

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Спосіб розпізнавання медичної маски на обличчі  
з використанням алгоритмів глибокого навчання

Виконала: студентка IV курсу, групи СНС-41  
спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

\_\_\_\_\_ Кульчицька І.В.  
(підпис) (прізвище та ініціали)

Керівник \_\_\_\_\_ Никитюк В.В.  
(підпис) (прізвище та ініціали)

Нормоконтроль \_\_\_\_\_ Литвиненко Я.В.  
(підпис) (прізвище та ініціали)

Завідувач кафедри \_\_\_\_\_ Боднарчук І.О.  
(підпис) (прізвище та ініціали)

Рецензент \_\_\_\_\_  
(підпис) (прізвище та ініціали)

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
Боднарчук І.О.  
(підпис) (прізвище та ініціали)  
«\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки  
(шифр і назва спеціальності)

Студенту Кульчицькій Ірині Володимирівні  
(прізвище, ім'я, по батькові)

1. Тема роботи Спосіб розпізнавання медичної маски на обличчі  
з використанням алгоритмів глибинного навчання

Керівник роботи Никитюк Вячеслав Вячеславович, к.т.н.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «07» 02 2023 року № 4/7-133

2. Термін подання студентом завершеної роботи 21.06.2023 р.

3. Вихідні дані до роботи наукові літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

1. Аналіз предметної області.

2. Теоретична частина.

3. Практична частина.

4. Безпека життєдіяльності, основи охорони праці.

Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титулка. 2. Актуальність. 3. Мета, задачі дослідження. 4. Огляд окремих методів розпізнавання зображень. 5. Основні типи машинного навчання. 6. Нейронні мережі глибинного навчання. 7. Типова структура CNN для обробки зображення.

8. Процес розпізнавання обличчя. 9. Засоби програмної розробки

10. Набори тестових даних. 11. Алгоритм розробленої програми для навчання нейромережі.

12. Результати тестування моделі. 13. Основні результати проведеного дослідження

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці			

7. Дата видачі завдання \_\_\_\_\_ 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	08.02 – 12.02.23	Виконано
2.	Підбір джерел про розпізнавання зображень	13.02 – 26.02.23	Виконано
3.	Опрацювання джерел про розпізнавання зображень	27.02 – 15.03.23	Виконано
4.	Виконання дослідження щодо розпізнавання медичної маски на обличчі з використанням алгоритмів глибинного навчання	16.03 – 10.04.23	Виконано
5	Розроблення програмного коду	11.04 – 25.04.23	Виконано
6.	Оформлення розділу «Аналіз предметної області»	26.04 – 05.05.23	Виконано
7.	Оформлення розділу «Теоретична частина»	06.05 – 16.05.23	Виконано
8.	Оформлення розділу «Практична частина»	17.05 – 31.05.23	Виконано
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи хорони праці»	12.05 – 25.05.23	Виконано
10.	Оформлення кваліфікаційної роботи	20.05 – 03.06.23	Виконано
11.	Нормоконтроль	01.06 – 05.06.23	Виконано
12.	Перевірка на плагіат	06.06 – 09.06.23	Виконано
13.	Попередній захист кваліфікаційної роботи	07.06 – 12.06.23	Виконано
14.	Захист кваліфікаційної роботи	22.06.23	

Студент

\_\_\_\_\_ (підпис)

Кульчицька І.В.

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

Никитюк В.В.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Спосіб розпізнавання медичної маски на обличчі з використанням алгоритмів глибинного навчання // Кульчицька Ірина Володимирівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, кафедра комп'ютерних наук, група СНс-41 // Тернопіль, 2023 // С. – 60, рис. – 27, табл. – 1, слайдів – 13, бібліогр. – 35.

**Ключові слова:** глибинне навчання, нейронна мережа, розпізнавання обличчя, захисна маска

Кваліфікаційна робота присвячена розробці способу розпізнавання медичної маски на обличчі із використанням алгоритмів глибинного навчання.

У першому розділі проаналізовано предметну область дослідження, виконано постановку проблеми, описані основні методи розпізнавання особи на зображенні (Віоли-Джонса, локальних ознак, нейромережі).

У другому розділі роботи наведено особливості навчання з та без учителя, а також глибинного навчання. Досліджено нейромережі глибинного навчання. Докладно проаналізовано згорткові нейронні мережі на предмет їх використання в роботі. Також наведено алгоритм процесу розпізнавання обличчя на відео. Обґрунтовано використання методів попередньої обробки зображення. Описані методи розпізнавання обличчя (за геометрією обличчя та за будовою черепа).

У третьому розділі наведено практична складова дослідження. Описані основні програмні засоби, використані в роботі. Показано процес попереднього навчання нейромережі. Здійснено вибір мережі та датасету, який застосовувався для тестування її роботи. Розроблено програмний засіб, наведено алгоритм його роботи для навчання нейромережі. Представлені результати тестування моделі.

У четвертому розділі розглянуто важливі питання безпеки життєдіяльності та основ охорони праці.

## ANNOTATION

Method of Medical Mask Recognizing on the Face Using Deep Learning Algorithms // Kulchytska Iryna // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science // Ternopil, 2023 // P. - 60, Fig. - 21, Table - 1, Slide - 13, References - 35.

**Keywords:** deep learning, neural network, face recognition, safety mask

Thesis deals with the development of a software method for recognizing a medical mask on the face using deep learning algorithms.

In the first chapter, the subject area of the research is analyzed, the problem statement is made, and the main methods of face recognition in the image are described (Viola-Jones, local features, neural networks).

The second section of the work gives the features of learning with and without a teacher, as well as in-depth learning. Deep learning neural networks have been studied. Convolutional neural networks are analyzed in detail for their use in work. The algorithm of the face recognition process in the video is also given. The use of image preprocessing methods is substantiated. Face recognition methods are described (by the geometry of the face and the structure of the skull).

The third chapter presents the practical component of the research. The main software tools used in the work are described. The process of preliminary training of the neural network is shown. The network and dataset used for testing its operation were selected. A software tool has been developed, the algorithm of its operation for training a neural network is given. The results of model testing are presented.

The fourth chapter deals with important issues of life safety and the basics of labor protection.

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

CNN (convolutional neural network) – згорткова нейронна мережа.

CV (computer vision) – комп'ютерний зір.

OpenCV (Open Source Computer Vision Library) – бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом.

ReLU (rectified linear unit) – функція активації нейрона.

SSD – Single Shot Multibox Detector.

БД – база даних.

ВООЗ – Всесвітня Організація Охорони Здоров'я.

МГК – метод головних компонент.

МГС – метод градієнтного спуску.

МН – машинне навчання.

ПЗ – програмне забезпечення.

РО – розпізнавання обличчя.

ФА – функція активації.

ШІ – штучний інтелект.

ШНН – штучна нейронна мережа.

## ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Постановка проблеми .....	10
1.2 Опис предметної області .....	12
1.2.1 Метод Віоли-Джонса .....	13
1.2.2 Аналіз локальних ознак.....	15
1.2.3 Нейромережні методи.....	18
РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА .....	20
2.1 Контрольоване та неконтрольоване навчання .....	20
2.2 Глибинне навчання .....	21
2.3 Нейронні мережі глибинного навчання.....	22
2.4 Перенавчання нейромереж та методи запобігання перенавчанню.....	23
2.5 Згорткові нейронні мережі.....	24
2.5.1 Загальні поняття .....	24
2.5.2 Операція згортки .....	27
2.5.3 Функція активації.....	28
2.5.4 Пулінг .....	29
2.5.5 Типова структура .....	29
2.5.6 Переваги та недоліки .....	31
2.6 Процес РО .....	32
2.7 Методи попередньої обробки зображень .....	33
2.8 Методи РО .....	34
РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА .....	35
3.1 Використання програмних засобів.....	35
3.2 Попереднє навчання моделі .....	36
3.3 Мережа MobileNetV2.....	37
3.4 Використовуваний набір даних .....	39
3.5 Виявлення обличчів із використанням OpenCV .....	41
3.6 Алгоритм програми для навчання нейромережі.....	42

3.7 Результати тестування моделі.....	45
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .....	49
4.1 Долікарська допомога при ураженні електричним струмом.....	49
4.2 Вимоги ергономіки до організації робочого місця оператора ПК.....	51
ВИСНОВКИ.....	55
ПЕРЕЛІК ДЖЕРЕЛ .....	56
ДОДАТКИ	



## ВСТУП

**Актуальність теми.** З кінця 2019 року COVID-19 негативно вплинув на світ. Пандемія виникла в Ухані, Китай. ВООЗ в березні 2020 року оголосила її смертельним захворюванням. Усі медичні працівники та організації, практикуючі лікарі та дослідники шукають відповідні вакцини та ліки для боротьби з пандемією, розробляються різні вакцини.

При поточній ситуації у світі, коли коронавірусом COVID-19 заражається все більше людей, потреба у захисних медичних масках на обличчі стала ще більш актуальною [1]. В автобусах, у місцях громадського харчування, у відкритих навчальних закладах та інших масових місцях заборонено перебування без маски. Однак, не всі і не завжди це перевіряють.

Виявлення маски для обличчя стало популярним через пандемію, яка вимагає від людини носити маски для обличчя, зберігати соціальне дистанціювання та використовувати дезінфікуючі засоби для рук, щоб мити руки [2]. У той час як інші проблеми соціального дистанціювання та санітарної обробки були вирішені досі, проблема виявлення лицьових масок ще не вирішена належним чином. Що стосується цієї пандемії можна значно зменшити поширення вірусу та серйозність захворювання, знижуючи ризик передачі цього смертельного вірусу від інфікованої людини до здорової завдяки маскам та соціальній дистанції. Проте проблема розпізнавання маски з'явиться в майбутньому знову.

Виявлення маски на обличчі виявилось складною задачею в сфері обробки зображень та CV [3]. Детектування осіб має різні варіанти використання, від PO до фіксації рухів особи, коли вимагається, щоб особа була виявлена з дуже високою точністю. Завдяки швидкому прогресу в галузі алгоритмів МН, проблеми, пов'язані з технологією виявлення за маскою обличчя непогано досліджені. Сьогодні це завдання найбільш актуальне, тому що потрібне виявлення осіб не тільки на статичних зображеннях та відео, а й під час перевірки та спостереження в реальному часі, що значно ускладнює виконання.

