із кожної ділянки зображення і записують на твердий диск. Для дуже глибоких нейромереж, таких як, наприклад, мережа VGG16, цей процес триває 2,5 доби на графічному процесорі GPU при 5000 навчальних зображень. Ці ознаки потребують сотні гігабайт для зберігання на твердому диску.

- Низька швидкодія розпізнавання об'єктів. Протягом обчислень ознаки витягаються із кожної ділянки для кожного об'єкта. Розпізнавання триває ~ 47 секунд на 1 зображення при використанні графічного процесора GPU.

У моделі згортової нейромережі Fast R-CNN запропоновані методи покращення мережі R-CNN [10]. У згортковій нейромережі Fast R-CNN запропоновано новий метод навчання, який виправляє недоліки R-CNN і підвищує швидкість і точність навчання. Алгоритм роботи мережі Fast R-CNN зображено на рис. 2.4. Вхідне зображення і ділянка множинної зацікавленості (ROI) надходять на вхід згортової нейромережі. Ділянка

множинної зацікавленості - це шар, який виконує субдискретизацію пікселів вхідного зображення та створює карту ознак фіксованого розміру 7×7 пікселів. Основна мета цього шару - прискорення швидкості навчання і тестування.

Кожна ділянка множинної зацікавленості об'єднується у карті ознак фіксованого розміру, а за тим, у вектор ознак за допомогою повнозв'язних шарів нейронів (fully-connected layer, FC). Кожен вектор ознак надходить на входи послідовності повнозв'язних шарів, які в кінці мережі поділяються на два вихідних шари.

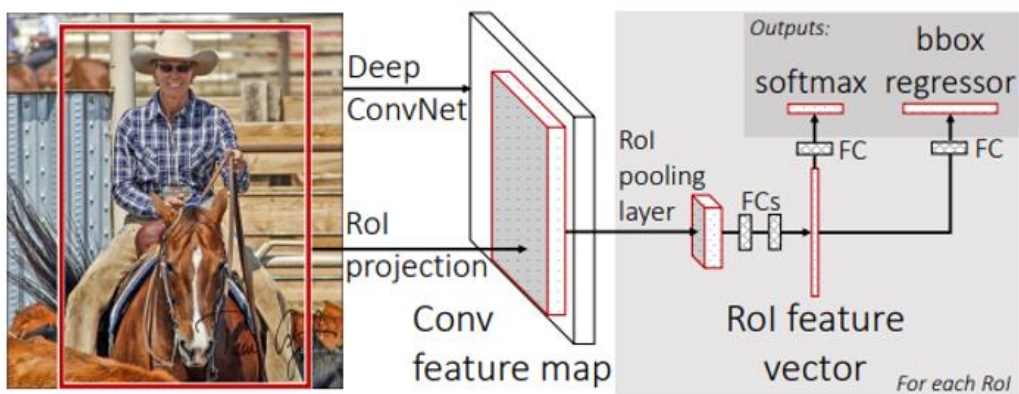


Рисунок 2.4 - Алгоритм мережі Fast R-CNN.

Нейромережа Fast R-CNN має наступні переваги:

- Вищу якість розпізнавання, ніж у R-CNN.
- Навчання здійснюється в один етап.
- При навчанні оновлюються ваги усіх шарів нейромережі.
- Для зберігання ознак не потрібно простору на твердому диску.

У попередніх моделях нейромереж (R-CNN, Fast R-CNN) використовувалися методи пропозиції ділянок для первинної генерації обмежувального прямокутника для об'єктів, а вже потім ці ділянки класифікувалися. Після класифікації застосовувалася постобробка обмежувальних прямокутників для виключення дублікатів при розпізнаванні.

Такі методи є повільними і важко оптимізуються, тому що кожен компонент необхідно навчати незалежно.

У моделі нейромережі YOLO запропоновано визначати розпізнавання об'єктів як одне регресійне завдання - просто від пікселів зображення до координат обмежувальних прямокутників та імовірностей класів для кожного об'єкта [11]. Алгоритм роботи моделі YOLO зображено на рис. 2.5.

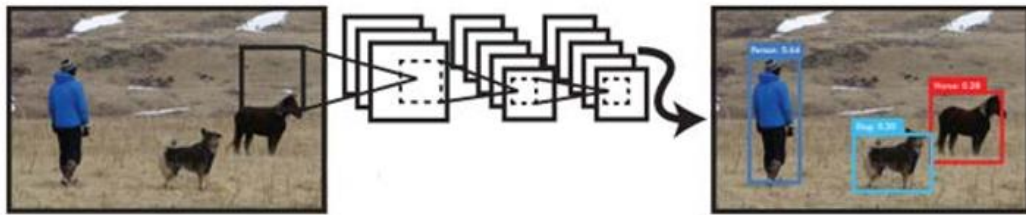


Рисунок 2.5 - Алгоритм нейромережі YOLO

Алгоритм виявлення об'єктів виглядає таким чином:

1. Вхідне зображення перетворюють до розміру 448x448 пікселів.
2. Запускають згорткову нейромережу для розпізнавання об'єктів.
3. Обчислюють результат розпізнавання як числове значення довіри.

Одна згорткова нейромережа одночасно передбачає множину обмежувальних прямокутників та імовірності належності до класів для них. Алгоритм розпізнавання об'єктів в нейромережі YOLO зображено на рис. 2.6 і виглядає так:

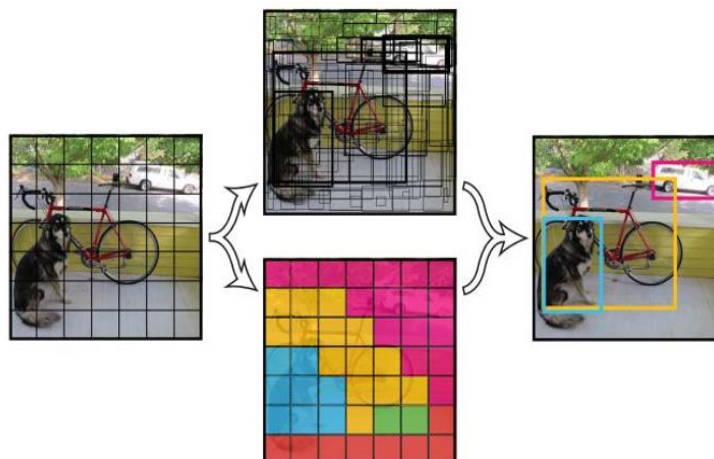


Рисунок 2.6 - Алгоритм розпізнавання об'єктів в мережі YOLO

Модель мережі YOLO ділить вхідне зображення сіткою $S \times S$. Якщо центр об'єкта потрапляє в комірку сітки, то ця комірка бере участь у подальшому детектуванні об'єкта. У кожній такій комірці передбачається B обмежувальних прямокутників та обчислюється їхня імовірність. Це імовірність того, що цей обмежувальний прямокутник містить шуканий об'єкт. Далі визначається імовірність передбачення положення та розмірів прямокутника. Формально, імовірність визначається за формулою:

$$P = P(Obj) \cdot IOU(truth, pred)$$

де $P(Obj)$ - імовірність того, що в обмежувальному прямокутнику міститься об'єкт, а $IOU(truth, pred)$ – це відношення ділянки перекриття передбаченого та істинного обмежувальних прямокутників до ділянки об'єднання цих прямокутників, яке визначається за формулою:

$$IOU(truth, pred) = \frac{truth \cap pred}{truth \cup pred}$$

Кожен обмежувальний прямокутник містить 5 різних передбачень: координати x , y , w , h та імовірність P . Координати (x, y) визначають центр ділянки кордонів комірки сітки. Координати (w, h) визначають розмір прямокутника відносно всього зображення. У кожній комірці сітки також передбачається умовна імовірність класу $C = P(Class_i | Obj)$. Під час тестування перемножуються імовірності класів і окремі ділянки передбачення об'єктів:

$$C \cdot P = P(Class_i | Obj) P(Obj) \cdot IOU(truth, pred) = P(Class_i) \cdot IOU(truth, pred),$$

що дає оцінку імовірності належності кожної ділянки до кожного класу. Ці передбачення кодується тензором розміру $S \cdot S(B \cdot 5 + C)$.

Архітектура згорткової нейромережі моделі YOLO зображена на рис. 2.7.

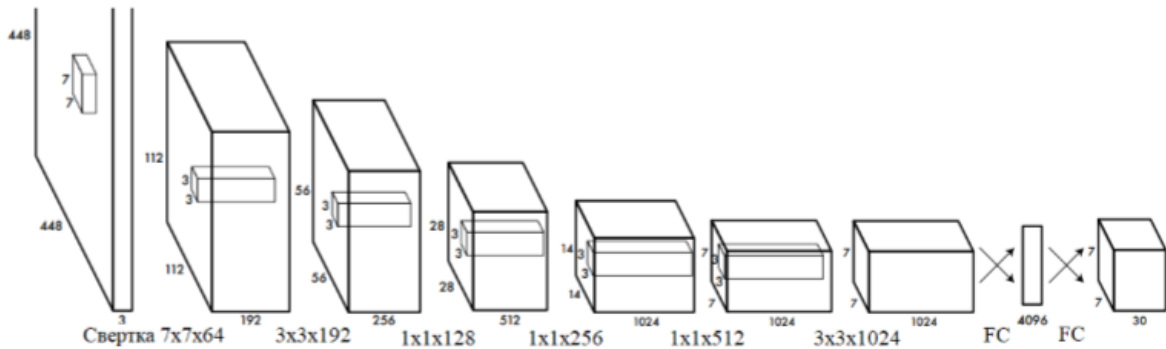


Рисунок 2.7 - Архітектура згорткової нейромережі YOLO

Модель нейромережі YOLO має такі переваги:

- Велика швидкість розпізнавання. Нейромережа має швидкодію 45 кадрів в секунду, а її швидка версія (Fast YOLO) - більш ніж 150 кадрів в секунду.

- нейромережа YOLO робить прогноз по всьому зображенню, на відміну від методу ковзного вікна та методів, що засновані на ділянках пропозиції. Це дозволяє моделі мережі кодувати контекстну інформацію про класи. Завдяки цьому мережа YOLO робить в два рази менше помилок на зображенні у порівнянні з мережею R-CNN, оскільки модель мережі R-CNN може робити помилки на зображенні через те, що не може побачити ширший контекст.

- Завдяки тому, що мережа YOLO аналізує узагальнені подання об'єктів, то при тренуванні на природніх зображеннях і тестуванні на різних творах образотворчого мистецтва, мережа YOLO набагато перевершує найкращі методи розпізнавання (такі як R-CNN та Fast R-CNN).

Подальший розвиток ідеї архітектури YOLO отримали у моделі мережі SSD, що використовує такий саме принцип розпізнавання об'єктів одразу по усьому зображенню, що відображено у її назві - Single Shot Detector [12]. Модель мережі SSD - перша глибока мережа для детектування об'єктів, яка не використовує ознаки, що складаються з обмежувальних прямокутників. Це призводить до суттєвого збільшення продуктивності і детектування об'єктів з більшою точністю. Основне покращення швидкодії відбувається при відмові від застосування обмежувальних прямокутників. Також покращення точності відбувається при використанні невеликого згорткового фільтра для передбачення класу об'єкта і іншого згорткового фільтра для корекції положення обмежувальних прямокутників із різними співвідношеннями сторін і різним розміром. Ці згорткові фільтри застосовують до декількох карт ознак, як на ранніх, так і на більш пізніх шарах нейромережі, що дозволяє детектувати об'єкти різного розміру.

На рис. 2.8 зображено порівняння нейромережових архітектур YOLO і SSD. У моделі мережі SSD з'являються додаткові шари, які дозволяють детектувати об'єкти різного розміру.

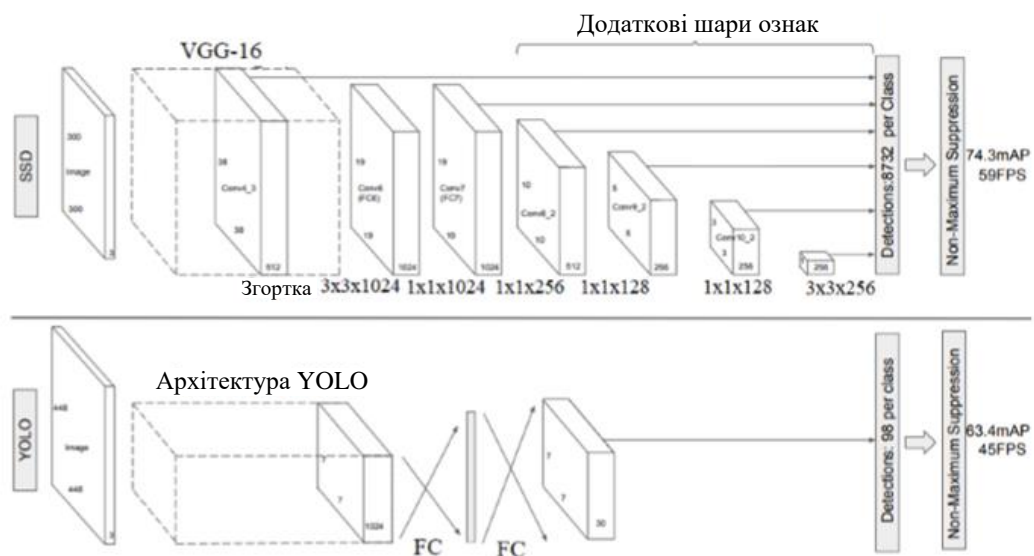


Рисунок 2.8 - Порівняння архітектури мережі SSD (зверху) і мережі YOLO (знизу)

При порівнянні моделей цих нейромереж можна отримати однозначної відповіді, яка із моделей є кращою. Для застосування у реальному житті необхідним є баланс між точністю та швидкістю. Крім того, необхідно враховувати й інші фактори, які впливають на якість моделі мережі:

- Модель формування ознак (VGG, Inception, ResNet, MobileNet).
- Роздільна здатність вхідного зображення.
- Як передбачення виключається з розрахунку функції втрат.
- Кількість передбачень в моделі мережі.
- Спосіб кодування обмежувальних прямокутників.
- Збільшення обсягів (аугментація) наборів даних.
- Набір даних, що використовується для тренування.
- Використання великомасштабних зображень в тренуванні або тестуванні.
- Робота із програмним забезпеченням для глибокого навчання.
- Локалізація функції втрат.
- Параметри тренування, такі як розмір пакета навчання (кількість даних для тренування на одному кроці), крок тренування, падіння швидкості тренування.

Також на адекватність порівняння моделей нейромереж впливає те, що сучасні технології розвиваються досить швидко, і будь-яке поточне порівняння швидко застаріває. У цьому розділі об'єднано результати з окремих джерел інформації. Результати змагання PASCAL VOC 2012 наведено у табл. 2.1 [13]. Як критерій якості використовується середня точність по всіх класах (mAP), а найкращі результати отримано у моделей нейромереж SSD 512 ($mAP = 74,9$) та SSD 300 ($mAP = 72,4$).

Таблиця 2.1 - Результати порівняння моделей нейромереж по *mAP* для класу «person»

Модель	<i>mAP</i>	person
Fast R-CNN	68,4	72,0
YOLO	67,9	63,5
SSD300	72,4	77,4
SSD512	74,9	78,3

На рис. 2.9 показано порівняльний результат точності різних моделей нейромереж, які навчалися на наборах PASCAL VOC 2007 та PASCAL VOC 2012 [13]. Значення вимірювалися на валідаційному наборі даних PASCAL VOC 2012. Для мережі SSD показано результати для вхідного зображення 300×300 та 512×512. Для мережі YOLO показано результати для вхідного зображення 288×288, 416×416 та 514×514. Більш висока роздільна здатність вхідного зображення для моделі мережі забезпечує більш прийнятне значення *mAP*, але меншу продуктивність.

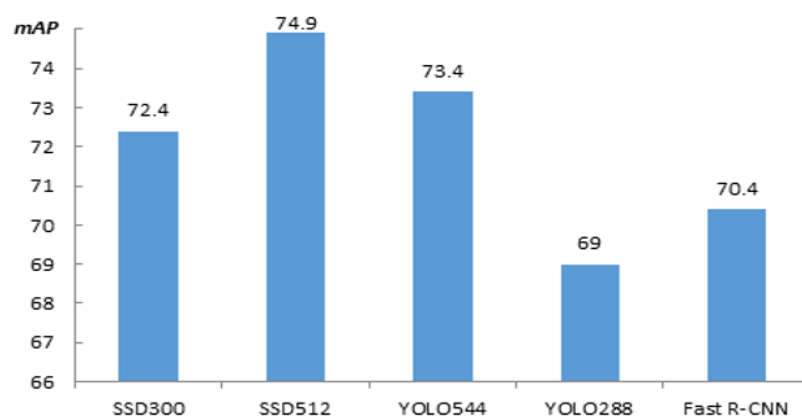


Рисунок 2.9 – Результати точності різних моделей нейромереж в PASCAL VOC 2007 та PASCAL VOC 2012

На рис. 2.10 показано порівняльний результат продуктивності різних нейромереж, що вимірюється в FPS (кадрів у секунду) [13]. Тренувальні та валідаційні набори даних аналогічні попереднім.

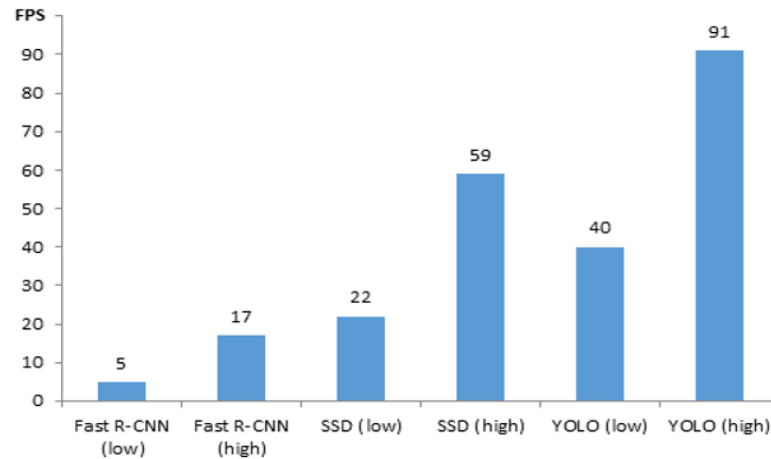


Рисунок 2.10 - Порівняння швидкодії різних моделей нейромереж

За наведеними вище по тексту результатами видно, що найбільшу продуктивність забезпечує нейромережева модель YOLO, але вона має гіршу точність у порівнянні з іншими моделями мереж. Кращу точність має модель мережі R-CNN і її модифікації, але вона має надто низьку продуктивність. Модель мережі SSD має оптимальне співвідношення точності та продуктивності, а тому ця модель взята за основу у цій роботі для розробки моделі розпізнавання медичних масок на обличчі.

Для того, щоб розробити власну модель на основі згорткових нейромереж, потрібно розглянути принципи роботи та навчання згорткових нейромереж.

2.3 Процес навчання згорткових нейромереж

Основне завдання у навчанні (тренуванні) нейромереж – це мінімізація заданої функції помилки [14]. Зазвичай, оптимізується апостеріорна ймовірність.

$$p(\theta|D) = p(\theta) \prod_{d \in D} p(d|\theta),$$

де функція правдоподібності $p(\theta)$ - помилка на тренувальній вибірці, апріорний розподіл $p(d | \theta)$ — це регуляризація.

Завдання оптимізації - у заданій функції знайти аргументи, по яких ця функція мінімізується. Функція помилки в нейромережах має багато локальних екстремумів і один глобальний. Для знаходження оптимального екстремуму використовується евристичний метод оптимізації, названий градієнтним спуском.

Представивши функцію помилки поверхнею, задача оптимізації зводиться до підрахунку градієнта. Якщо функцію, яка визначає поверхню, позначити через E :

$$E(\boldsymbol{\theta})=E(\theta_1, \theta_2, \dots, \theta_n),$$

де $(\theta_1, \theta_2, \dots, \theta_n)$ – аргументи функції, то її градієнт ∇E – це вектор окремих похідних функції кількох змінних по кожному з аргументів:

$$\nabla_{\boldsymbol{\theta}} E = \begin{pmatrix} \frac{\partial E}{\partial \theta_1} \\ \vdots \\ \frac{\partial E}{\partial \theta_n} \end{pmatrix}.$$

Градієнт визначає напрямок, у якому функція зростає найшвидше. А, отже, напрямок, у якому вона найшвидше спадає – це напрямок, зворотній до градієнту: $-\nabla_{\boldsymbol{\theta}} E$. Позначивши через $\boldsymbol{\theta}_t$ вектор параметрів моделі на кроці t , а через E – функцію, що мінімізується, можна визначити вектор оновлення параметрів на етапі t як:

$$\begin{aligned} \mathbf{u}_t &= -\eta \nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}_{t-1}), \\ \boldsymbol{\theta}_t &= \boldsymbol{\theta}_{t-1} + \mathbf{u}_t, \end{aligned}$$

де η – коефіцієнт швидкості навчання. Коефіцієнт швидкості навчання регулює крок оновлення параметрів.

У стандартному методі градієнтного спуску коефіцієнт швидкості навчання задається вручну, що може привести до таких проблем:

– Якщо крок занадто малий, то навчання буде занадто довгим і підвищується імовірність потрапити у локальний (невдалий) мінімум функції (див. Рис. 2.11(а)),

– Якщо крок великий, то можна пропустити потрібний локальний мінімум (див. Рис. 2.11(б)).

Для уникнення цих проблем застосовується стохастичний градієнтний спуск - розрахунок помилки і поновлення ваг відбувається не після проходу всієї навчальної множини, а після кожного прикладу:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta \nabla E(f(\mathbf{x}_t, \boldsymbol{\theta}_{t-1}), y_t)$$



Рисунок 2.11 - Проблеми з коефіцієнтом швидкості градієнтного спуску:

- а) — занадто малі кроки,
- б) — занадто великі кроки.

Основна перевага методу стохастичного градієнтного спуску - велика швидкість обчислення помилки, оскільки поновлення ваг відбувається після

кожного кроку, що збільшує швидкість навчання. Також стохастичний градієнтний спуск працює випадково, що дозволяє при перескакуванні знайти більш оптимальний локальний мінімум функції помилки.

На практиці частіше використовується стохастичний градієнтний спуск по міні-батчам (пакетах), тобто невеликим підмножинам навчального набору. Це дозволяє зберегти переваги методу стохастичного градієнтного спуску, і при цьому користуватися процедурами матричної алгебри, які обраховуються швидко на пристроях, які підтримують паралельні обчислення.

При оптимізації великої нейромережі, функція помилки є складною композицією інших функцій. Представивши складну функцію як композицію простих, можна обрахувати її похідну, що і потрібно для виконання градієнтного спуску. Функція представляється у вигляді графу обчислень, вузлами якого є функції, а ребра зв'язують функції зі своїми аргументами. На рис. 2.12 зображено приклад графу обчислень для функції:

$$f(x, y) = x^2 + xy + (x + y)^2.$$

На рис. 2.12а - з використанням функцій множення \times , додавання $+$ та піднесення у квадрат x^2 , на рис. 2.12б - з використанням функцій множення \times та додавання $+$, на рис. 2.12в - у вигляді дерева з використанням функцій множення \times та додавання $+$.

У нейромережах в якості елементарних застосовуються функції, з яких отримуються штучні нейрони. Наприклад, функції скалярного добутку двох векторів $x \top y$ і функції логістичної сигмоїди σ достатньо для того, щоб побудувати будь-яку нейромережу, складену із нейронів з функцією активації логістичної сигмоїди σ .

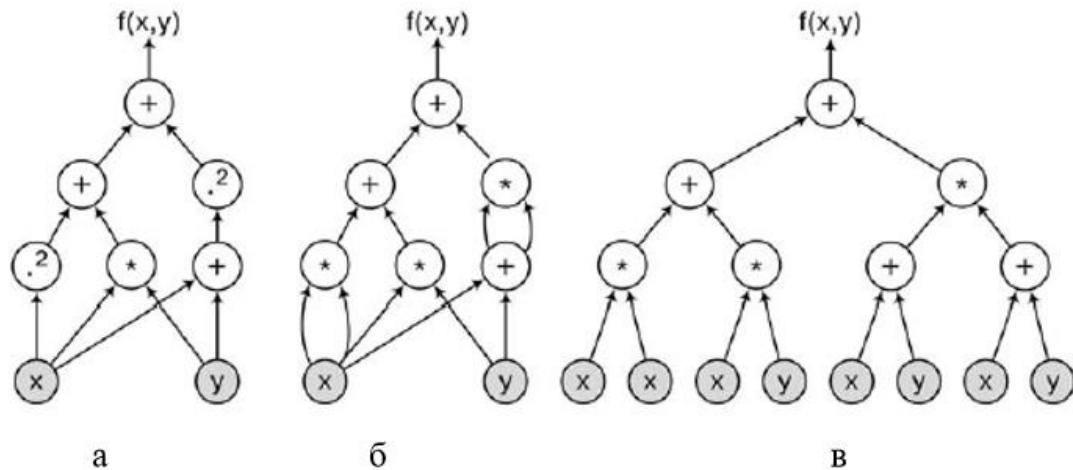


Рисунок 2.12 – Приклад графу обчислень для функції

$$f(x, y) = x^2 + xy + (x + y)^2.$$

Таким чином, представивши складну функцію через прості, можна обчислити похідну цієї складної функції, обчисливши окремі похідні її складових:

$$(f \circ g)'(x) = (f(g(x)))' = f'(g(x))g'(x)$$

Якщо x - вектор $x=(x_1, \dots, x_n)$, то замість окремої похідної обчислюється градієнт:

$$\nabla_x f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}.$$

Застосувавши цей метод для кожного компонента окремо, в результаті вийде:

$$\nabla_x(f \cdot g) = \begin{pmatrix} \frac{\partial f \cdot g}{\partial x_1} \\ \vdots \\ \frac{\partial f \cdot g}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial g} \frac{\partial g}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial g} \frac{\partial g}{\partial x_n} \end{pmatrix} = \frac{\partial f}{\partial g} \nabla_x g.$$

Якщо g - вектор, то вийде:

$$\nabla_x f = \frac{\partial f}{\partial g_1} \nabla_x g_1 + \dots + \frac{\partial f}{\partial g_k} \nabla_x g_k = \sum_{i=1}^k \frac{\partial f}{\partial g_i} \nabla_x g_i$$

У результаті вийде формула матричного добутку:

$$\nabla_x f = \nabla_x g \nabla_g f,$$

$$\nabla_x g = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \dots & \frac{\partial g_k}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_1}{\partial x_n} & \dots & \frac{\partial g_k}{\partial x_n} \end{pmatrix}.$$

Тобто можна обрахувати похідні і градієнти будь-якої композиції функцій, включно і з векторними.