**Мета роботи** – розробити спосіб для розпізнавання маски на обличчі з

застосуванням алгоритмів глибинного навчання.

**Для досягнення мети виділено ряд завдань:**

– проаналізувати методи, що вирішують завдання розпізнавання та/або виявлення осіб;

– виконати опис вибраного методу розпізнавання зображень;

– побудувати та описати роботу алгоритму CNN;

– провести опис алгоритму з передвиборною мережею;

– розробити ПЗ для розпізнавання маски на обличчі.

**Практичне значення** цієї роботи полягає в тому, що в результаті буде отримано модель нейромережі для розпізнавання на обличчі особи захисної медичної маски.

# РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Постановка проблеми

Питання безпеки майже завжди постає перед усіма людьми, а в сучасному світі воно майже завжди торкається компанії. Вирішення цього питання надається та вдосконалюється зі зростанням свободи доступу до інформації та технологій, а також рівня залучення людей до цих технологій. Наприклад, зараз обов'язки охоронця значно скоротилися через застосування камер спостереження. Сьогодні одним із найбільш затребуваних підходів до здійснення безпеки є використання камер. Важко собі уявити організацію, яка б відмовилася від них. Переваги використання камер очевидні, але для більш ефективного їх застосування поєднуються можливості камер записувати відео з можливостями інтелекту людини – розпізнати хто чи що на цьому відео. Камери використовуються також для контролю якості роботи співробітників та загальної ситуації на виробництві компанії [4].

Поряд із збільшенням споживання систем відеомоніторингу розвиваються й технології, що використовуються у системах. Так, наприклад, дедалі частіше можна зустріти спільну роботу відеокамер і штучного інтелекту, який дозволяє розпізнавати об'єкти на відео. Багато таких систем використовуються не тільки у сфері безпеки, а й, наприклад, у системах автопілота. Штучний інтелект надає великі можливості для відеоспостереження, серед яких є розпізнавання осіб, людей та об'єктів, а також деталей на кшталт номерного знаку автомобілів. Це дозволяє реалізувати всі чотири функції відеоаналітики: виявлення, стеження, розпізнавання та прогнозування.

Для розпізнавання певних груп осіб та надання їм доступу, на підприємствах встановлюють електронні турнікети з ідентифікацією відбитків пальців або зі зчитувачами карток, але на сьогоднішній день у зв'язку з швидким поширенням біометричних технологій компанії все частіше переходять на інші методи розпізнавання, точніші та зручніші. Наприклад, розпізнавання осіб в реальному часі за відеопослідовністю [5].

Поряд з розвитком методів розпізнавання та стрімкого зростання застосування біометричних систем, методи ідентифікації по обличчю виявляють свої недоліки та стають уразливими.

На сьогоднішній день, щоб обмежити поширення COVID-19, застосування масок стає звичайним явищем у громадських місцях. Крім того, багато постачальників державних послуг вимагають, щоб люди були у масках згідно з правилами (наприклад, закривали рота та носа) під час відвідування певних місць або території. Ця ситуація у всьому світі підштовхнула на дослідження автоматизованих (заснованих на CV) методів виявлення масок, які можуть допомогти контролювати суспільну поведінку та запобігти поширенню пандемії.

Маски є захисним засобом, який використовують люди у повсякденному житті для захисту від різних хвороб. Така ситуація робить звичайну технологію контролю доступу неефективною у багатьох випадках, а саме таких як контроль доступу до приміщень, контроль доступу з використанням ідентифікації по обличчю, підрахунок відвідуваності, перевірка безпеки на вокзалах тощо. У зв'язку з великим попитом на технології розпізнавання та ідентифікації, особливо в сьогоднішні дні, виникла велика необхідність поліпшення методів розпізнавання для існуючих способів технології розпізнавання осіб, стосовно осіб, які використовують захисні маски.

Ця робота ґрунтується на використанні методів МН, а також CV з використанням ШНМ. Застосування такої технології можна знайти у системах контролю доступу в приміщеннях, підрахунку відвідуваності, перевірки безпеки на вокзалах та ін [6]. Більш конкретна сфера застосування цієї технології — виявлення людей, які не дотримуються карантинних заходів, соціального дистанціювання та не мають захисних медичних масок, а також подальшої заборони їм доступу до приватних чи громадських місць.

В рамках соціальної спрямованості, результати дозволяють значною мірою спростити та прискорити процес контролю доступу чи допуску на певну територію чи громадське місце, таких як: метро, аеропорт, університет тощо.

## 1.2 Опис предметної області

У роботі будь-якого алгоритму розпізнавання чи виявлення обличчя слід виділити два логічні блоки:

- екстрактор характерних ознак;
- механізм класифікації.

Дія екстрактора полягає в виділенні з величезного потоку вхідних даних корисної для класифікатора інформації. При ідентифікації особи цією інформацією можуть бути характеристики однозначно визначених ознак (наприклад, які застосовуються у криміналістиці - відносно розташування очей, брів, губ та носа). Класифікатор при ухваленні рішення про призначення мітки класу об'єкту, що розпізнається, повинен керуватися саме цими ознаками. Вибір ознак є найважливішим завданням. Очевидно, що при їх виборі враховуються найбільш унікальні властивості, оскільки за ними можна достовірно судити про належність об'єкта до того чи іншого класу. Існує безліч різних підходів до одержання ознак класу.

Першими у ознаковій категорії були геометричні методи. У ході розпізнавання виділялися ключові точки з використанням детекторів основних частин обличчя та аналізувалося їхнє взаємне розташування. Зрозуміло, це досить просто, порівняно з тим, що ми маємо в наші дні. Треба враховувати, такі методи застосовувалися тоді, коли обчислювальні можливості комп'ютерних систем не дозволяли використовувати складніші алгоритми.

Набагато ефективнішими є ознакові методи розпізнавання на основі карти крайових ліній. При цьому у вихідному зображенні виділяють краї з різними пороговими значеннями і отриманий набір порівнюють з картою країв моделі за допомогою обчислення відстані по Гаусдорфу [7]. Ознаками класів цього є яскравісні перепади і неоднорідності.

Деякі з описаних нижче методів вирішують проблему розпізнавання осіб. У кваліфікаційній роботі потрібно вирішити завдання дещо простіше, а саме розпізнати зображення та виявити обличчя. Однак, описані методи дозволяють краще зрозуміти алгоритм розпізнавання зображень в цілому.

### 1.2.1 Метод Віолі-Джонса

Названий на честь двох дослідників CV, які запропонували його у 2001 році - Паоло Віолі та Майкла Джонса. Незважаючи на те, що цей метод відносно застарів, він досить потужний і його застосування виявилось виключно помітним при виявленні осіб в реальному часі. Цей алгоритм може виявляти особи реальному часі з вражаючою швидкістю [8]. Цей метод має множини різноманітних реалізацій, зокрема у складі OpenCV [9].

Використовується для відшукування об'єктів на зображенні в реальному часі і при цьому володіє надзвичайною низькою імовірністю помилкового спрацьовування. Як ознаки для алгоритму розпізнавання застосовано авторами ознаки (вейвлети) Хаара [10].

Загальна схема цього методу показана на рис. 1.1.

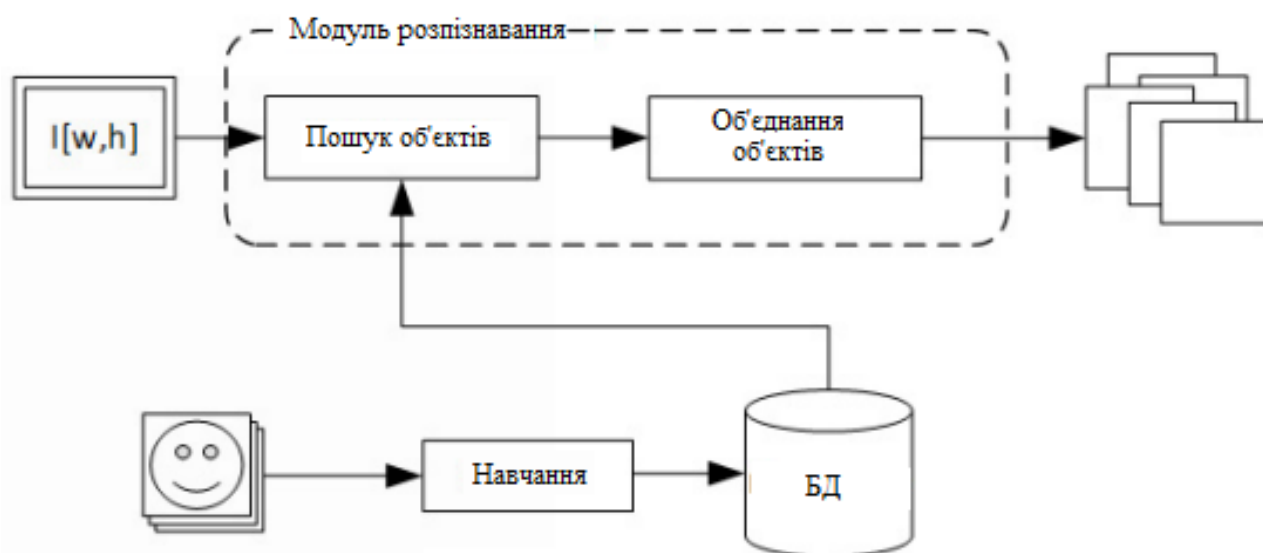


Рисунок 1.1 - Схема методу Віолі-Джонса [10]

У цьому алгоритмі використовується AdaBoost («бустинг-посилення») - алгоритм посилення класифікаторів (необхідний для вибору найбільш відповідних ознак об'єкта на зображенні), а також застосовуються каскади ознак (відкидаються вікна, де не знайдено об'єкт).