Отже, можна застосовувати формулу диференціювання композиції на графі або від витоків до стоків, отримуючи окремі похідні кожного вузла по одній змінній $\frac{dx}{dx}$, $\frac{da}{dx}$, ... , $\frac{df}{dx}$. Або від стоків до витоків, отримуючи окремі похідні по всіх проміжних вузлах $\frac{df}{df}$, $\frac{df}{de}$, ... , $\frac{df}{dx}$. Дійшовши до витоків графа - вершин x_1, \dots, x_n , отримуються окремі похідні $\frac{df}{dx_1}, \dots, \frac{df}{dx_n}$, тобто сам градієнт $\nabla_x f$.

Оскільки окремі похідні обчислюються у напрямку, протилежному до графу обчислень, то такий алгоритм називають алгоритмом зворотного поширення (back propagation).

2.4 Розробка структури згорткової нейронної мережі

При проектуванні структури згорткової нейронної мережі ми повинні дотримуватись основних рекомендацій до вибору параметрів згорткових нейромереж, які дозволять забезпечити найбільшу ефективність роботи нейромережі та достовірність одержуваних результатів. Дані рекомендації було складено на основі аналізу відомих найефективніших архітектур згорткових нейронних мереж [11].

Рекомендації щодо архітектури вхідного шару нейромережі:

- Вхідні двовимірні зображення мають бути квадратними.
- Ширина і висота вхідного зображення у пікселях має багаторазово ділитися без остачі на 2, аж до однозначних чисел.

Рекомендації щодо архітектури згорткового шару нейромережі:

- Параметри шару P (додавання нулів) і S (зміщення фільтрів) повинні обиратися так, щоб розміри згорткового шару у пікселях були цілочисельними і, бажано, рівними розмірності попереднього шару, за умови, що попередній шар не є згортковим шаром.
- Потрібно використовувати переважно фільтри малого розміру: 3×3 або 5×5 пікселів, однак, для великих вхідних зображень розміри фільтрів для першого згорткового шару можуть бути обрані 7×7 пікселів.
- Чим більший розмір вхідного шару нейромережі, тим більше фільтрів рекомендується використовувати у згорткових шарах, при цьому, рекомендовано, щоби кількість фільтрів для глибших згорткових шарів поступово збільшувалася.

Рекомендації щодо архітектури шару субдискретизації нейромережі:

- Бажано як опцію шару субдискретизації використовувати вибір максимального значення, замість розрахунку середнього значення яскравостей пікселів.
- Бажано використовувати розміри вікна шару субдискретизації 2 x 2 пікселі, тому що великі розміри вікна мають значний вплив на руйнування ознак, які виділяються згортковими шарами нейромережі.

Удосконалена структурна схема згорткової нейронної мережі SSD [10] показана на рис. 2.13. На вході нейромережі ми бачимо згорткову нейронну мережу VGG-16, що використовується як базова модель. Після базової моделі додатково додано деякі функціональні шари.

Задача розпізнавання медичних масок на обличчі вирішується за два етапи: 1) спочатку детектується наявність обличчя на даному зображенні або у потоці відео (виявлення обличчя), а потім у 2) частині розпізнається наявність або відсутність маски на обличчі (класифікація наявності/відсутності маски). Для того, щоб детектувати обличчя, було використано інструменти бібліотеки OpenCV. Недавня версія OpenCV має модуль глибокої нейромережі (DNN), який постачається з попередньо навченою згортковою нейромережею детектування обличчя (CNN). Нова модель покращує ефективність детектування обличчя порівняно з традиційними моделями. Після того, як отримано наявність обличчя, потрібно передбачити обмежувальне поле навколо нього і подати його до другої частини моделі нейромережі, щоб перевірити, чи є на обличчі маска чи ні. Друга частина моделі нейромережі тренується нами за допомогою набору даних, який складається із зображень як з маскою, так і без маски. Ми використовували спеціалізовані бібліотеки Keras разом із Tensorflow для навчання нашої моделі нейромережі.

Базова модель нейромережі, яка використовується у даній розробці – це VGG-16, що попередньо навчена на основі набору даних «ImageNet» (включає сотні тисяч зображень). Від базової моделі ми залишаємо тільки «голову» і доповнюємо її низкою наших самовизначених шарів мережі. Для

оптимізації мережі використано 1 усереднювальний шар пулінгу, 1 шар згладжування, 1 щільний (повнозв'язний) шар із вихідною формою (None, 128) та 1 активаційний шар ReLU, 1 50%-вий шар випадіння (дропауту) для оптимізації. Нарешті, ще один щільний (повнозв'язний) шар з вихідною формою (None, 2) та активаційною функцією Sigmoid. Загальна структура отриманої згорткової нейронної мережі наведена на рис. 2.13.

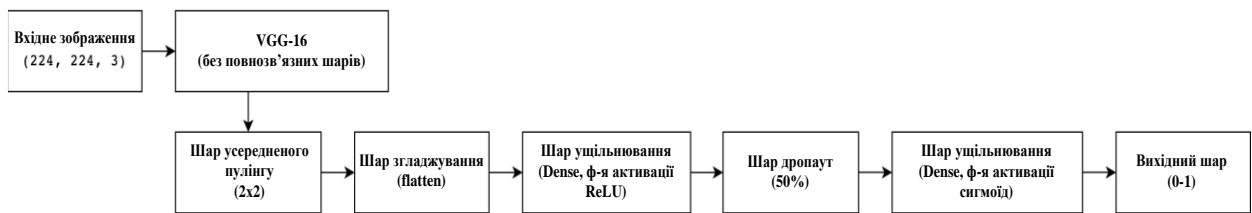


Рисунок 2.13 – Загальна структура розробленої згорткової нейронної мережі

Перша частина навчання передбачає зберігання усіх міток зображень у масиві NumPy, а відповідні їм зображення також переробляються у формат (224, 244, 3) для базової моделі. Використовується збільшення числа навчальних зображень (аугментація) - дуже корисна техніка, оскільки вона збільшує обсяг навчальних даних (кількість навчальних зображень) з абсолютно новою перспективою. Для аугментації виконувались такі випадкові перетворення зображень (рандомно): обертання до 20 градусів, збільшення та зменшення (масштабування) до 15%, зміна ширини або висоти зображення до 20%, застосування кута зсуву до 15 градусів проти годинникової стрілки, перевертання масиву входів по горизонталі, а точки поза межами входів заповнювалися з найближчого доступного пікселя вводу.

2.5 Розробка алгоритму розпізнавання медичних масок на обличчі

Алгоритм розпізнавання медичних масок на обличчі, розроблений на основі всього вище описаного у цьому розділі, представлено на рис. 2.14. Цей алгоритм складається з двох етапів: 1) етап навчання та 2) етап використання.



Рисунок 2.14 – Алгоритм розпізнавання медичних масок на обличчі

Зображення, що надходять на вхід системи, можуть мати різну розмірність у пікселях, тому вони спочатку масштабуються (тобто приводяться до єдиного розміру, який сприймається нейромережею). Ті зображення, на яких навчається нейромережа, крім цього, ще піддаються процесу аугментації (внесення шумів та невеликих афінних перетворень) з метою збільшення ентропії тренувальної інформації. Після цього відбувається навчання нейромережі VGG-16 на розпізнавання медичних масок. Після цього здійснюється збереження навченої нейромережі (серіалізація), а далі її можна використовувати для знаходження обличчя на вхідному зображенні та подальшого розпізнавання маски на цьому зображенні («У масці» чи «Без маски»). І врешті здійснюється виведення результату розпізнавання.

UML діаграма послідовностей розробленої інформаційної технології представлено на рис. 2.15.

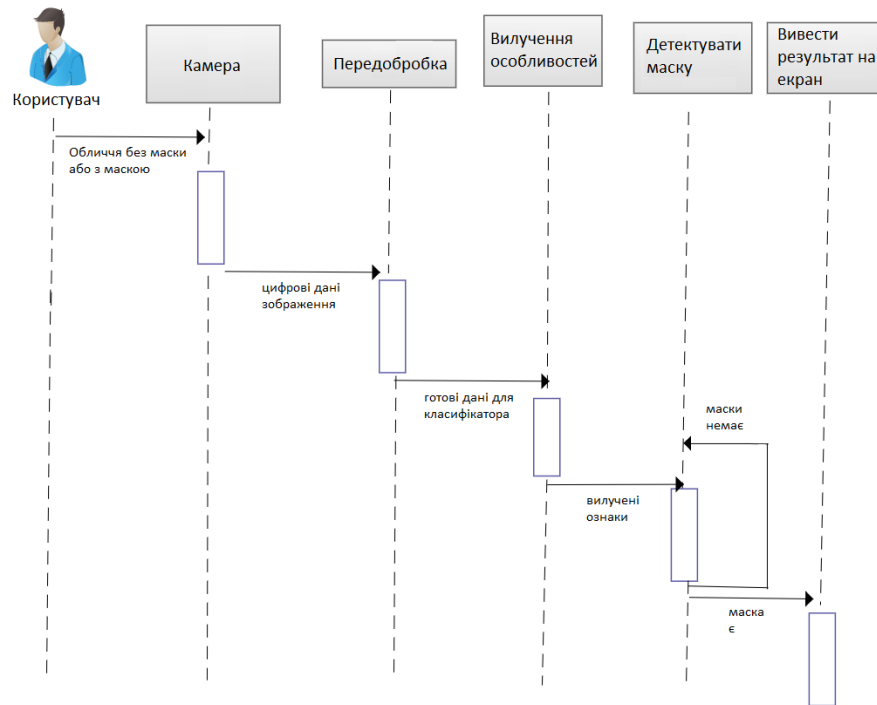


Рисунок 2.15 – UML діаграма послідовностей

У наступному розділі буде описано процес програмної реалізації розробленої інформаційної технології розпізнавання медичних масок на обличчі на основі модифікованої згорткової ейронної мережі.

2.6 Висновок до розділу 2

У розділі було розроблено структуру інформаційної технології розпізнаванні медичних масок на обличчі, обґрунтовано вибір як основи для розв'язання поставленої задачі задачі архітектури згорткової нейромережі SSD. Проаналізовано процес навчання згорткових нейронних мереж та розроблено модифіковану структуру згорткової нейронної мережі, яка доповнена шістьма новими визначеними в цій роботі шарами. Розроблено алгоритм роботи інформаційної технології розпізнавання медичних масок на обличчі на основі модифікованої згорткової ейронної мережі та UML діаграму послідовностей.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ МЕДИЧНИХ МАСОК НА ОБЛИЧЧІ

3.1 Обґрунтування вибору мови та програмних інструментів розробки

Суттєво полегшити процес реалізації будь-якого програмного додатку дозволить правильний вибір мови програмування та програмних інструментів розробки.

Вибір мови програмування для конкретного завдання залежить не тільки від її функціональних можливостей і потенційної сфери застосування, але і від великого числа інших факторів, таких як: наявність зручних середовищ програмування, тип ліцензії, обсяг професійної спільноти, наявність та якість спеціалізованих бібліотек і фреймворків.

Для програмної реалізації розробленої згорткової нейронної мережі та основних модулів API було обрано об'єктно-орієнтовану мову програмування Python [15].

Python – це об'єктно-орієнтована інтерпретована мова програмування високого рівня з динамічною типізацією. Структури даних високого рівня та динамічна семантика разом із динамічним зв'язуванням роблять її придатною для швидкодіючої розробки програм, а також як засіб для поєднання існуючих інструментальних компонентів. Python підтримує різноманітні модулі та пакети модулів, що сприяє модульності та повторному ефективному використанню коду. Інтерпретатор Python та стандартні бібліотеки є доступними як у скомпільованій, так і у вихідній формі на усіх основних платформах. У мові програмування Python підтримується декілька парадигм програмування, зокрема, об'єктно-орієнтоване, функціональне, процедурне та аспектно-орієнтоване.

Основною перевагою Python перед іншими мовами програмування є велика кількість спеціалізованих бібліотек та фреймворків для наукових

досліджень у галузі машинного навчання та штучного інтелекту, зокрема, у галузі штучних нейронних мереж. Також у Python є ефективні структури даних високого рівня і простий, але ефективний підхід до об'єктно-орієнтованого програмування. Це робить код простим для сприйняття та лаконічним, порівняно, наприклад, із мовою C++. Динамічна обробка типів, елегантний синтаксис Python, а також те, що це інтерпретована мова, роблять Python ідеальним для швидкої розробки прикладних програм та написання скриптів у багатьох галузях і на більшості платформ. Інтерпретатор мови Python може бути значно розширений функціями та типами даних, які були розроблені на C або C++ (або на іншій мові, яка викликається із C).

Як середовище програмування для Python було обрано IDE PyCharm, яке має низку зручностей, наприклад, таких як: автодоповнення, підсвічування синтаксису, інтеграція з системами контролю версій, менеджер пакетів і т. д. Також середовище програмування PyCharm має студентський тип ліцензії, що дозволяє студентам використовувати його безкоштовно.

3.2 Обґрунтування вибору фреймворків та бібліотек

Для програмної реалізації процесів ініціалізації та тренування згорткової нейромережі було застосовано такі бібліотеки як Keras та Tensorflow.