Результатом роботи алгоритму є класифікатор (функція прогнозування,

функція гіпотез), званий «сильним класифікатором» [11]. Сильний класифікатор складається з лінійних комбінацій "Слабких класифікаторів" (кращих характеристик). З високого рівня, щоб знайти їх, алгоритм запускається для  $T$  ітерацій, де  $T$  - кількість слабких класифікаторів, які потрібно знайти. На кожній ітерації алгоритм знаходить частоту помилок для всіх функцій, а потім вибирає функцію із найнижчою частотою помилок для цієї ітерації. AdaBoost відбирає сукупність слабких класифікаторів для їх об'єднання та надає кожному з них свою вагу. Ця виважена комбінація є сильним класифікатором (див. рис. 1.2).

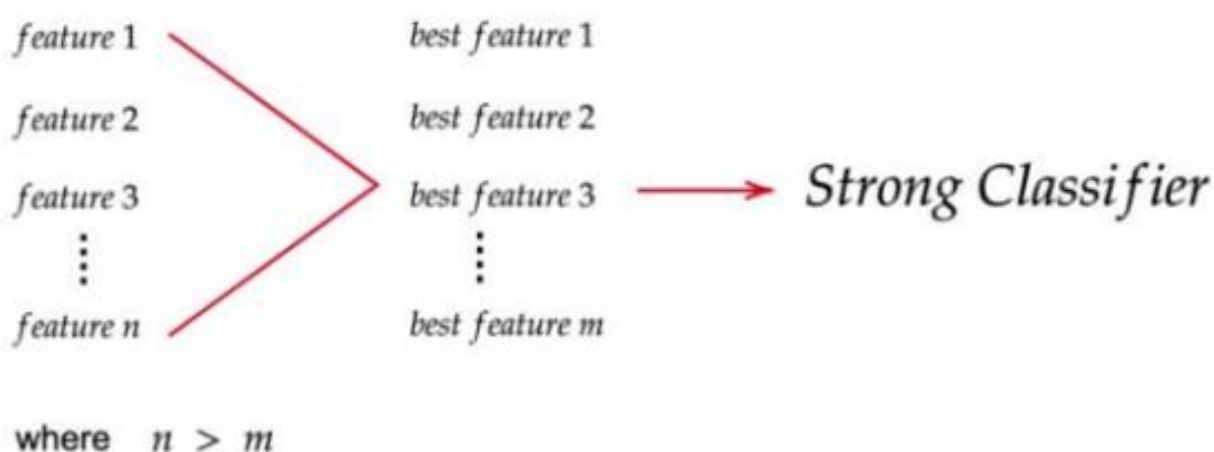


Рисунок 1.2 – Мета застосування алгоритму AdaBoost

Для кожного атрибуту слабкий класифікатор формулює порогову функцію класифікації, яка має бути оптимальною, Вона надає найменшу кількість прикладів, котрі класифіковані неправильно.

Каскадний класифікатор - це багатоступінчастий класифікатор, який може виявляти швидко і точно. Кожен етап складається із сильного класифікатора, створеного алгоритмом AdaBoost. Від етапу до етапу кількість слабких класифікаторів у сильному класифікаторі збільшується [11]. Вхідні дані оцінюються послідовно (поетапно). Якщо класифікатор певного етапу дає негативний результат, введення негайно відкидається. Якщо вихід позитивний, вхід перенаправляється на наступний етап, як показано на рис. 1.3.

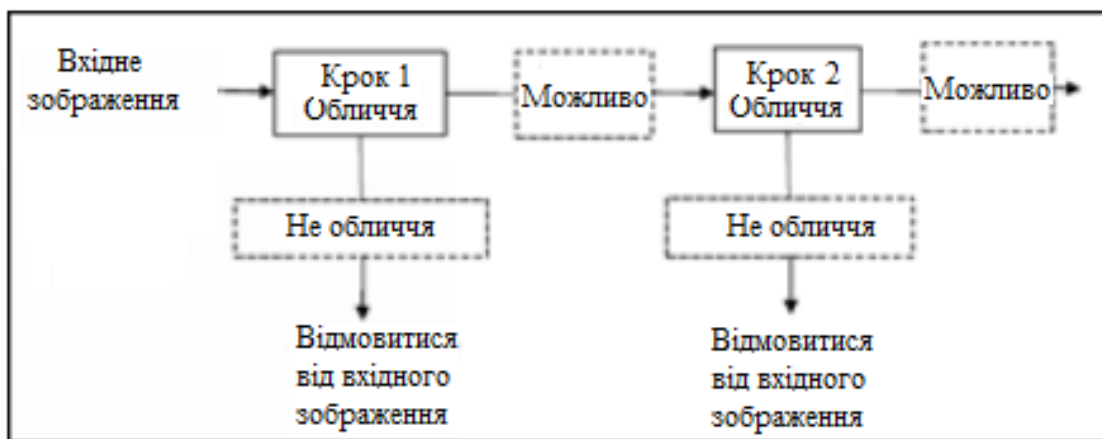


Рисунок 1.3 – Каскадний класифікатор

Переваги методу: добре розпізнає особи, повернуті на кут до 60 градусів. В інших випадках, якщо кут більше 60 градусів, його ефективність прямує до нуля. У зв'язку з цим ефективно розпізнавати обличчя людини, розташоване під довільним кутом, ставати неможливо і цей метод не підходить під вимоги сучасних систем спостереження. Також варто згадати як перевагу і швидкодію роботи цього методу

Але у застосуванні методу можна виділити проблеми:

- освітлення;
- стан голови у просторі;
- тривалий час навчання.

Для перевірки даних алгоритмів вчені США розробили програму FERET. Тестували як вищеописані алгоритми, а й безліч інших методів. Ефективність їх приблизно однакова, і становить 90% для знімків, зроблених одному місці і з однаковою апаратурою. При використанні різної апаратури та різного освітлення ефективність падає на 10%. Якщо з моменту останнього знімка проходить близько року, то відсоток розпізнавання зменшується до 50%.

### 1.2.2 Аналіз локальних ознак

Цей метод використовує топографічне індексування для опису особи. Насправді він є банком локально корельованих детекторів. Детектори



здійснюють розкладання МГК. Помилковість такого представлення дорівнює помилці розкладання МГК, проте таке представлення дозволяє різко зменшити критичність до варіацій освітлення обличчя, знизити чутливість до обертання і зменшити обчислювальну складність проти МГК.

Принципово інший підхід до розпізнавання є основою категорії кластерних методів. Їхній початковий принцип полягає в кореляційному порівнянні зображень. Ці методи не ставлять завдання виділити структурні елементи; ознаками тут є всі пікселі зображення. Однак, оскільки кореляційна схема надмірна, має велику обчислювальну складність і придатна для розпізнавання об'єктів, що не деформуються, з незмінним становищем і освітленням, то ці методи практично не використовують для розпізнавання осіб. Проте, звужуючи простір вихідних ознак, можна значно покращити швидкість алгоритму. Слід зазначити, що ці кластерні методи здійснюють спробу максимального поділу класів зображень осіб і є саме інструментом класифікації зображень. Ознаками екстрактора у всіх кластерних методах є області, що найбільш відрізняються від середньої особи області розпізнаваної особи. МГК є класичним кластерним методом розпізнавання. Сам метод і деякі принципи, що використовуються в ньому, часто застосовуються на стадіях попередньої цифрової обробки як кластерних, так і в інших способах розпізнавання. Тому метод буде розглянуто докладніше.

Теоретичне підґрунтя методу полягає у переведенні вихідного зображення в простір ознак набагато меншою розмірністю, в якому його можна максимально повно охарактеризувати.

Процес підготовки зображень обличчя за МГК наведено на рис. 1.4.

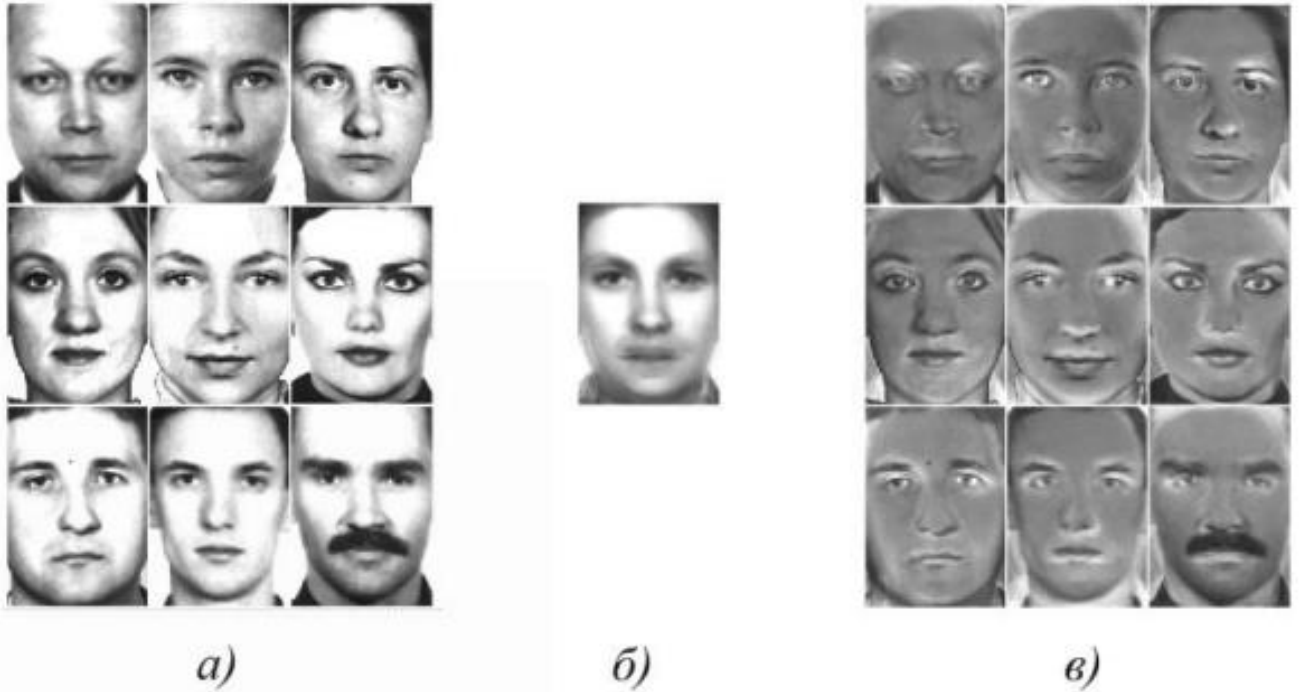


Рисунок 1.4 - Підготовка зображень осіб для класифікації МГК

Для проведення класифікації в МГК будь-яке зображення (рис. 1.4, а) представляється як вектор в  $n$ -вимірному просторі, де  $n$  — кількість пікселів зображення.

Використання методу з множиною векторів лінійного простору дає змогу перейти до власне такого базису, що основна дисперсія буде спрямована вздовж перших кількох його головних осей. Це дозволяє представити унікальність векторів тренувального набору за допомогою декількох компонентів. Отриманий простір максимально інформативно описує об'єкт. Насправді для здійснення методу використовується перетворення Карунена-Лоева. На першому кроці знаходиться середня особа із зображень тренувального набору, як показано на рис. 1.4, б.

На другому кроці обчислюються зображення центрованих осіб середньої особи за допомогою віднімання середнього зображення з вихідних. Саме ця різниця (рис. 1.4, в) трансформується у форму вектора багатовимірного простору осіб.

Отримані вектори утворюють згустки (кластери), оскільки їх компоненти корелюються між собою. Простір осіб може бути описаний власними векторами, і, виходячи з цього, потрібно менше вимірювань для його представлення. Такі

вектори є власними особами, які характеризують відмінності щодо середньої особи.

### 1.2.3 Нейромережні методи

ШІ - здатність машини або програми знаходити рішення за допомогою обчислень. Головною задачею ШІ є розроблення технологій та алгоритмів розв'язання пізнавальних завдань, котрі властивих мозку людини.

МН - це спроба фактично навчити комп'ютери самостійно учитися на значному обсязі даних замість жорстких норм. МН дає змогу комп'ютерам навчатися самостійно. Це можливо завдячуючи обчислювальній потужності модерних машин, які легко опрацьовують достатньо значні набори даних.

Також МН можна визначити, як набір алгоритмів та інструментів, які допомагають машинам розуміти закономірності в даних та використовувати цю базову структуру для виконання дії та прийняття рішення про поставлене завдання. Є багато способів, якими машини прагнуть зрозуміти закономірності, котрі містяться у основі даних.

Приклад найпростішої моделі МН показано на рис. 1.5.

Термін МН був придуманий в 1959 Артуром Самуелем, співробітником ІВМ і піонером в області комп'ютерних ігор і ШІ [11, 12]. Репрезентативною книгою досліджень МН у 1960-х роках була книга Нільссона про машини для навчання, присвячена в основному МН для класифікації шаблонів.

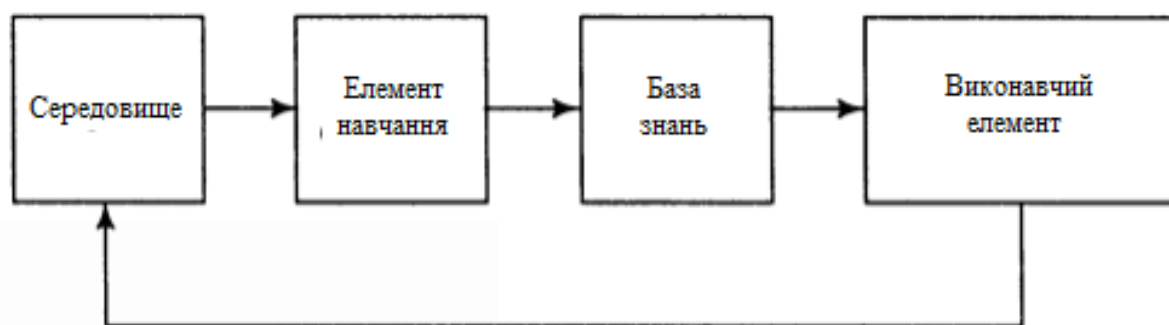


Рисунок 1.5 - Приклад найпростішої моделі МН

Зацікавленість, пов'язана із розпізнаванням образів, зберігався й у 1970-х роках, як описано Дудою та Хартом у 1973 році [13]. В 1981 був представлений звіт про використання навчальних стратегій, так що ШНМ вчиться розпізнавати 40 символів з комп'ютерного терміналу.

Нейромережа чи ШНН — це спроба відтворення роботи мозку на комп'ютері з допомогою шарів штучних нейронів.

Під час перших досліджень в галузі ШІ вчені намагалися відтворити людський інтелект для вирішення конкретних завдань, для прикладу - гри з людиною. Було введено велику кількість норм, яких має дотримуватися комп'ютер. На основі цих правил машина ухвалювала рішення у згоді з конкретним списком можливих дій.

## РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА

### 2.1 Контрольоване та неконтрольоване навчання

Перший вид (або навчання з учителем) потребує застосування зазначених датасетів, що мають вхідні дані та очікувані результати на виході. При такому виді навчання навчання ШНМ повинна отримати як і вхідні дані, так і очікувані підсумки. Якщо результат, отриманий мережею, не відповідає тому, який був поданий як очікуваний, мережа має скоригувати свої обчислення. Так триває до тих пір, поки результат, який згенерувала ШНМ, не відрізнятиметься від очікуваного в межах заданої допустимої помилки.

Варіантом проблеми із контрольованим навчанням служить прогноз погодних умов. Нейромережа вчиться робити предбачення погоди, застосовуючи інформацію про погоду минулих днів. Навчальні дані містять вхідну інформацію (як то тиск, вологість, швидкість вітру) та вихідні підсумки (температура).

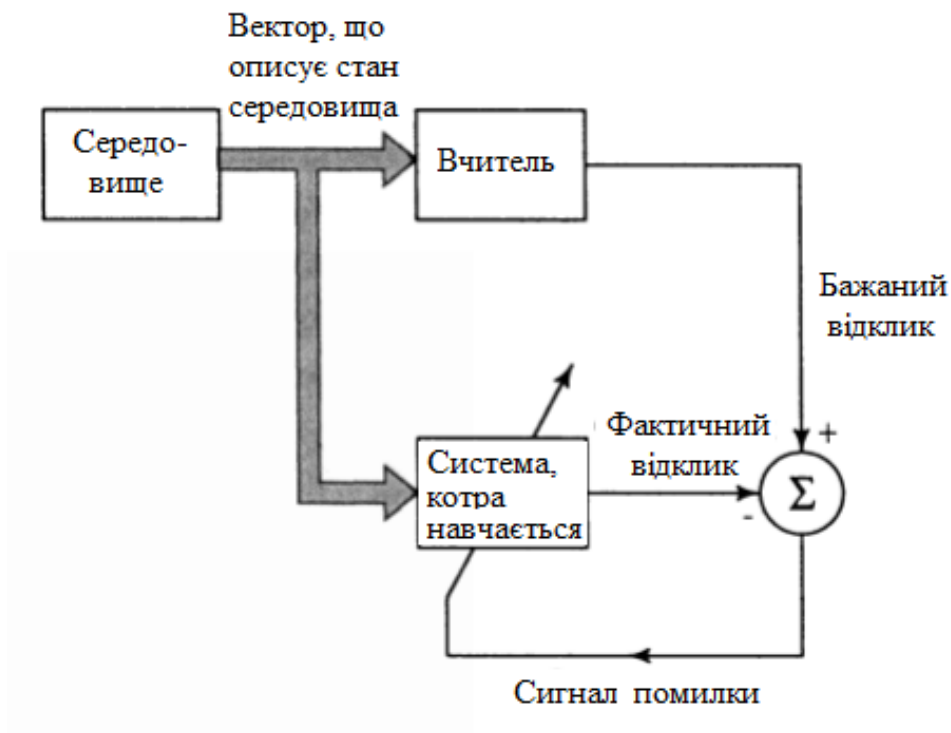


Рисунок 2.1 – Блокова діаграма контрольованого навчання [14]

Неконтрольоване навчання (або навчання без вчителя) - це МН із

застосуванням наборів неструктурованих даних. При такому навчанні ШНМ має самостійно логічно класифікувати їх. Яскравим прикладом такого виду є завдання виявлення аномалій. В цьому випадку нейромережа не навчається на розмічених даних. Однак вона класифікує інформацію на вході, а при виявленні даних, що явно відрізняються від інших, дає про це сигнал.

## **2.2 Глибинне навчання**

Є відгалуженням МН, яке повністю засноване на ШНМ, оскільки нейромережа імітуватиме людський мозок, тому глибинне навчання також є своєрідною імітацією людського мозку [15]. У такому навчанні нам не потрібно все явно програмувати. Концепція глибинного навчання не є новою. Вона існує вже кілька років. Зараз це галас, тому що раніше у нас не було такої обчислювальної потужності та великої кількості даних. Упродовж останніх 20 років обчислювальна потужність зростає в геометричній прогресії, тому на перший план вийшли глибинне навчання та МН. Формальне визначення глибинного навчання – нейрони.

Незважаючи на те, що глибинне навчання набуло широкого поширення лише відносно нещодавно, спроби навчати алгоритми, засновані на багатосарових ШНМ, здійснювалися з 60-х років 20 століття. У 1980 Куніхіко Фукушіма представив модель Неокогнітрон, призначену для розпізнавання візуальних образів.

ШІ, особливо глибинне навчання, є значним і, цілком можливо, революційним стрибком вперед у поліпшенні технологій, що підтримують охорону здоров'я. Такі комп'ютерні алгоритми є інструментами, які можна навчити для поліпшення якості догляду за пацієнтами за рахунок підвищення точності діагностики та зниження навантаження на медичних працівників. Вони пропонують можливості для автоматизації та прогнозування, раніше небачені у охороні здоров'я. Для пошуку кореляцій та закономірностей у клінічних даних можна використовувати дуже великі набори даних, що, ймовірно, неможливо при використанні людського мозку без сторонньої допомоги чи традиційних

підходів. Нарешті, системи глибокого навчання можуть допомогти медичним організаціям ефективніше взаємодіяти з персоналом та пацієнтами.