TensorFlow [16] - це безкоштовна програмна бібліотека з відкритим кодом для потоків даних та диференційованого програмування для цілого ряду різноманітних завдань. Це математична символічна бібліотека, яка також використовується для програм машинного навчання, таких як штучні нейронні мережі. Вона використовується як для наукових досліджень, так і для програмних розробок у Google, TensorFlow - це система другого покоління Google Brain. TensorFlow може працювати на кількох процесорах та графічних процесорах (з додатковими розширеннями SYCL та CUDA для обчислень загального призначення на графічних процесорах). Tensor Flow є

доступним на 64-розрядних платформах Linux, macOS, Windows та різних мобільних обчислювальних платформах, включно із iOS та Android. Його гнучка архітектура дозволяє легко організувати обчислення на різних платформах (процесори, графічні процесори та ін.), а також від робочих столів до мобільних пристроїв та кластерів серверів. Назва бібліотеки Tensor Flow походить від операцій, які виконуються у штучних нейронних мережах над багатовимірними масивами даних, які носять назву тензорів.

Keras [17] - це Application Programming Interface (API), розроблений для людей, а не для комп'ютерів. У Keras дотримано найкращих практик зменшення когнітивного навантаження: пропонуються прості та послідовні API, мінімізується число дій користувача, що необхідні для загальних випадків використання, а також забезпечується чіткість та зрозумілість повідомлень про помилки. У Keras також є об'ємна документація та навчальні посібники для розробників. Keras містить багато реалізацій часто використовуваних нейромережевих «будівельних блоків», таких як нейромережеві шари, цільові параметри, функції активації нейронів, оптимізатори та багато інструментів для спрощення роботи з різними даними, зображеннями та текстами для спрощення процесу кодування при написанні коду моделювання штучних нейронних мереж. Код фреймворку розміщений на GitHub, а форуми підтримки професійної спільноти містять сторінку з проблемами GitHub та канал Slack. Keras - мінімалістична бібліотека Python для глибокого навчання нейромереж, яка може працювати разом (зверху) Tensor Flow. Вона була розроблена для того, щоб зробити впровадження моделей глибокого навчання швидшим та простішим для розробок та досліджень. Вона працює на Python 2 або 3 і може також працювати на центральних процесорах та графічних процесорах. Keras розроблений і підтримувався інженером Google Франсуа Шолле, використовуючи при цьому чотири керівні принципи:

А) Модульність: Модель мережі можна розуміти як графік або послідовність. Усі елементи моделі глибокого навчання - це дискретні компоненти, які можна комбінувати довільно.

Б) Мінімалізм: Бібліотека забезпечує достатньо всього для досягнення результату, без максимальної читабельності та надмірностей.

В) Розширюваність: Нові компоненти навмисно легко додавати та використовувати в рамках, призначених для дослідників при дослідженні та випробуванні нових ідей.

Г) Python: Немає потреби в окремих файлах моделі зі спеціальними файловими форматами. Все є нативним для Python. Keras розроблений для модульності та мінімалізму, що дозволяє дуже швидко формувати моделі глибокого навчання та запускати їх поверх серверної частини TensorFlow або Theano.

Для роботи із зображеннями та відео було обрано широко відому та розповсюджену бібліотеку комп'ютерного зору OPENCV [18].

OpenCV (Open Source Computer Vision Library) — бібліотека комп'ютерного зору та машинного навчання з відкритим кодом. OpenCV було створено, щоб забезпечити загальну інфраструктуру для програм комп'ютерного зору та прискорити використання машинного сприйняття в комерційних продуктах. Будучи продуктом з ліцензією BSD, OpenCV дозволяє компаніям легко використовувати та змінювати код. Бібліотека містить понад 2500 оптимізованих алгоритмів, включаючи повний набір як класичних, так і найсучасніших алгоритмів комп'ютерного зору та машинного навчання. Ці алгоритми можна використовувати для виявлення та розпізнавання облич, ідентифікації об'єктів, класифікації дій людей у відео, відстеження рухів камери, відстеження рухомих об'єктів, вилучення 3D-моделей об'єктів, створення 3D-хмар точок із стереокамер, з'єднання зображень для отримання високої роздільної здатності. зображення цілої сцени, знаходити подібні зображення в базі даних зображень, видаляти червоні очі із зображень, зроблених за допомогою спалаху, стежити за рухами

очей, розпізнавати пейзаж і встановлювати маркери для накладання на нього доповненої реальності тощо. OpenCV має понад 47 тисяч користувачів спільноти та оціночна кількість завантажень перевищує 18 мільйонів.

Бібліотека широко використовується в компаніях, дослідницьких групах і урядових органах. Окрім таких відомих компаній, як Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, які використовують бібліотеку, є багато стартапів, таких як Applied Minds, Video Surf і Zeitera, які широко використовують OpenCV. Застосування OpenCV охоплює широкий діапазон: від з'єднання зображень вулиць, виявлення вторгнень у відеоспостереження в Ізраїлі, моніторингу шахтного обладнання в Китаї, допомоги роботам у навігації та підбиранні об'єктів у Willow Garage, виявлення нещасних випадків утоплення в басейні в Європі, використання інтерактивного мистецтва в Іспанії та Нью-Йорку, перевірка злітно-посадкових смуг на предмет сміття в Туреччині, перевірка етикеток на продуктах на заводах по всьому світу та швидке виявлення обличчя у Японії. Він має інтерфейси C++, Python, Java і MATLAB і підтримує Windows, Linux, Android і Mac OS. OpenCV здебільшого орієнтується на програми бачення в реальному часі та використовує переваги інструкцій MMX і SSE, якщо вони доступні. Зараз активно розробляються повнофункціональні інтерфейси CUDA і OpenCL. Існує понад 500 алгоритмів і приблизно в 10 разів більше функцій, які створюють або підтримують ці алгоритми. OpenCV написаний на C++ і має інтерфейс шаблону, який без проблем працює з контейнерами STL.

Комп'ютерний зір - це мультидисциплінарна наукова галузь, яка працює із цифровими зображеннями чи відеозаписами для автоматизації розв'язання завдань, які може виконувати людська зорова система. До завдань комп'ютерного зору відносять збір, обробку та аналіз інформації з цифрових зображень різного формату. Це тісно пов'язано з машинним навчанням та штучним інтелектом, так як комп'ютер має інтерпретувати «побачене» та аналізувати цифрове зображення аналогічно, як бачить це людина.

Для розробки графічного інтерфейсу програмної реалізації інформаційної технології було обрано бібліотеку Tkinter:

Tkinter - це стандартна бібліотека створення графічного інтерфейсу для Python. Python у поєднанні з Tkinter забезпечує швидкий і простий спосіб створення графічних інтерфейсів програм. Tkinter надає потужний об'єктно-орієнтований інструментарій для графічного інтерфейсу. Створення графічних інтерфейсів програм за допомогою Tkinter стає нескладним завданням.

3.3 Формування набору зображень для навчання та тестування нейромережі

Для навчання та тестування нейромережі в задачі розпізнавання медичних масок на обличчі потрібно сформувати відповідний набір зображень людей у масках і без них. Зображення для цього набору даних обличчя у масках і без масок було зібрано з наборів даних зображень, доступних у відкритому доступі, разом із деякими даними, взятими з Інтернету. Зображення у масках були отримані з набору даних розпізнавання обличчя в масках у реальному світі (RMFRD) [19] (див. рис. 3.1) і набору даних виявлення маски обличчя Larxel на Kaggle [20] (див. рис. 3.2).



Рисунок 3.1 – Зображення людей у масках із набору RMFRD



Рисунок 3.2 – Зображення людей у масках із набору Larxel (Kaggle)

У зображеннях RMFRD були переважно азіатські обличчя. Таким чином, зображення людей у масках з Larxel (Kaggle) були додані до набору даних, щоб усунути цю переважність. Набір RMFRD також містить зображення обличчя без масок. Однак, як згадувалося раніше, він містить переважну більшість азіатських обличчя. Тому було вирішено не використовувати ці зображення. Із набору даних Flickr-Faces-HQ (FFHQ), який представлено в [21], використовувалися зображення без масок (рис. 3.3).



Рисунок 3.3 – Зображення людей без масок із набору Flickr HQ

Отриманий набір даних також містить зображення неправильно одягнених масок або рук, які закривають обличчя, які класифікуються як обличчя без масок (див. рис. 3.4).



Рисунок 3.4 – Зображення з неправильно одягненими масками або руками, які закривають обличчя

Остаточний набір даних містить 1376 зображень, розділених на два класи:

Таблиця 3.1 – Склад набору зображень для навчання та тестування нейромережі

Назва класу	Опис	Кількість зображень
У масці	Обличчя з масками, які використовуються правильно	690
Без маски	Обличчя без масок або маски, що одягнені неправильно	686

У розділі 4 цей набір даних використовується для навчання та тестування згорткової нейронної мережі, на основі якої розроблено програму розпізнавання медичних масок на обличчі.

3.4 Програмна реалізація розпізнавання медичних масок на обличчі

Для навчання згорткової нейромережі спочатку необхідно завантажити набір зображень людей з масками на обличчі та без і на основі цього набору сформуванати кінцевий масив міток двох класів.

Після того, як набір зображень було завантажено, потрібно розбити його на навчальну та тестову множини (вибірки) у пропорції 80% до 20%.

```
x_, x_test, y_, y_test = train_test_split(x, y, test_size=0.2)
```

Далі необхідно створити об'єкт згорткової нейромережі з бібліотеки TensorFlow та ініціалізувати її:

```
cnn = MaskCNN(
sequence_length=x_train.shape[1],
num_classes=y_train.shape[1],
vocab_size=len(vocab_processor.vocabulary_),
embedding_size=params['embedding_dim'],
filter_sizes=list(map(int, params['filter_sizes'].split(","))),
num_filters=params['num_filters'])
```

Перший та другий параметри об'єкту MaskCNN – розмірність вхідних даних у пікселях (224x224). Третій параметр об'єкту MaskCNN залежить від числа зображень, на яких була навчена модель мережі. Четвертий параметр об'єкту MaskCNN вказує на довжину вектора, у який буде перетворюватися кожне зображення. П'ятий параметр об'єкту MaskCNN – масив з розмірністю використовуваних згорткових фільтрів. Шостий параметр об'єкту MaskCNN вказує на число фільтрів (32 для кожного згорткового шару мережі).

Після ініціалізації нейромережі починається процес тренування. Процес тренування складається з певної кількості ітерацій навчання. На кожній ітерації навчання обирається чергове зображення з тренувальної вибірки та подається на вхід нейромережі. Процес тренування нейромережі полягає у повторенні такого фрагменту коду:

```
optimizer = tf.train.AdamOptimizer(1e-3)
grads_and_vars = optimizer.compute_gradients(cnn.loss)
train_op = optimizer.apply_gradients(grads_and_vars,
global_step=global_step)
def train_step(x_batch, y_batch):
    feed_dict = {
        cnn.input_x: x_batch,
        cnn.input_y: y_batch,
        cnn.dropout_keep_prob: params['dropout_keep_prob']}
    _, step, loss, acc = sess.run([train_op, global_step, cnn.loss,
cnn.accuracy], feed_dict)
```

Ця функція на кожній ітерації навчання обчислює значення виходу нейромережі при поданні зображення з тренувальної вибірки та на основі порівняння значення виходу нейромережі з бажаним результатом (на основі співставлення міток класів) проводить корегування ваг нейромережі на основі методу навчання «стохастичний градієнтний спуск». Всі ці операції починають виконуватися після виклику функції `sess.run` всередині спеціалізованої бібліотеки `Tensorflow`.

Побудова аналізатора аргументів та синтаксичний аналіз аргументів виконується так:


```

ap = argparse.ArgumentParser()
ap.add_argument("-d", "--dataset", required=True,
    help="path to input dataset")
ap.add_argument("-p", "--plot", type=str, default="graph.png",
    help="path to output loss/accuracy plot")
ap.add_argument("-m", "--model", type=str,
default="face_mask_detection.h5",
    help="path to save output model")
args = vars(ap.parse_args())

```

Для завантаження нейромережі VGG16 використаємо:

```

baseModel = VGG16(weights="imagenet", include_top=False,
    input_tensor = Input(shape=(224, 224, 3)))

```

Побудова голови моделі мережі, яка буде розміщена поверх базової моделі нейромережі:

```

headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(4, 4))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(64, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="softmax")(headModel)

```

Розміщення головної моделі мережі зверху базової моделі (це і стане фактичною моделлю мережі, яку ми будемо навчати):

```

model = Model(inputs=baseModel.input, outputs=headModel)

```

Компіляція моделі мережі:

```
print("[INFO] compiling model...")
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="categorical_crossentropy", optimizer=opt,
              metrics=["accuracy"])
```

Навчання нейромережі:

```
print("[INFO] training head...")
H = model.fit_generator(
    trainAug.flow(trainX, trainY, batch_size=BS),
    steps_per_epoch=len(trainX) // BS,
    validation_data=(testX, testY),
    validation_steps=len(testX) // BS,
    epochs=EPOCHS)
```

Завантаження навченої моделі мережі на диск:

```
print("[INFO] saving ImageNet detector model...")
model.save(args["model"], save_format="h5")
```

Коли навчена модель нейромережі завантажена на диск, її вже можна використовувати в подальшому для розпізнавання медичних масок на обличчі на нових невідомих мережі зображеннях.

3.5 Висновок до розділу 3

У третьому розділі обґрунтовано вибір об'єктно-орієнтованої мови програмування високого рівня Python та середовища програмування (IDE) PyCharm. Для програмної реалізації процесів ініціалізації та тренування згорткової нейромережі було використано такі спеціалізовані бібліотеки як Tensorflow та Keras. Для організації роботи із зображеннями на фото та відео було обрано широко використовувану бібліотеку комп'ютерного зору OPENCV, а для реалізації графічного інтерфейсу програми - стандартна бібліотека Tkinter. У результаті було здійснено програмну реалізацію інформаційної технології розпізнавання медичних масок на обличчі на базі згорткової нейромережі.

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМИ РОЗПІЗНАВАННЯ МЕДИЧНИХ МАСОК НА ОБЛИЧЧІ

4.1 Тестування програми розпізнавання медичних масок на обличчі

Вікно головної сторінки розробленої програмної реалізації інформаційної технології розпізнавання медичних масок на обличчі на основі згорткової нейронної мережі наведено на рис. 4.1.

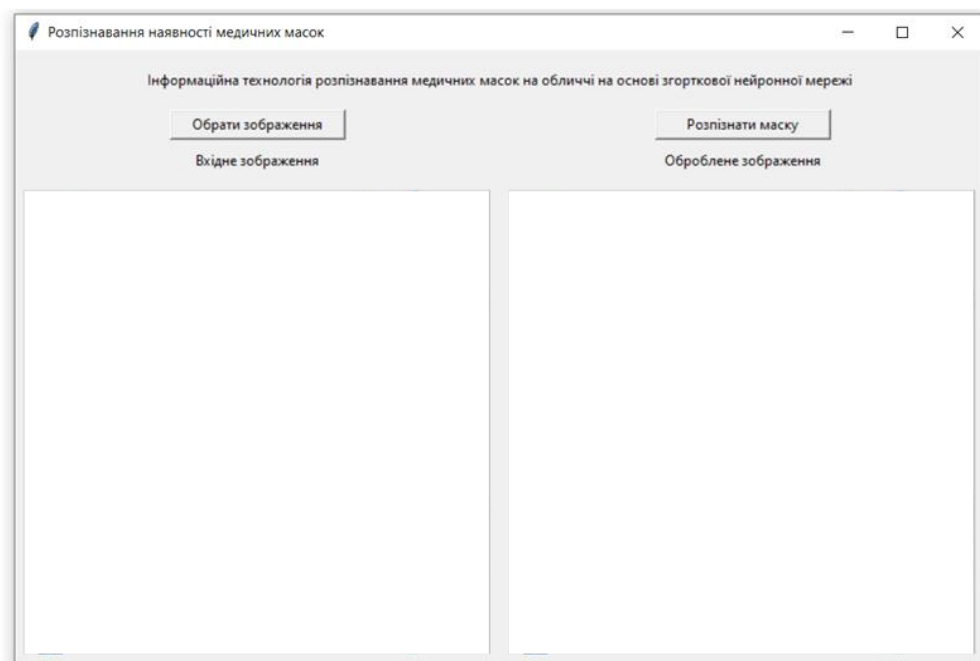


Рисунок 4.1 – Головна сторінка програми розпізнавання медичних масок на обличчі

Для того, щоб обрати потрібне для аналізу зображення, потрібно натиснути на кнопку Select Image. Після цього з'явиться вікно із файловою структурою, як показано на рис. 4.2, у якому користувач може виконати вибір файлу зображення.

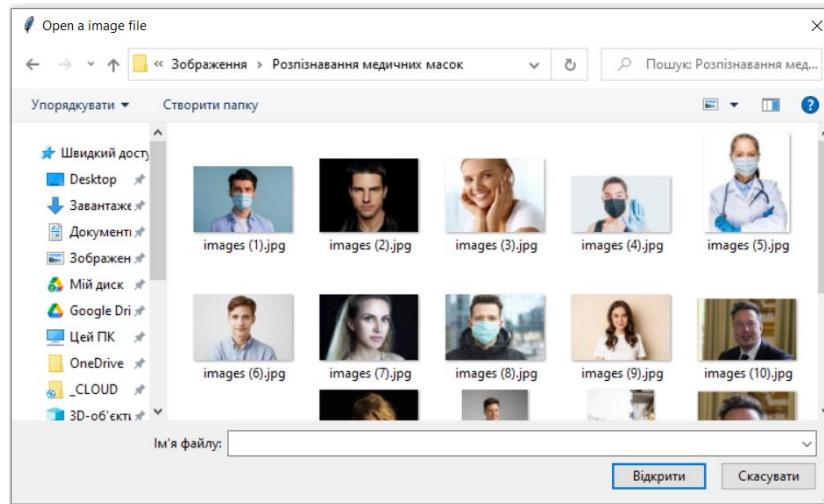


Рисунок 4.2 – Вибір зображення для розпізнавання маски

Після того, як мишкою обрано потрібне зображення, воно з'являється у обох робочих полях вікна програми як наведено на рис. 4.3 (тільки поки що без напису «Mask ON» або «NO Mask»).



Рисунок 4.3 – Відображення обраного зображення у вікні програми

Після цього потрібно натиснути на кнопку «Detect Mask». У результаті виконання програми на правому полі вікна під кнопкою «Detect Mask» зверху на зображенні буде додано текст, який буде вказувати «Mask ON» або «NO Mask» як наведено на рис. 4.4.

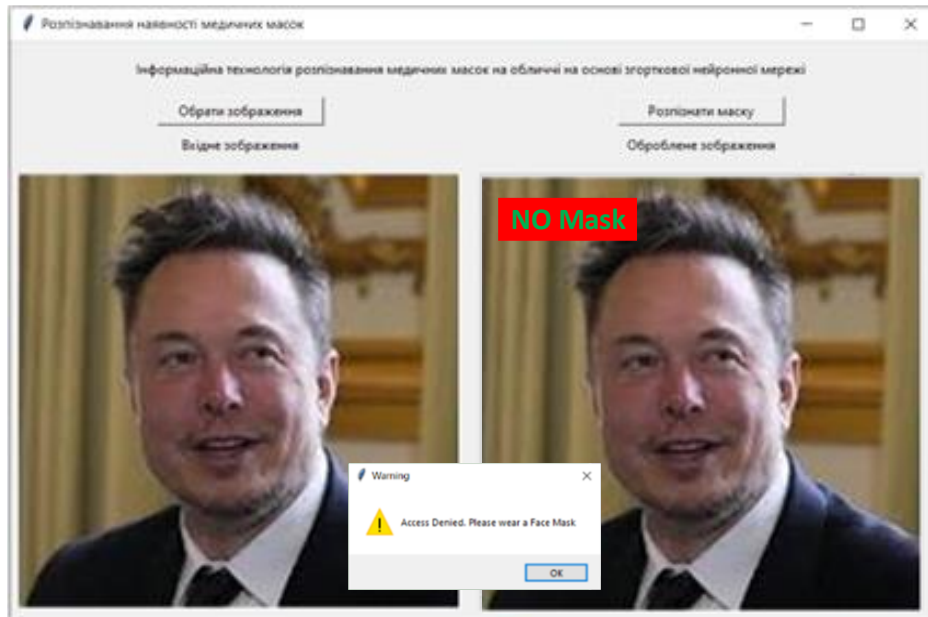


Рисунок 4.4 – Результат розпізнавання

Розроблене програмне забезпечення розпізнавання медичних масок на обличчі було натреновано за допомогою набору даних, що має 1376 зображень, з яких 686 зображень обличчя без медичної маски та 690 зображень обличчя з медичною маскою. Для навчальної вибірки було взято 1176 зображень, а для тестової – 200 зображень. Навчалася розроблена згорткова нейромережа комбінованої архітектури CNN-VGG16 за методом стохастичного градієнтного спуску, а вихідний шар її навчався за допомогою трансфертного методу прогнозування результату.

4.2 Аналіз результатів роботи програми розпізнавання медичних масок на обличчі

Для тестування розробленого програмного забезпечення та визначення достовірності його роботи було використано всього 200 зображень: 100 зображень людей у медичних масках та 100 зображень людей без масок. Результати тестування запропонованого програмного забезпечення наведено у табл. 4.1.

Таблиця.4.1 – Результати тестування запропонованого програмного забезпечення розпізнавання медичних масок на обличчі

	Результат розпізнавання	
	Mask ON	No Mask
На вході зображення з медичною маскою (загальна кількість зображень з маскою P=100)	Вірнопозитивний True positive – TP=97	Хибнонегативний False Negative – FN=3 помилка 2-го роду
На вході зображення без медичної маски (загальна кількість зображень без маски N=100)	Хибнопозитивний False positive – FP=1 помилка 1-го роду	Вірнонегативний True Negative– TN=99

Розрахуємо основні показники якості розробленого програмного забезпечення на основі даних табл. 4.1.

Достовірність:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% = \frac{97 + 99}{200} \times 100\% = 98\%$$

Точність.

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% = \frac{97}{97 + 1} \times 100\% = 99\%$$

Повнота.

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% = \frac{97}{97 + 3} \times 100\% = 97\%$$

Ми розрахували 3 основні показники якості системи розпізнавання медичних масок на обличчі і отримали: достовірність 98%, точність 99%, повнота 97%.

Для доведення досягнення поставленої мети дослідження треба порівняти ці показники якості із показниками програми-аналога (див табл 4.2). Таке порівняння наведено у табл. 4.3.

Таблиця.4.2 – Результати тестування програми-аналога (Face Mask Detector)

	Результат класифікації	
	Mask ON	No Mask
На вході зображення з медичною маскою (загальна кількість зображень з маскою P=100)	True positive – TP=92	False Negative – FN=8
На вході зображення без медичної маски (загальна кількість зображень без маски N=100)	False positive – FP=4	True Negative – TN=96

Таблиця 4.3 – Порівняння показників якості запропонованого програмного забезпечення із показниками програми-аналога (Face Mask Detector)

	Аналог (Face Mask Detector)	Запропонована програма
Достовірність	94 %	98 %
Точність	95,8 %	99 %
Повнота	92 %	97 %

Із табл. 4.3 видно, що розроблена програма має достовірність розпізнавання медичних масок на обличчі 98%, а аналогічна програма – 94%, тобто розроблена програма має на 4% вищу достовірність розпізнавання медичних масок на обличчі. Також із табл. 4.3 видно, що розроблена програма має вищу на 3,2% (99% проти 95,8%) точність та вищу на 5% (97% проти 92%) повноту, ніж програма-аналог. Тобто мета роботи досягнута – достовірність розпізнавання медичних масок на обличчі підвищена у розробленому програмному засобі на 4%.

4.3 Висновок до розділу 4

Було проведено тестування програмного забезпечення розпізнавання медичних масок на обличчі на основі згорткової нейронної мережі. Аналіз результатів роботи розробленої програми показав, що вона має достовірність розпізнавання медичних масок на обличчі на 4% кращу за аналог. Також розроблена програма має кращу за аналог на 3,2% (99% проти 95,8%) точність та кращу на 5% (97% проти 92%) повноту. Тобто мета роботи досягнута – достовірність розпізнавання медичних масок на обличчі покращена у розробленому програмному забезпеченні на 4%.

5 ЕКОНОМІЧНА ЧАСТИНА

Для успішного впровадження науково-технічної розробки критично важливо, щоб вона відповідала сучасним стандартам науково-технічного прогресу і враховувала економічні аспекти. Проведення оцінки економічної ефективності отриманих результатів науково-дослідної роботи становить ключову складову цього процесу.

Дослідження, представлене у магістерській роботі і приурочене до розробки та аналізу "Інформаційної технології розпізнавання медичних масок на обличчі на основі згорткової нейромережі", належить до проектів, спрямованих на виведення на ринок. Виведення на ринок може бути визначене під час самого виконання роботи і розглядається як етап комерціалізації науково-технічної розробки. Цей напрямок розглядається як пріоритетний, оскільки отримані результати можуть бути корисними для різних зацікавлених сторін і приносити економічні вигоди.

Проте для успішної реалізації цього процесу ключовим є пошук зацікавленого інвестора, який виявить інтерес до втілення даного проекту, та переконання його у доцільності вкладання інвестицій у цю розробку. З цією метою були визначені такі етапи виконання робіт:

1. Проведення комерційного аудиту науково-технічної розробки, включаючи визначення науково-технічного рівня та комерційного потенціалу.
2. Розрахунок витрат на реалізацію науково-технічної розробки.
3. Проведення розрахунку економічної ефективності впровадження та комерціалізації науково-технічної розробки для потенційного інвестора, а також обґрунтування економічної доцільності комерціалізації з точки зору інвестора.

5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою "Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі" є розширення функціональних можливостей програмного забезпечення для захисту файлів.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 5.1 [22].

Таблиця 5.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів

Продовження таблиці 5.1

Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці. Для оцінки науково-технічного рівня і комерційного потенціалу розробки експертами було запрошено трьох незалежних експертів з Вінницького національного технічного університету, кафедри «Комп'ютерні науки»: Колесницький О.К., к.т.н., професор, Арсенюк І. Р., к.т.н., доцент, Озеранський В.С., к.т.н., доцент.

Таблиця 5.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	Колесницький О.К.	Арсенюк І. Р.	Озеранський В.С.
	Бали:		
1.Технічна здійсненність концепції	3	3	3
2.Ринкові переваги (наявність аналогів)	2	1	2
3.Ринкові переваги (ціна продукту)	3	3	2
4.Ринкові переваги (технічні властивості)	3	2	3
5.Ринкові переваги (експлуатаційні витрати)	3	3	2
6.Ринкові перспективи (розмір ринку)	3	4	3
7.Ринкові перспективи (конкуренція)	2	2	2
8.Практична здійсненність (наявність фахівців)	4	4	4
9.Практична здійсненність (наявність фінансів)	3	3	3
10.Практична здійсненність (необхідність нових матеріалів)	4	4	4
11.Практична здійсненність (термін реалізації)	4	4	4
12.Практична здійсненність (розробка документів)	4	4	4
Сума балів	СБ ₁ =31	СБ ₂ =34	СБ ₃ =34
Середньоарифметична сума балів $СБ_c$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = \frac{31+34+34}{3} = 33$		

За результатами розрахунків, наведених в таблиці 5.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 5.3 [22].