### 2.3 Нейронні мережі глибокого навчання

Як і у тварин, ШНМ містить взаємопов'язані нейрони. На рис. 2.2 вони представлені кругами:

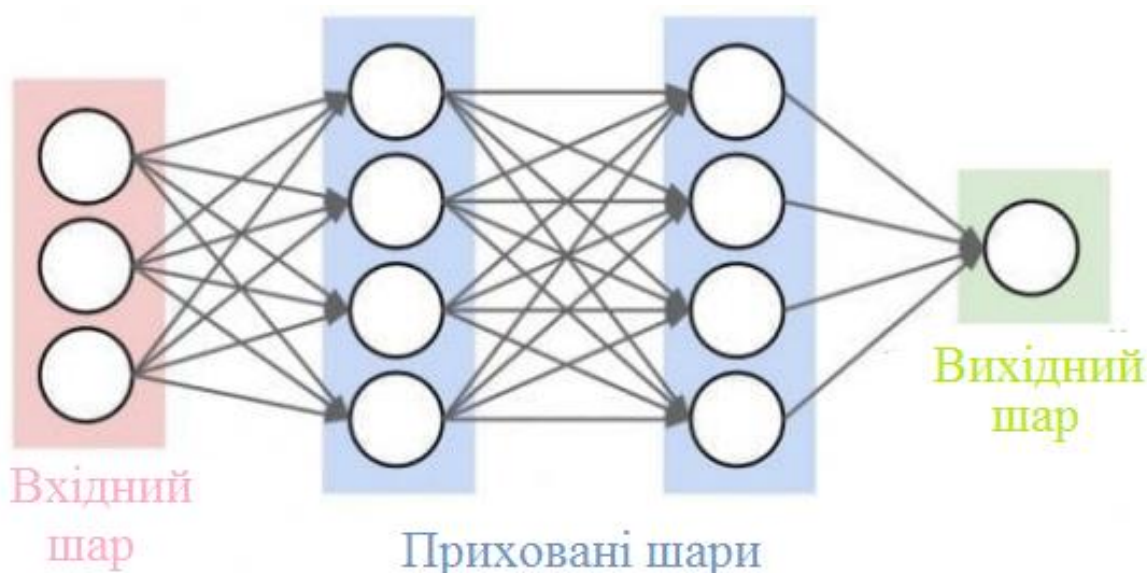


Рисунок 2.2 – Глибока нейромережа (з двома прихованими шарами)

Нейрони згруповані в три шари різних типів [15]:

- вхідний;
- прихований ( $i$ );
- вихідний.

Перший отримує дані. Другі проводять математичні розрахунки із ними. Однією із задач при побудові ШНМ є виявлення числа прихованих шарів і нейронів власне у кожному з них. "Глибина" у понятті "глибоке навчання" властиво і значить, що прихованих шарів є більше одного.

Нейрони перебувають у зв'язку між собою через деяку вагу. Випадковим чином вказуються вихідні ваги. Кожен нейрон містить ФА. Її сенс важко осягнути без залучення математичних міркувань..

Після того, як вхідний датасет перейшов через усі шари ШНМ, ФА повертає вихідні результати.

З метою зменшення величини функції втрат, необхідно варіювати ваги між нейронами. Звичайно, це можна випадково здійснювати, поки ця функція не перетвориться в нуль, але це дуже неефективно. Тому ми будемо застосовувати МГС, котрий дає змогу відшукати мінімум функції.

## 2.4 Перенавчання нейромереж та методи запобігання перенавчанню

Глибинні нейромережі націлені на вивчення та узагальнення закономірностей, виявлених у навчальних даних, щоб вони могли однаково працювати з тестовими чи новими даними. Хоча це ідеальний стан або сценарій, який можна очікувати від глибинних ШНМ, це легше сказати, ніж зробити, особливо для таких мереж, як CNN, рекурентні нейромережі і т. д., які складаються з мільйонів або мільярдів параметрів, що налаштовуються і які вразливі для перенавчання.

Перенавчання відбувається, коли модель навчається на прихованих випадкових закономірностях у навчальних даних, але відсутніх у генеральній сукупності. Перенавчання характеризує узагальнюючу здатність моделі.

Для вирішення проблеми перенавчання глибинних нейромереж застосовують такі техніки регуляризації:

- L1- та L2-регуляризація ваг;
- рання зупинка навчання;
- метод "dropout".

Сенс першої техніки полягає в обмеженні абсолютних значень ваг ШНМ. Для цього до функції втрат застосовується один із регуляризаторів:

L2-регуляризатор – сума квадратів ваг  $L1 = \lambda \sum_{\omega} \omega^2$

L2-регуляризатор – сума модулів ваг  $L2 = \lambda \sum_{\omega} |\omega|$

Друга техніка досить проста. Необхідно відкласти частину навчальної вибірки у валідаційну вибірку. Потім на кожній епосі навчання обчислювати помилку на цій вибірці - якщо помилка починає зростати після певного



мінімуму, отже, нейромережа почала переучитися і потрібно зупинити процес навчання. Є дані, що дана техніка певною мірою близька до L2-регуляризації [11]. Також аналіз помилки на валідаційній вибірці дозволяє оцінювати узагальнюючу здатність моделі.

Третій метод також знайшов широке застосування попередження перенавчання ШНМ, оскільки є більш ефективним, ніж перший, і часто застосовується перш за все. Ідея методу “dropout” полягає в наступному – для кожного нейрона, крім нейронів вихідного шару, визначається ймовірність  $p$ . З цією ймовірністю нейрон вимикається. Алгоритм навчання стає таким: на кожному семплі  $x$  з навчальної вибірки обчислюється, чи буде активний нейрон, і в залежності від результату нейрон залишається активним, або відключається і його вихід встановлюється рівним нулю. Це веде до того, що нейрон відсутній у обчислювальному графі. Прямий прохід і зворотне поширення градієнта не йдуть далі неактивного нейрона [16]. Експерименти [16], і практика навчання ШНМ показують, що метод “dropout” дійсно дає дуже серйозні покращення як навчена модель у різних додатках.

## **2.5 Згорткові нейронні мережі**

### **2.5.1 Загальні поняття**

CNN – форма архітектури ШНМ, котру у 1988 році запровадив Ян Лекун. Вона призначена для виконання ефективного розпізнавання образів, є складовою частиною парадигм глибокого навчання [17]. Ідея CNN міститься у послідовності згорткових шарів та шарів підвибірки (субдискретизуючих). Мережа є односпрямованою (зворотні зв'язки відсутні) та обов'язково багат шаровою. Для її навчання застосовуються розповсюджені методи, найбільш часто метод зворотного розповсюдження помилки. Варто згадати, що ФА нейронів може бути властиво будь-якою.

CNN особливо корисні для пошуку шаблонів (патернів) на зображеннях для розпізнавання об'єктів, облич та сцен. Вони навчаються безпосередньо на

зображеннях, використовуючи шаблони для класифікації зображень та усуваючи потребу в ручному видаленні ознак [18].

Додатки, що вимагають розпізнавання об'єктів та CV, такі як безпілотні автомобілі та додатки для PO, багато в чому залежать від CNN. Залежно від вашої програми ви можете побудувати свою CNN з нуля або використовувати готову модель на ваших даних.

Головна ідея CNN полягає в тому, обробка частини зображення повинна відбуватися незалежно від розташування цієї частини. Реалізується ця ідея явно – ми покриваємо вхідне зображення малими вікнами (наприклад, 5x5 пікселів) та виділяємо атрибути у кожному 20 такому вікні малою ШНМ. При цьому ознаки виділяються в кожному вікні однакові, отже, буде одна мережа. Вона матиме  $5 \times 5 = 25$  входів. З кожного зображення вийде багато різних входів цієї ШНМ. Після цього вихід можна буде знову подати у вигляді зображення, замінивши вікна 5x5 на їх центральні пікселі. На цьому зображенні можна буде застосувати другий згортковий шар вже з іншою нейромережею. У кожному згортковому шарі буде трохи вільних параметрів порівняно з повнозв'язковими нейронними мережами.

Присутність операції згортки і дала назву архітектурі ШНМ. Її суть – будь-який фрагмент зображення перемножується поелементно на матрицю (ядро) власне згортки, а вже сам результат сумується та поміщається в аналогічне місце зображення на виході. Цей алгоритм зображено на рис. 2.3 та 2.4.

Зазвичай вихідне зображення містить три матриці (по кольірних каналах RGB) - червону, зелену і блакитну. Такі матриці називають каналами. Такі самі матриці будуть виходити після кожного згорткового шару. У них є просторова структура, котра відповідає малюнку на виході. Каналів стає більше. Таким чином, значення кожного атрибуту, котрі були попередньо виділені з вікон у первісному зображенні, стануть матрицею. Такі матриці прийнято називати картами ознак.