Таблиця 5.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою "Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі" становить 33 бали, що, відповідно до таблиці 5.3 рівень комерційного потенціалу розробки вище середнього, що свідчить про комерційну важливість проведення даних досліджень.

Результатом магістерської кваліфікаційної роботи є програмний продукт, який буде корисний державним установам, магазинам, кафе, медичним та іншим закладам, де потрібно контролювати носіння медичних масок.

5.2 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему "Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі", під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

5.2.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [23]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.1)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p=21$ дні.

$$Z_o = 18000 \cdot 5 / 21 = 4091 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	18000	818,2	5	4091
Інженер-програміст	20000	909,1	45	40909
Тестувальник	19000	863,6	13	11227
Всього				56227

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (5.2)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$Z_{\text{дод}} = (56227) \cdot 11 / 100\% = 6185 \text{ грн.}$$

5.2.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{zn}}{100\%} \quad (5.3)$$

де H_{zn} – норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (56227+6185) \cdot 22 / 100\% = 13730,7 \text{ грн.}$$

5.2.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою "Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі".

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\text{в}j}, \quad (5.4)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

C_j – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

C_{ej} – вартість відходів j -го найменування, грн/кг.

Проведені розрахунки зведемо до таблиці.

Таблиця 5.5 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Вартість витраченого матеріалу, грн
Папір А4	175	1	175
Ручка	46	1	46
Диск оптичний CD	18	1	18
Flesh-пам'ять 64	386	1	386
Всього			625
З врахуванням коефіцієнта транспортування			687,5

5.2.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_6), які використовують при проведенні НДР на тему "Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі" відсутні.

5.2.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення в роботі відсутні.

5.2.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних)

необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення. Програмні засоби, які були використанні при написанні магістерської роботи є безкоштовними.

5.2.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_{б.}}{T_{в}} \cdot \frac{t_{вик}}{12}, \quad (5.5)$$

де $Ц_{б.}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{в}$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (45000 \cdot 1) / (2 \cdot 12) = 1875 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.6 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Комп'ютер	45000	2	1	1875,00
Приміщення лабораторії	310000	20	1	1291,67
Всього				3166,67

5.2.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{eni}}{\eta_i}, \quad (5.6)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 7,5$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,25 \cdot 295 \cdot 7,5 \cdot 0,5 / 0,8 = 345,7 \text{ грн.}$$

5.2.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему "Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі" належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cs} = (Z_o + Z_p) \cdot \frac{H_{cs}}{100\%}, \quad (5.7)$$

де H_{cb} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{cb} = 20\%$.

$$B_{cb} = (56227) \cdot 20 / 100\% = 11245,45 \text{ грн.}$$

5.2.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» відсутні.

5.2.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_s = (Z_o + Z_p) \cdot \frac{H_{is}}{100\%}, \quad (5.8)$$

де H_{is} – норма нарахування за статтею «Інші витрати», прийmemo $H_{is} = 50\%$.

$$I_s = (56227) \cdot 50 / 100\% = 28113,64 \text{ грн.}$$

5.2.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.9)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $H_{нзв} = 100\%$.

$$B_{нзв} = (56227) \cdot 100 / 100\% = 56227 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему "Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі". розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{доп} + Z_n + M + K_e + B_{спец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{сп} + I_e + B_{нзв}. \quad (5.10)$$

$$B_{заг} = 56227 + 6185 + 13730,7 + 687,5 + 3166,67 + 345,7 + 11245,45 + 28113,64 + 56227 = 175929,21 \text{ грн.}$$

Загальні витрати $ЗВ$ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (5.11)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо $\eta = 0,7$.

$$ЗВ = 175929,21 / 0,7 = 251327,44 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою "Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі" передбачають комерціалізацію протягом 3-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, прийmemo 1 особа

C_o – вартість послуги у році до впровадження інформаційної системи, прийmemo 4400,00 грн;

$\pm \Delta C_o$ – зміна вартості послуги від впровадження результатів, прийmemo зростання на 600,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 3-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [22]:

$$\Delta \Pi_i = (\pm \Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (5.12)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2021 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).

Прийmemo $\rho = 40\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (1 \cdot 600 + 4400 \cdot 750) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 640701,87 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (1 \cdot 600 + 4400 \cdot (750 + 700)) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 1239092,1 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (1 \cdot 600 + 4400 \cdot (750 + 700 + 660)) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 1802819,6 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^i}, \quad (5.13)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 18\%$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} III &= 640701,87/(1+0,18)^1 + 1239092,1/(1+0,18)^2 + 1802819,6/(1+0,18)^3 = \\ &= 2442549,78 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (5.14)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв} = 2$;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 251327,44 грн.

$$PV = k_{инв} \cdot 3B = 2 \cdot 251327,44 = 502654,87 \text{ грн.}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = III - PV \quad (5.15)$$

де III – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 2442549,78 грн;

PV – теперішня вартість початкових інвестицій 502654,87 грн.

$$E_{абс} = III - PV = 2442549,78 - 502654,87 = 1939894,91 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_g = \sqrt[T_{жс}]{1 + \frac{E_{абс}}{PV}} - 1, \quad (5.16)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_{жс}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 3 роки.

$$E_g = \sqrt[3]{1 + \frac{E_{абс}}{PV}} - 1 = (1 + 1939894,91 / 502654,87)^{1/3} - 1 = 1,06.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (5.17)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,1$;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,25.

$\tau_{мін} = 0,1 + 0,25 = 0,35 < 1,06$ свідчить про те, що внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою "Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі" доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_g}, \quad (5.18)$$

де E_6 – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 1,06 = 0,9 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

5.4 Висновки до розділу 5

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою "Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі" становить 33 бали, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього). Також термін окупності становить 9 місяців, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

ВИСНОВКИ

У роботі було розв'язано задачу розпізнавання медичних масок на обличчі на основі згорткових нейронних мереж. В ході аналізу предметної області сформульовано постановку задачі розпізнавання медичних масок на обличчі, проведено аналітичний огляд відомих методів розпізнавання зображень, які можна застосовувати до поставленої задачі. Було обґрунтовано доцільність використання нейромереж, а саме згорткових нейромереж для розпізнавання медичних масок на обличчі. Було знайдено та проведено аналіз різних програмних реалізацій програм для виявлення медичних масок на обличчі та обрано програму-аналог. Головним недоліком аналогів виявилось невисока достовірність розпізнавання медичних масок на обличчі, що поставило мету дослідження – підвищення достовірності розпізнавання медичних масок.

У другому розділі було розроблено структуру інформаційної технології розпізнавання медичних масок на обличчі на основі згорткової нейромережі, обґрунтовано вибір архітектури згорткової нейромережі SSD як основи для даної задачі. Розглянуто процес тренування згорткових нейромереж та розроблено модифіковану структуру згорткової нейромережі, яка була доповнена шістьма новими шарами. Розроблено алгоритм роботи програмної реалізації інформаційної технології розпізнавання медичних масок на обличчі та UML діаграму послідовностей.

У третьому розділі було обґрунтовано вибір об'єктно-орієнтованої мови програмування Python та середовища PyCharm. Для програмної реалізації процесів ініціалізації та тренування згорткової нейромережі було обрано такі спеціалізовані бібліотеки як Keras та Tensorflow. Для обробки зображень було обрано розповсюджену бібліотеку комп'ютерного зору OPENCV, а для проектування графічного інтерфейсу програми - бібліотеку Tkinter. У результаті було програмно реалізовано інформаційну технологію розпізнавання медичних масок на обличчі на базі згорткової нейромережі.

У четвертому розділі описано результати тестування програмного забезпечення розпізнавання медичних масок на обличчі на основі згорткової нейронної мережі. Аналіз результатів роботи розробленої програми показав, що вона має достовірність розпізнавання медичних масок на обличчі на 4% кращу за аналог. Також розроблена програма має кращу за аналог на 3,2% (99% проти 95,8%) точність та кращу на 5% (97% проти 92%) повноту. Тобто мета роботи досягнута – достовірність розпізнавання медичних масок на обличчі покращена у розробленому програмному забезпеченні на 4%.

У п'ятому розділі згідно проведених досліджень рівень комерційного потенціалу розробки за темою "Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі" становить 33 бали, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього). Також термін окупності становить 9 місяців, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сорока В. А., Паночишин Ю. М. «Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі», в Матеріали Міжнародна науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2024)», Вінниця, 2024, [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/viewFile/19514/16170>
2. Computer_vision [Електронний ресурс]. Режим доступу: https://en.wikipedia.org/wiki/Computer_vision.
3. Маєнраа Т. The Local Binary Pattern Approach to Texture Analysis — Extensions and Applications. Oulu University Press, 2003.
4. Review: RetinaNet — Focal Loss (Object Detection) [Електронний ресурс]. Режим доступу: <https://towardsdatascience.com/review-retinanet-focal-loss-object-detection-38fbaбafabe4>.
5. Yosinski J. How transferable are features in deep neural networks? [Електронний ресурс]. Режим доступу: <https://arxiv.org/pdf/1411.1792.pdf>.
6. Dougherty G. / Digital Image Processing for Medical Applications / G. Dougherty – Cambridge University Press. – ISBN 978-0-5218-6085-7.
7. COVID-19: Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning <https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/>.
8. Згорткова нейронна мережа [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Згорткова_нейронна_мережа.
9. Krizhevsky A. ImageNet Classification with Deep Convolutional Neural Networks [Електронний ресурс]. Режим доступу: <https://papers.nips.cc/paper/4824imagenet-classification-with-deep-convolutional-neural-networks.pdf>.

10. Inceptionv3 [Електронний ресурс]. Режим доступу: <https://se.mathworks.com/help/deeplearning/ref/inceptionv3.html>.
11. Згорткова нейронна мережа [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Згорткова_нейронна_мережа.
12. Глибокі нейронні мережі для вирішення завдань розпізнавання і класифікації зображення [Електронний ресурс]. – Режим доступу: <http://itcm.comp-sc.if.ua/2017/Sineglazov.pdf>.
13. Ren S. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/1506.01497>.
14. Гудфеллоу Я. Глибоке навчання / пер. з англ. А. А. Слинкіна. 2-е вид., випр. М.: ДМК Пресс, 2018. 652 с.: цв. ил.
15. Python [Електронний ресурс]. – Режим доступу: <https://www.python.org/>.
16. Tensorflow [Електронний ресурс]. – Режим доступу: <https://www.tensorflow.org/>.
17. Keras [Електронний ресурс]. – Режим доступу: <https://keras.io/>.
18. OpenCV [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/OpenCV>.
19. Wang, Z., Wang, G., Huang, B., Xiong, Z., Hong, Q., Wu, H., Yi, P., Jiang, K., Wang, N., Pei, Y., Chen, H., Miao, Y., Huang, Z., & Liang, J., 2020, Masked Face Recognition Dataset and Application, <https://arxiv.org/abs/2003.09093v2>.
20. Larxel, 2020 May, Face Mask Detection, Kaggle - Version 1, <https://www.kaggle.com/andrewmvd/face-maskdetection>.
21. Karras, T., Laine, S., and Aila, T., A StyleBased Generator Architecture for Generative Adversarial Networks, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 4396-4405, doi: 10.1109/CVPR.2019.00453.

22. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

23. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

24. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. К. Колесницький. – Вінниця : ВНТУ, 2023. – (PDF, 58 с.)

Додаток А (обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА
НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна технологія розпізнавання медичних масок на обличчі на основі згорткової нейромережі

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Unischek

Оригінальність 91,76% Схожість 8,24%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Ознайомлені з повним звітом подібності, який був згенерований системою Unischek щодо роботи.

Автор роботи



Сорока В.А.

Керівник роботи



Паночишин Ю.М.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

```

"Face Mask Recognition System"
from tkinter import *
import tkinter.messagebox
from PIL import ImageTk, Image
import cv2
import wget
from tkinter import filedialog
from tkinter import messagebox
import smtplib
from email.message import EmailMessage
from tensorflow.keras.models import load_model
from time import sleep
from keras.preprocessing.image import img_to_array
from keras.preprocessing import image
import tkinter.filedialog as tkFileDialog
import cv2
import numpy as np
from email.message import EmailMessage
path = ""
image1 = ""
image2 = ""
panelA = None
panelB = None
path2 = "./sample1.jpg"
#show_image1(path2)
def select_image():
    global panelA, panelB, image1, plate, plate_text
    global path
    path = tkFileDialog.askopenfilename()
    if len(path) > 0:
        path1=path
        path1=(path1.split('/'))
        imgOriginalScene = cv2.imread(path)

        image1 = cv2.resize(imgOriginalScene,(390,240))
        image1 = cv2.cvtColor(image1, cv2.COLOR_BGR2RGB)
        image1= Image.fromarray(image1)

```

```

image1= ImageTk.PhotoImage(image1)
image2 = cv2.imread(path2)
image2 = cv2.resize(image2,(390,240))
image2 = cv2.cvtColor(image2, cv2.COLOR_BGR2RGB)
image2= Image.fromarray(image2)
image2= ImageTk.PhotoImage(image2)
if panelA is None or panelB is None:
    panelA= Label(MainFrame, image = image1)
    panelA.grid(row=4,column=0,sticky=W)
    panelB = Label(MainFrame, image = image2)
    panelB.grid(row=4,column=1,sticky=W)

else:
    panelA.configure(image=image1)
    panelA.grid(row=4,column=0,sticky=W)
    panelB.configure(image=image2)
    panelB.grid(row=4,column=1,sticky=W)
    panelA.image = image1
    panelA.grid(row=4,column=0,sticky=W)
    panelB.image =image2
    panelB.grid(row=4,column=1,sticky=W)
def show_image1(path2):
    #print(path2)
    image2 = cv2.imread(path2)
    image2 = cv2.resize(image2,(390,240))
    image2 = cv2.cvtColor(image2, cv2.COLOR_BGR2RGB)
    image2= Image.fromarray(image2)
    image2= ImageTk.PhotoImage(image2)
    global panelA, panelB
    if panelA is None or panelB is None:

        panelA= Label(MainFrame, image = image1)
        panelA.grid(row=4,column=0,sticky=W)

# otherwise, update the image panels
else:
    # update the pannels
    panelA.configure(image=image1)
    panelA.grid(row=4,column=0,sticky=W)
    panelB.configure(image=image2)
    panelB.grid(row=4,column=1,sticky=W)

```

```

        panelA.image = image1
        panelA.grid(row=4,column=0,sticky=W)
        panelB.image =image2
        panelB.grid(row=4,column=1,sticky=W)
flag = False
def predict():
    if(path==""):
        tkinter.messagebox.showinfo("Image Not Selected","Please select an
image first \n ")
    else:
        print("[INFO] loading network...")
        model =load_model('./face_mask_detection.h5')
        labels = ['Mask ON','NO Mask'] #These labels will be used for showing
output
        start_point = (15, 15)
        end_point = (370, 80)
        thickness = -1
        print("[INFO] reading image...")
        frame = cv2.imread(path)
        roi_gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
        roi_gray = cv2.resize(frame,(224,224))
        roi = roi_gray.astype('float')/255.0
        roi = img_to_array(roi)
        roi = np.expand_dims(roi,axis=0)
        print("[INFO] classifying image...")
        preds = model.predict(roi)[0]
        label=labels[preds.argmax()]
        if(label=='NO Mask'):
            image = cv2.rectangle(frame, start_point, end_point, (0,0,255),
thickness)

cv2.putText(image,label,(30,60),cv2.FONT_HERSHEY_SIMPLEX,2,(255,255,255),3)
        cv2.imwrite("./Output/detected.jpg", frame)
        show_image1('./Output/detected.jpg')
        messagebox.showwarning("Warning", "Access Denied. Please wear a
Face Mask")

        msg = EmailMessage()
        msg['Subject'] = 'Subject - Attention! Someone violated our
facemask policy.'
        msg['From'] = 'manisekhon018@gmail.com'
        msg['To'] = 'manisekhon018@gmail.com'

```

```

        msg.set_content(
            'A person has been detected without a face mask. Below is the
attached image of that person.Please Alert the Authorities.\n'
            'From:\n'
            'Name- Manpreet Singh\n'
            'Branch-D4CSE-A2\n'
            'Univ Roll No.-1706472\n'
            'GNDEC,Ludhiana'
        )
    with open("Output/detected.jpg", "rb") as f:
        fdata = f.read()
        fname = f.name
        msg.add_attachment(fdata, maintype='Image', subtype="jpg",
filename=fname)
        with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp:
            smtp.login('manisekhon018@gmail.com', '$ekhoN123')
            smtp.send_message(msg)
        print('[INFO] alert mail Sent to authorities')
    if(label=='Mask ON'):
        image = cv2.rectangle(frame, start_point, end_point, (0,255,0),
thickness)

cv2.putText(image,label,(30,60),cv2.FONT_HERSHEY_SIMPLEX,1.6,(0,0,0),3)
    print("[INFO] saving image...")
    cv2.imwrite("./Output/detected.jpg",frame)
    #select_image()
    show_image1('./Output/detected.jpg')
    global flag
    flag = True

root = Tk()
root.title("GUI :Facemask Detection Alert System")
root.geometry("1200x720")
root.configure(background = 'Orange')
Tops = Frame(root,bg = 'red',pady = 1, width =1750, height = 90, relief =
"ridge")
Tops.grid(row=0,column=0)
Title_Label = Label(Tops,font=('Comic Sans MS',20,'bold'),text = "
Manpreet Singh, D4CSE-A2, 1706472 ' \n\t Facemask Detection Alert System using
Open-CV,\n\t\t Keras/Tensorflow and Deep Learning\t\t",pady=9,bg=
'orange',fg='blue',justify ="center")

```

```

Title_Label.grid(row=0,column=0)
MainFrame = Frame(root,bg = 'orange',pady=2,padx=2, width =1350, height = 100,
relief = RIDGE)
MainFrame.grid(row=1,column=0)
Label_1 =Label(MainFrame, font=('Comic Sans MS', 18,'bold'), text="\t
Detect whether a person is wearing a mask or not",padx=2,pady=2,
bg="orange",fg = "white",justify="center")
Label_1.grid(row=0, column=0)
Label_2 =Label(MainFrame, font=('arial', 15,'bold'), text="",padx=2,pady=2,
bg="orange",fg = "black")
Label_2.grid(row=1, column=0,sticky=W)
Label_9 =Button(MainFrame, font=('arial', 19,'bold'), text=" Select Image
",padx=2,pady=2, bg="blue",fg = "white",command=select_image)
Label_9.grid(row=2, column=0)
Label_9 =Button(MainFrame, font=('arial', 19,'bold'), text=" Detect Mask
",padx=2,pady=2, bg="blue",fg = "white",command=predict)
Label_9.grid(row=2, column=1,sticky=W)
Label_2 =Label(MainFrame, font=('arial', 10,'bold'), text="",padx=2,pady=2,
bg="orange",fg = "black")
Label_2.grid(row=3, column=0,sticky=W)
Label_3 =Label(MainFrame, font=('arial', 30,'bold'), text="
\t\t\t",padx=2,pady=2, bg="orange",fg = "black")
Label_3.grid(row=4, column=0)
Label_3 =Label(MainFrame, font=('arial', 10,'bold'), text="\t\t\t\t
",padx=2,pady=2, bg="orange",fg = "black")
Label_3.grid(row=5, column=1)
root.mainloop()


```


Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РОЗПІЗНАВАННЯ
МЕДИЧНИХ МАСОК НА ОБЛИЧЧІ НА ОСНОВІ ЗГОРТКОВОЇ
НЕЙРОМЕРЕЖІ

Виконав: студент 2-го курсу,
групи 2КН-22м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)


Сорока В. А.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

Паночишин Ю.М.
(прізвище та ініціали)

« 07 » 12 2023 р.

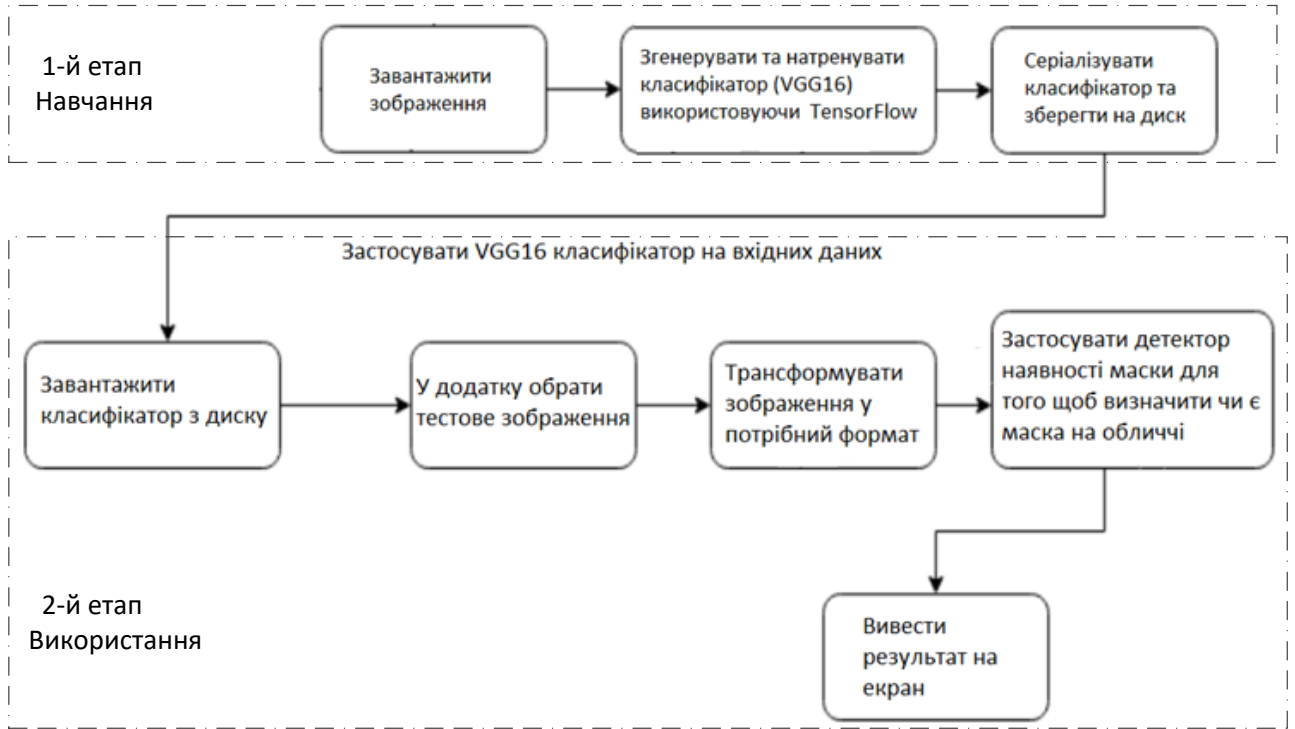


Рисунок В.1 –Загальний алгоритм роботи програми розпізнавання медичних масок на обличчі



Рисунок В.2 – Структура інформаційної технології розпізнавання медичних масок на обличчі на основі згорткової нейромережі