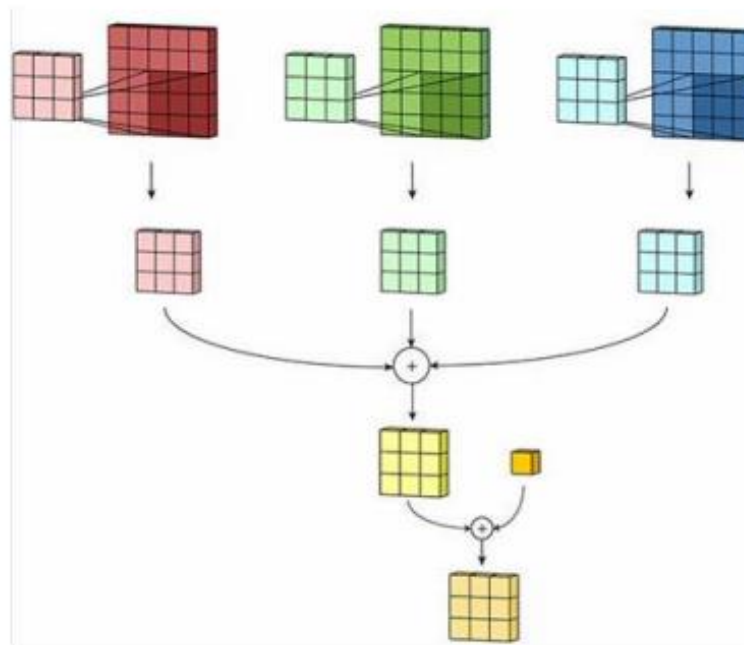


Рисунок 2.3 – Операція згортки

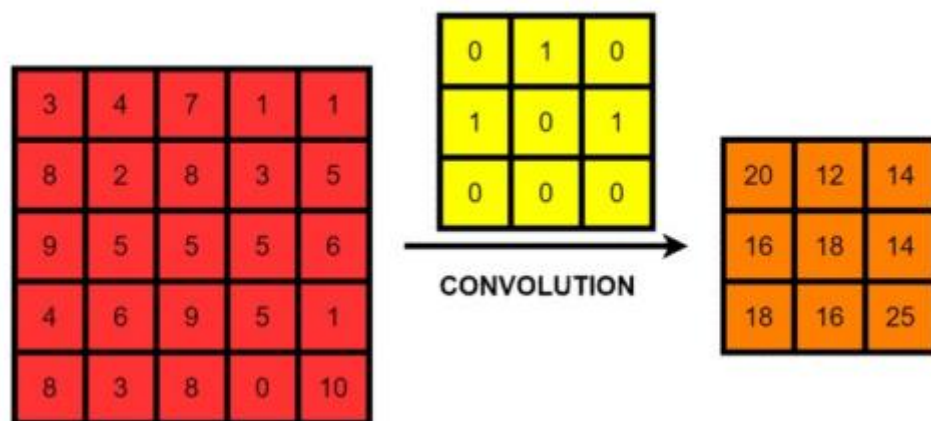


Рисунок 2.4 – Робота алгоритму згортки для одноканального зображення

Подібна інтерпретація має скоріше метафоричний чи ілюстративний характер. Насправді карти ознак, що виробляються складною мережею, дуже складні для інтерпретації. На практиці суть цих ознак навіть не намагаються зрозуміти, і тим більше змінювати, а натомість для покращення результатів розпізнавання змінюють структуру та архітектуру мережі.

У звичайному перцептроні, який є повнозв'язною ШНМ, кожен нейрон перебуває у зв'язках із всіма нейронами минулого шару, причім кожен зв'язок володіє особистим ваговим параметром. У CNN в операції згортки

застосовується тільки обмежена матриця ваг з малим розміром, що переміщається спочатку за зображенням, а потім по всьому шару, що обробляється.

Природно, в CNN набір ваг далеко не один, їх є велике різноманіття, що дозволяє кодувати фрагменти зображення. Проходження кожним рядом ваг формує особистий екземпляр карти ознак, роблячи ШНМ багатоканальною. При перебиранні шару матрицею ваги її пересувають, як правило, аж ніяк не на повний її розмір, а лише на малу віддаль.

Операція субдискретизації зменшує розмірність побудованих карток ознак. Завдяки їй мережа стає інваріантною до масштабу вхідного зображення.

### 2.5.2 Операція згортки

Нехай  $x^l$  - карта ознак у шарі  $l$ . Тоді результат проведення двовимірної згортки з ядром розміру  $2d + 1$  і матрицею ваг  $W$  розміру  $(2d + 1) \times (2d + 1)$  на наступному шарі дорівнюватиме

$$y_{i,j}^l = \sum_{-d \leq a, b \leq d} W_{a,b} x_{i+a, j+b}^l \quad (2.1)$$

де  $y_{i,j}^l$  – результат на рівні  $l$ , а  $x_{i,j}^l$  - вхід згортки, він же вихід минулого шару. Для отримання результату на наступному шарі ми використовуємо згортку до квадратного вікна з минулого шару.

Операція згортки володіє такими властивостями:

- має розрідженість;
- згортка багатократно застосовує ті ж самі ваги;
- згортка зберігає структуру вхідного зображення.

Після операції згортки можна, застосувати ту чи іншу нелінійну функцію  $h$ : вона буде просто застосовуватися до кожного елемента отриманого тензора окремо.

### 2.5.3 Функція активації

Іншим важливою складовою є ФА, про яку було сказано раніше. ФА після шару згортки є джерелом нелінійності, що дає можливість навчати ШНМ. ФА повинна відповідати таким критеріям (бути):

- монотонно неспадаючою;
- обмеженою, тобто. прагнути до нуля при  $x \rightarrow \infty$ ;
- диференційованою.

Активаційний шар, як правило, логічно поєднують з шаром згортки (приймають, що ФА вмонтована в нього). Існує безліч ФА: логістична сигмоїда, гіперболічний тангенс та інші. Найчастіше використовується останній. Однак, це не обов'язково, функцію можна вибрати іншу. У 2000-х роках було запропоновано та досліджено нову функцію активації ReLU [19], яка дала змогу значно пришвидшити процес навчання, спростивши при цьому розрахунки (через простоту власне функції). Суть функції полягає у відсіканні негативної частини скаляру.

Формула ReLU має такий вигляд:

$$ReLU(x) = \begin{cases} 0, & \text{якщо } x < 0, \\ x, & \text{якщо } x \geq 0. \end{cases} \quad (2.3)$$

Станом на 2022 рік ця функція та її модифікації є ФА, що найбільш часто використовуються, в глибоких нейромережах, зокрема, в згорткових. Існує методика визначення оптимальної кількості блоків лінійної ректифікації.

Функція ReLU ефективніша, ніж логістична регресія чи гіперболічний тангенс, оскільки обчислення її похідної вимагає складних розрахунків: якщо  $x < 0$ , тоді похідна дорівнює нулю, якщо  $x \geq 0$ , тоді похідна дорівнює одиниці. Це також дозволяє зменшити витрати на обчислювальні ресурси. Крім класичної реалізації ReLU існують її модифікації, що дозволяють отримати найкращий результат у деяких завданнях [19].

### 2.5.4 Пулінг

У класичному згортковому шарі, крім лінійної згортки та наступної за нею ФА, є ще одна операція – пулінг.

У CNN важливіша наявність чи відсутність будь-якої ознаки, ніж його точні координати. Наприклад, при розпізнаванні осіб важливо зрозуміти, чи є на зображенні обличчя і кому воно належить, а не дізнатися, де воно розташоване на зображенні. Тому ми можемо дозволити собі узагальнити ознаки, при цьому втративши частину інформації про їх розташування. До того ж це дає можливість скоротити розмірність [19].

В основному як операція пулінга застосовується операція взяття максимуму (max-pooling). Формула для розрахунку пулінгу:

$$x_{i,j}^{l+1} = \max_{-d \leq a \leq d, -d \leq b \leq d} z_{i+a, j+b}^l \quad (2.2)$$

Приклад роботи операції макс-пулінг представлений на рис. 2.5.

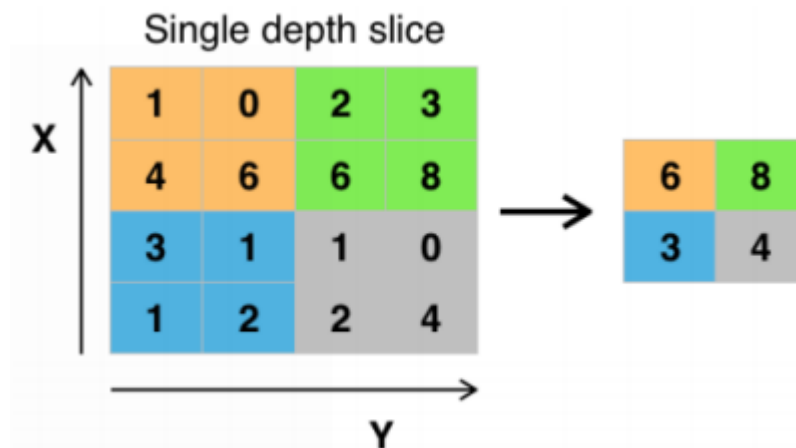


Рисунок 2.5 – Макс-пулінг

### 2.5.5 Типова структура

На рис. 2.6 показана типова структура CNN для опрацювання зображень. Розглянемо її докладніше. Мережа містить значне число шарів. Після зображення на вході сигнал переходить через кілька згорткових шарів, де

відбувається чергування згортки та підвибірки.

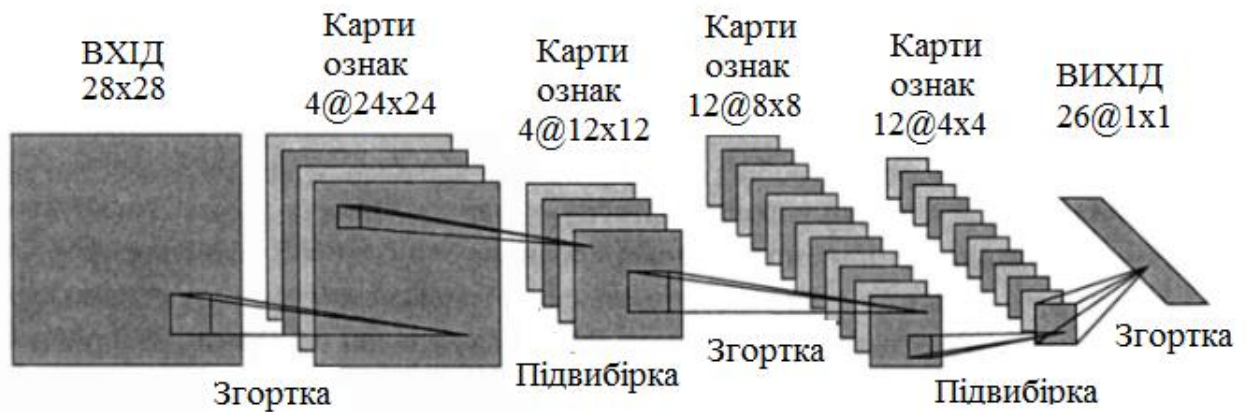


Рисунок 2.6 – Приклад роботи CNN для обробки зображень

Це дає змогу становити «карти ознак», у подальших шарах карта стає меншою на вигляд, але більшою через число каналів.

Згортковий шар. Є основним блоком CNN, котрий містить для кожного каналу окремий фільтр, ядро котрого опрацьовує минулий шар елементів, сумуючи добуток поелементно для фрагменту. Ваги ядра визначаються по ходу навчання.