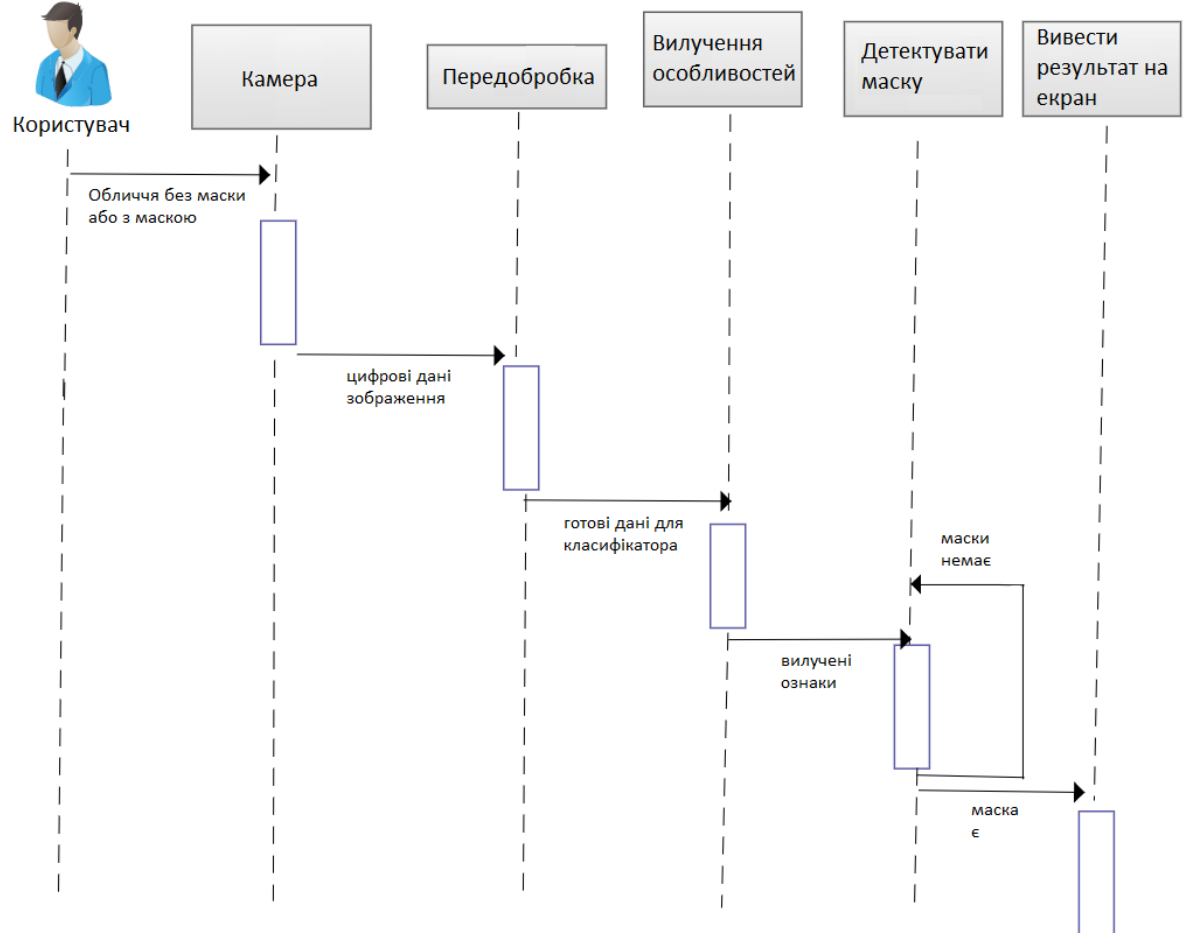


Рисунок В.3 – UML діаграма послідовностей

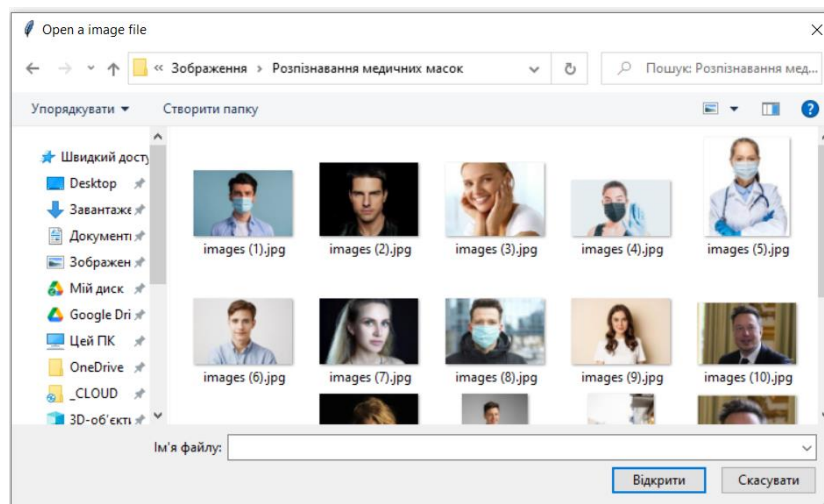
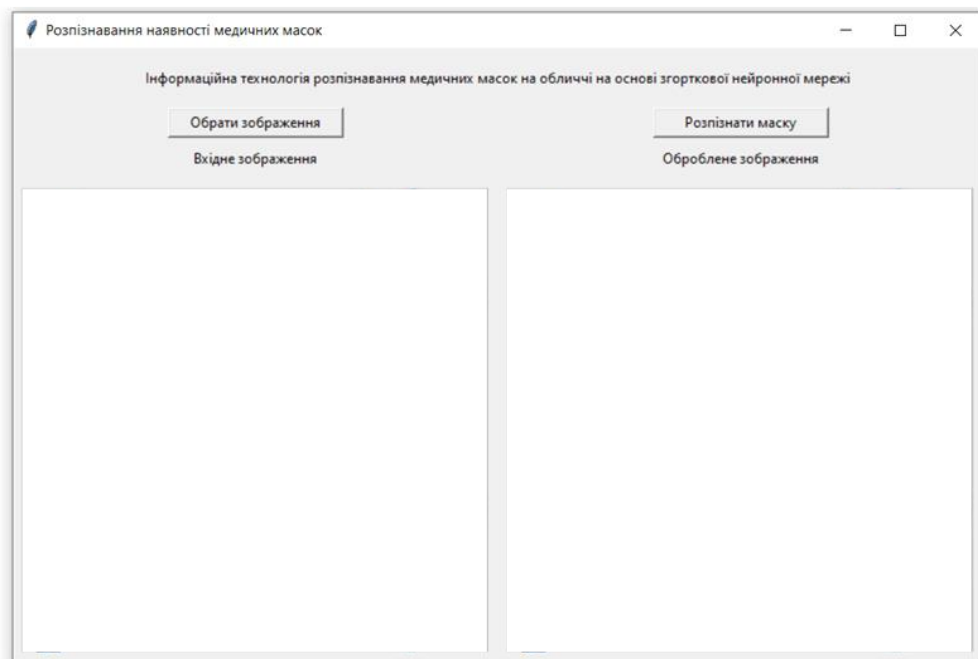


Рисунок В.4 – Стартові вікна програми розпізнавання медичних масок на обличчі

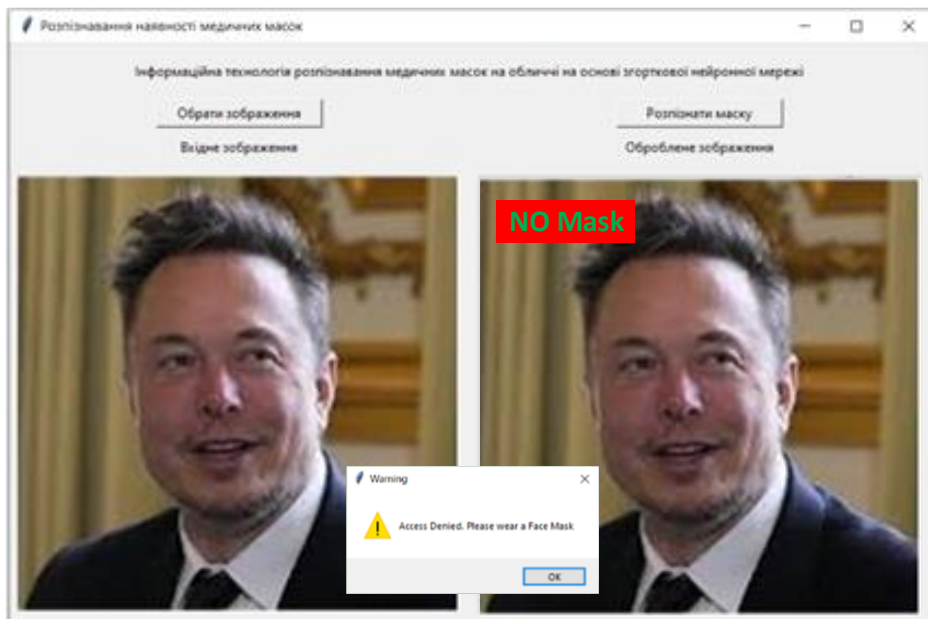


Рисунок В.5 – Результати роботи програми розпізнавання медичних масок на обличчі

	Результат розпізнавання	
	Mask ON	No Mask
На вході зображення з медичною маскою (загальна кількість зображень з маскою P=100)	Вірнопозитивний True positive – TP=97	Хибнонегативний False Negative – FN=3 помилка 2-го роду
На вході зображення без медичної маски (загальна кількість зображень без маски N=100)	Хибнопозитивний False positive – FP=1 помилка 1-го роду	Вірнонегативний True Negative– TN=99

Рисунок В.6 – Результати тестування запропонованого програмного забезпечення розпізнавання медичних масок на обличчі

	Аналог (Face Mask Detector)	Запропонована програма
Достовірність	94 %	98 %
Точність	95,8 %	99 %
Повнота	92 %	97 %

Рисунок В.7 – Порівняння показників якості запропонованого програмного забезпечення із показниками програми-аналога (Face Mask Detector)

Додаток Г (довідниковий)

Інструкція користувача

Для запуску програмного забезпечення розпізнавання медичних масок на обличчі на основі згорткової нейромережі необхідно запустити скрипт `gui_mask.py`. При цьому з'явиться вікно головної сторінки програмни, яке наведене на рис. Г.1.

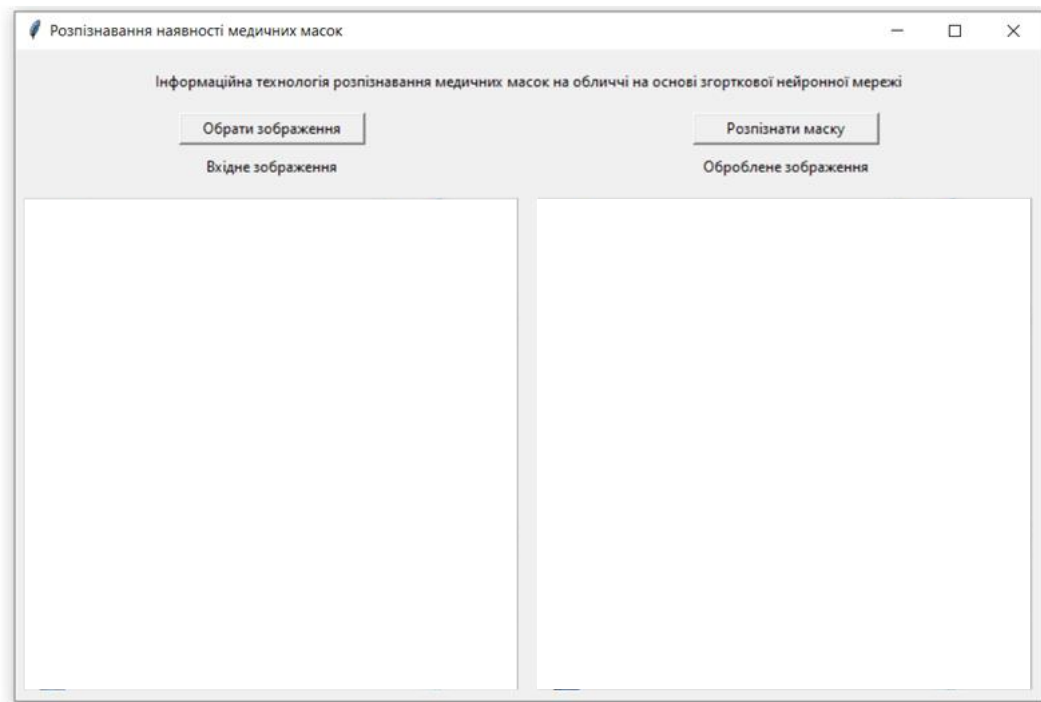


Рисунок Г.1 – Головна сторінка програми розпізнавання медичних масок на обличчі

Для того, щоб обрати потрібне для аналізу зображення, необхідно натиснути на кнопку `Select Image`. Після цього з'явиться вікно із файловою структурою, як показано на рис. Г.2, у якому користувач може виконати вибір файлу зображення.

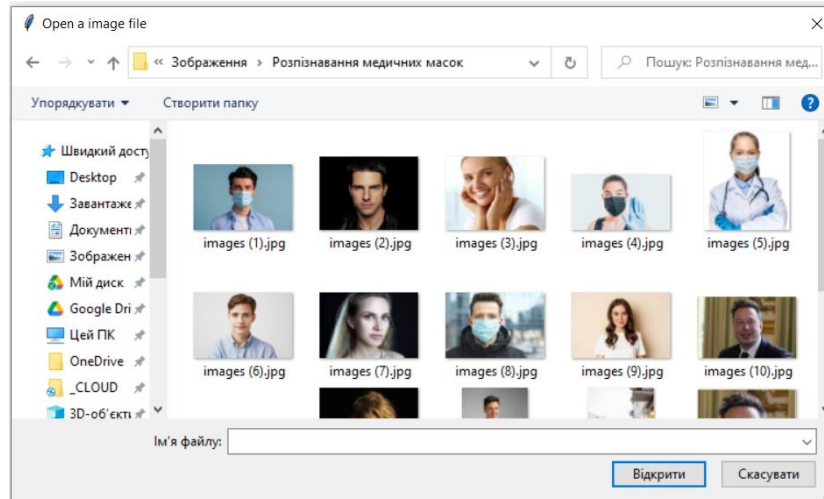


Рисунок Г.2 – Вибір зображення для розпізнавання маски

Після того, як мишкою обрано потрібне зображення, воно з'являється у обох робочих полях вікна програми як наведено на рис. Г.3 (тільки поки що без напису «Mask ON» або «NO Mask»).



Рисунок Г.3 – Відображення обраного зображення у вікні програми

Після цього потрібно натиснути на кнопку «Detect Mask». У результаті виконання програми на правому полі вікна під кнопкою «Detect Mask» зверху на зображенні буде додано текст, який буде вказувати «Mask ON» або «NO Mask» як наведено на рис. Г.4.

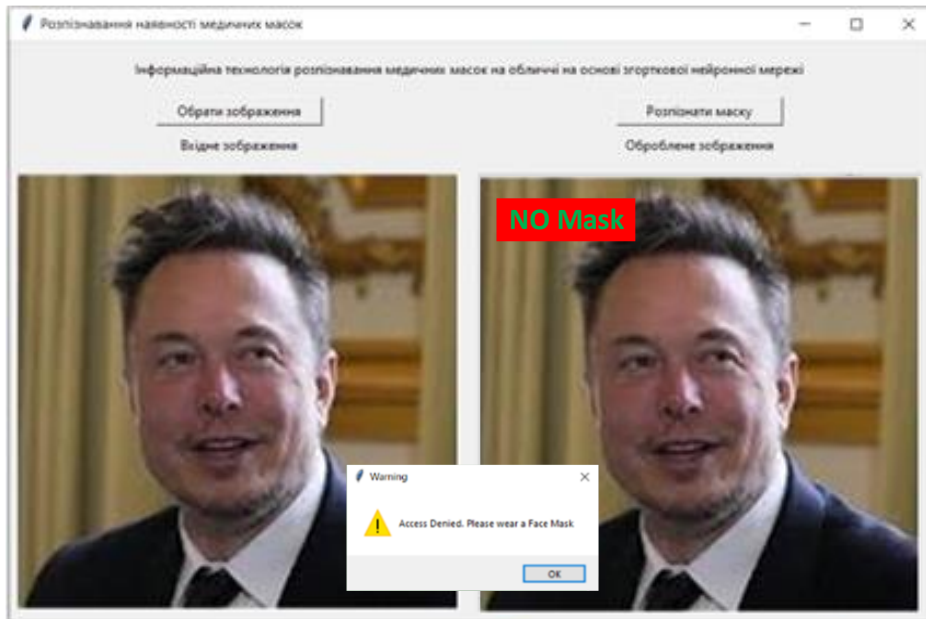


Рисунок Г.4 – Результат розпізнавання

Для завершення роботи з програмою достатньо закрити вікно додатку будь-яким відомим способом.