Особливістю згорткового шару є порівняно невелика кількість параметрів, що встановлюється під час навчання. Коли вихідна картинка містить  $100 \times 100$  пікселів по трьох каналах (це означає 30000 вхідних нейронів), а власне шар згортки застосовує фільтри з ядром  $3 \times 3$  пікселя з виходом на 6 каналів, відповідно в ході навчання береться тільки 9 ваг ядра. Якщо взяти кількість каналів вхідного зображення рівним 3, тоді загальна кількість параметрів  $9 \times 3 \times 6 = 162$ , що значно менше, ніж число шуканих атрибутів повнозв'язкової нейромережі.

Пулінг чи субдискретизація. Цей шар є нелінійним ущільненням власне карти ознак. Набір пікселів (як правило,  $2 \times 2$  розміру) перебудовується до одного пікселя, шляхом нелінійного перетворення. Для цього найчастіше використовується функція максимуму. Не заборонено застосовувати й інші функції, як варіант, середньої величини чи L2-нормування, проте практичні

дослідження довели переваги саме функції максимуму. Властиво перетворення торкають прямокутники або квадрати, що не перетинаються, кожен з яких стискається в один піксель (береться з найбільшим значення).

На відміну від самих карт ознак шар підвиборки можна якось інтерпретувати. Уявімо, що після якогось кроку згортки було отримано набір карт ознак (вона може бути одна насправді). Оскільки мережа вже розпізнала якусь ознаку і відобразила її на карті, для подальшої роботи вся карта ознаки не потрібна, її можна узагальнити, погіршивши якість зображення. З іншого боку, така фільтрація вже непотрібних ознак дозволяє мережі не перевчитися. Найчастіше шари пулінгу вставляються між двома шарами згортки. Це видно на рис. 2.6. З метою різкішого зменшення обсягу одержуваних представлень використовують менше фільтрів, котрий іноді зовсім відмовляються від шарів підвиборки.

Після кількох проходження згортки картинки та ущільнення з використанням пулінга система трансформується від конкретної мережі пікселів з великою роздільною щільністю до абстрактніших карт. Зазвичай на кожному подальшому шарі зростає число каналів і падає щільність картинки в каналі. На фініші буде значний сет каналів, котрі містять незначне число даних.

Найпростішим і найпопулярнішим способом навчання з вчителем на розмічених даних, але існує набір інших методів навчання CNN без вчителя. Як приклад, фільтри згортки властиво дозволено вчити самостійно, через подачу на них отриманих випадково частинок вихідних картинок навчального датасету та прикладаючи до них будь-який існуючий алгоритм. Таким чином, черговий згортковий шар мережі навчатиметься на частинках від вже навченого власне попереднього шару.

Для покращення роботи CNN, збільшення її стійкості та унеможливлення перенавченню застосовується також виняток - це таке тренування підмережі, при якому виключаються випадкові одиночні нейрони.

## **2.5.6 Переваги та недоліки**

Переваги мереж згортки [20]:



- останнім часом CNN стали кращими у вирішенні завдань розпізнавання зображень;
- у порівнянні з перцептроном значно менше число ваг, що налаштовуються, так як одне ядро ваг береться для всього зображення, затомість формувати для кожного пікселя вхідної картинки свої вагові параметри. Тому неймережа під час навчання узагальнює інформацію, а чи не запам'ятовує кожен піксель кожної продемонстрованої картинки, як це робить перцептрон;
- якісне розпаралелювання розрахунків. Це дає змогу реалізувати алгоритми на GPU;
- навчання ядер фільтрів згортки із застосуванням методу зворотного розповсюдження помилки.

До недоліку можна віднести число атрибутів, що налаштовуються. Їх чимало і в деяких діапазон значень обмежений лише бажанням дослідника. Усі вони чинять значний вплив на результат, проте беруться фахівцями емпірично [20].

## **2.6 Процес РО**

РО - це програмний алгоритм, що використовується для перевірки або ідентифікації особи людини через обробку відеокадра або цифрового зображення, на якому видно її обличчя. Технології розпізнавання осіб використовують кілька різних методів, але вони порівнюють риси особи на зображенні з особами, які у містяться у БД.

Процес РО - це множина різноманітних завдань, які служать для ідентифікації особи за цифровим зображенням або відео. Назагал цей процес відбувається наступним чином (див. рис. 2.7): після того, як в систему надійшло з камери зображення, першим кроком у системі є детектування осіб, в якому визначається, чи містить зображення чи відео будь-які особи. Виявлення обличчя також визначає розташування ключових функцій (званих «орієнтирами»), таких як очі, ніс і т. д. Ці орієнтири також можуть використовуватися для визначення розташування обличчя на зображенні або

відео та визначення його розміру та орієнтації.



Рисунок 2.7 – Схема РО

Далі на етапі виявлення із застосуванням алгоритмів окреслюються межі особи. Відповідно проходить етап розпізнавання, коли власне обличчя перетворюється (переінакшується його яскравість, воно піддається вирівнюванню та масштабуванню) і зводиться до певного виду. На заключному етапі проходить визначення ознак та власне порівняння їх із ознаками з БД. Цей етап носить назву ідентифікації чи верифікації.

Верифікація – це процес зіставлення зразків за схемою один до одного (1:1). Ідентифікація – процес зіставлення за схемою «один до багатьох» (1: N).

## 2.7 Методи попередньої обробки зображень

Після того, як система фіксує обличчя в кадрі і встановлює позицію голови, її розмір та основні риси, обличчя, здебільшого, піддають нормалізації: його кодують, масштабують, трансформуються до горизонталі центральна лінія очей. Також, при передобробці обличчя, використовуються різні фільтри для зменшення рівня шуму.

Найчастіше зустрічаються наступні методи:

- нелінійні фільтри, котрі застосовують для викорінення імпульсного шуму (деяких точок з максимальною (білою) або мінімальною (чорною) яскравістю) на зображенні;
- Гауссівський фільтр для розмиття картинки, котрий застосовують

при наявності на зображенні дрібних деталей, котрі не потребують відокремлення від фону та їх можна розмити;

- медіанні фільтри, які застосовують для зберігання порогів яскравості контурів і придушення імпульсних шумів. Вони непогано працюють у таких випадках, коли щільність шуму є невеликою.

## **2.8 Методи РО**

Інша важлива частина систем автоматичного РО — це власне самі алгоритми розпізнавання. Зараз їх є багато, і кожен із них має свою особливість, швидкість роботи та надійність РО.

Алгоритми розпізнавання поділяються на дві категорії, залежно від технології розпізнавання, котра використовується:

- РО за геометрією обличчя (2D-технології);
- РО за будовою черепа (3D-технології).

Перші опрацьовують «плоскі», двовимірні картинки та РО проходить через аналіз його текстуру і частини обличчя з, яка має високу контрастність. Саме тому при недотриманні якісного освітлення чи позиції обличчя РО проходить дуже важко.

Алгоритми другої групи більш стійкі до цих змін, так як при створенні моделі обличчя, в них враховуються особливості будови черепа. Але хоч системи 3D і мають такий великий плюс, через відсутність варіантів задоволення потреб значного числа користувачів при ідентифікації та через низьку швидкість, 3D-технології наразі не застосовуються широко. Також такі варіанти РО потребують значних ресурсів для здійснення обчислень, і вартість такого устаткування значно більша [21].

## РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА

### 3.1 Використання програмних засобів

Використано можливості OpenCV, що виконує роль базової компоненти, котра надає інструментарій для роботи з вхідними даними, представленими послідовністю відео. Для роботи була обрана мова Python та бібліотеки Tensorflow (для створення та навчання мережі), NumPy (для зручної роботи з масивами даних зображень), Matplotlib (для зображення результатів навчання) та ін.

Бібліотека OpenCV існує для багатьох мов програмування [22]. Але найчастіше вона використовується для написання програм МН мовою Python, особливо у сфері CV. Реалізована на C/C++, проте є розробки для Python, Java, Ruby, Matlab, Lua та інших мов. Дана бібліотека дозволяє представити відеоряд як набір кадрів (фреймів), кожен з яких передбачається розглядати як окреме зображення.

Також OpenCV підтримує роботу з потоковим відео, отриманим безпосередньо з камери, що надає потенціал обробки та аналізу вхідних даних у режимі реального часу, що є вкрай важливим аспектом у проектуванні системи CV. Також дана бібліотека містить зручний функціонал по роботі із зображеннями, алгоритми їх обробки, розпізнавання осіб та ін. Саме тому було прийнято рішення про використання OpenCV.

TensorFlow - бібліотека для МН від Google для розв'язання проблем формування та тренування нейромережі з метою автоматичного відшукування та класифікації образів [23]. Основний API був реалізований для Python. Цей фреймворк дозволяє абстрагуватися від внутрішніх деталей МН та зосередитися безпосередньо на розв'язанні задачі, дозволяє створювати, навчати та використовувати ШНМ будь-якого відомого типу [18].

NumPy або Numerical Python — це відкрита бібліотека під Python, котра пропонує наступне [24]:

- потужний N -вимірний масив;

- високорівневі функції;
- інструменти для інтеграції коду C/C++ та Fortran;
- використання лінійної алгебри, перетворень Фур'є та можливостей випадкових чисел.
- ефективний багатовимірний контейнер загальних даних.

NumPy надає таку функціональність на рівні з MatLab.

Matplotlib є низькорівневою бібліотекою для побудови двовимірних діаграм та графіків [25]. З її допомогою можна збудувати будь-який графік. Бібліотека володіє підтримкою 2D та 3D графіки. Однак для більш складної візуалізації даних потрібно більше коду, ніж у найрозвиненіших бібліотеках.

### 3.2 Попереднє навчання моделі

Для того, щоб більш ефективно застосовувати обчислювальні ресурси, можна використовувати моделі, навчені заздалегідь на будь-яких інших даних і доступні в Інтернеті або які поставляються з бібліотекою OpenCV.

Для використання такої моделі завантажується навчена нейромережа. Однак для її додаткового навчання необхідно «заморозити» шари цієї мережі. Потім додаються нові шари в структуру і відбувається навчання мережі, котра сформувалася.

На рис. 3.1 представлено цей алгоритм.

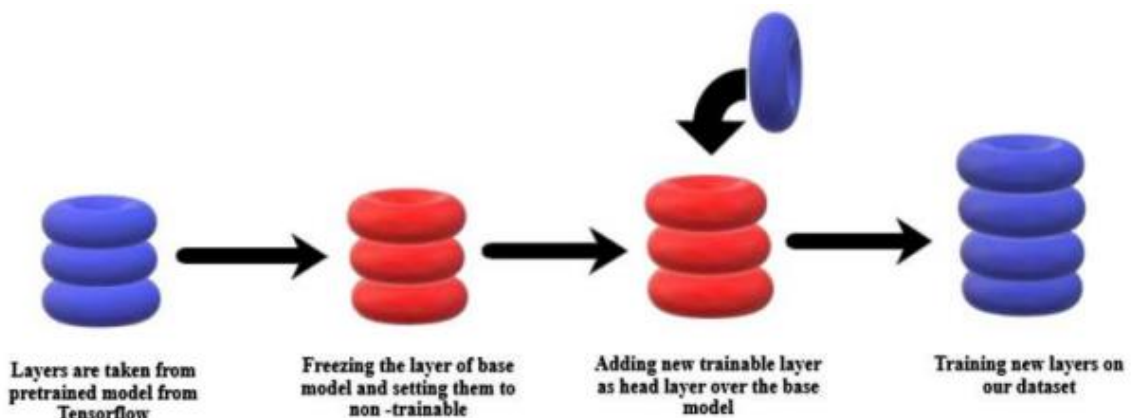


Рисунок 3.1 – Алгоритм використання попередньої моделі

### 3.3 Мережа MobileNetV2

MobileNetV2 - це вже готова та навчена CNN. Згідно з даними [26], глибина цієї мережі становить 53 шари згортки. Усього ж мережа містить на сто шарів більше, а саме 154. Таким чином можна скористатися заздалегідь навченою версією мережі. Вона була навчена більш, ніж на мільйон зображень від бази даних ImageNet. Попередньо навчена мережа може класифікувати зображення до 1000 категорій різних об'єктів. В результаті мережа вивчила багаті уявлення функції для широкого спектра зображень. Вхідний розмір зображень MobileNetV2 224\*224. На виході мережа віддає 1280 карток ознак розміром 7\*7.

Мережа складається з блоків, який автори називають розширюючим згортковим блоком. Кожен такий блок складається із трьох шарів.

1. Спочатку йде "точкова згортка" з великою кількістю каналів. Автори називають це шаром розширення. На вході цей шар набуває тензору розмірності

$$D_f \cdot D_f \cdot C_{in} \quad (3.1)$$

На виході цей шар видає наступний тензор:

$$D_f \cdot D_f \cdot (t \cdot C_{in}) \quad (3.2)$$

де  $t$  – новий гіперпараметр, названий рівнем розширення. Автори рекомендують задавати цьому параметру значення від 5 до 10, на практиці цей параметр часто дорівнює 6. Цей шар створює відображення вхідного тензора у просторі великої розмірності.

2. Потім йде глибинна згортка із ReLU6-активацією. Цей шар разом з попереднім, по суті, утворює будівельний блок MobileNetV1 - мережі-батька MobileNetV2. На вході цей шар приймає тензор розмірності:

$$D_f \cdot D_f \cdot (t \cdot C_{in}) \quad (3.3)$$

На виході цей шар видає наступний тензор:

$$(D_{f/s}) \cdot (D_{f/s}) \cdot (t \cdot C_{in}) \quad (3.4)$$

де  $s$  – крок згортки.

3. Наприкінці йде  $1 \times 1$ -згортка з лінійною ФА, що зменшує кількість каналів. «Цільове різноманіття» високої розмірності, отримане після попередніх кроків, можна вмістити в підпростір меншої розмірності без втрати корисної інформації, що, власне, і робиться на цьому кроці. На вході такий шар приймає тензор розмірності

$$(D_{f/s}) \cdot (D_{f/s}) \cdot (t \cdot C_{in}) \quad (3.5)$$

А на виході віддає тензор розмірності

$$(D_{f/s}) \cdot (D_{f/s}) \cdot (t \cdot C_{out}) \quad (3.6)$$

де  $C_{out}$  – кількість каналів на виході блоку.

Структуру шарів MobileNetV2 можна побачити на рис. 3.2.

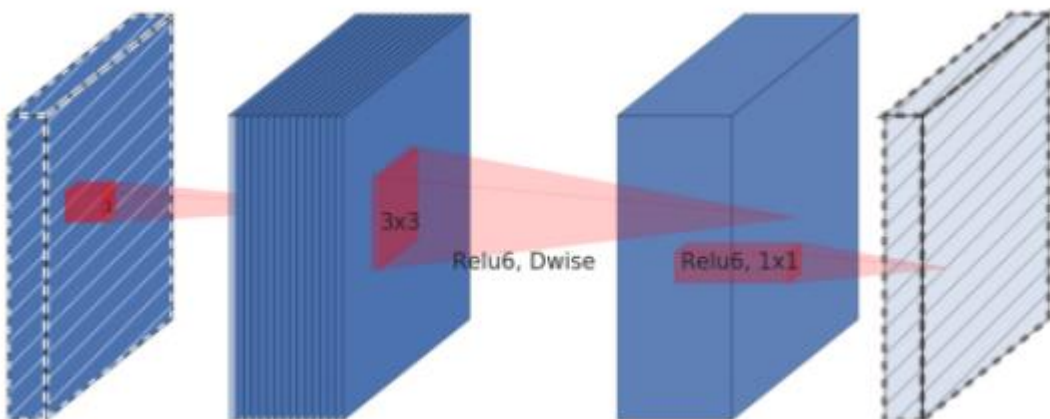


Рисунок 3.2 – Структура шарів мережі MobileNetV2

Повернемося до завдання, яке вирішуватиме CNN. Щоб передбачити, чи

правильно людина надягла маску, початковим етапом буде навчання моделі з використанням правильного набору даних.

Після навчання класифікатора потрібна точна модель виявлення осіб. Завдання - підвищити точність виявлення масок, не вимагаючи занадто великих ресурсів. Для виконання цього завдання використовувався модуль DNN із OpenCV, який містить модель виявлення об'єктів.

SSD з ResNet-10 як основа. Цей підхід допомагає виявляти обличчя в реальному часі.

### 3.4 Використовуваний набір даних

Для виявлення масок на обличчі доступно лише кілька наборів даних. Більшість їх або створені штучно, що неточно відбиває реальний світ, або набір даних сповнений шуму і неправильних міток.

У відкритому доступі вже є набір даних, зібраний у свою чергу з різних наборів зображень з відкритим вихідним кодом. Деякі з них – дані з набору даних Kaggle Medical Mask Dataset [27] від Миколая Вітковски та набору даних Праджни Бхандарі, доступного на PyImageSearch [28]. Набір даних Kaggle містить зображення людей у медичних масках та файли XML, що містять їх описи та маски (рис. 3.3). У цьому наборі даних було лише 678 зображень.

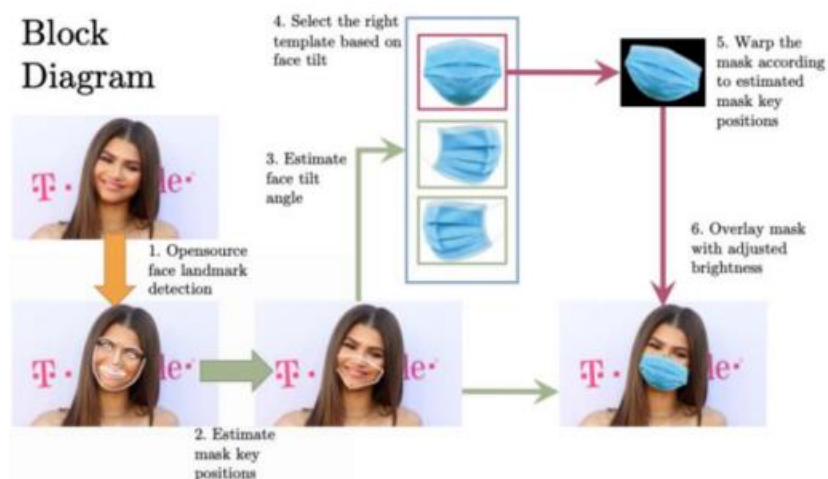


Рисунок 3.3 – Схема побудови тестового набору даних



В даний час датасет містить наступні 4 види масок (рис. 3.4).

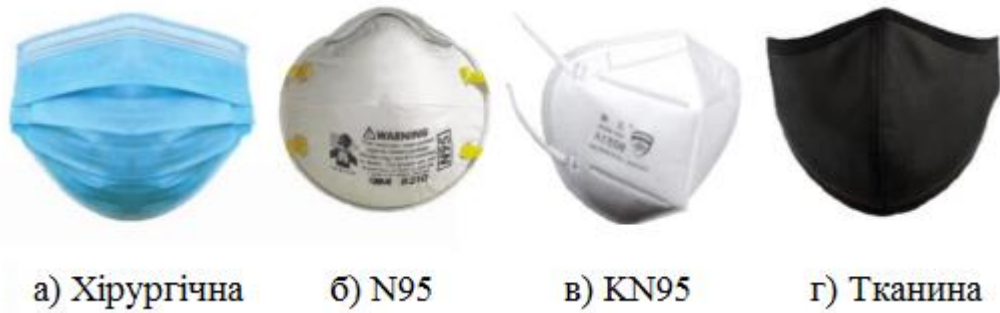


Рисунок 3.4 – Види масок у датасеті

Цей датасет можна використовувати для створення набору даних синтетичної маски обличчя, який можна використовувати для навчання різних прикладних мереж МН, таких як розпізнавання осіб для системи відвідуваності в приміщенні, виявлення масок, класифікація масок і т.д.

Інший набір даних штучної маски було взято у Праджни Бхандарі з PyImageSearch. У цьому наборі вже 1376 зображень. Це 690 зображень з маскою та 686 зображень без маски.

У штучний набір даних, було взято стандартні зображення обличчя та нанесено лицьові орієнтири. Орієнтири на обличчі дозволили визначити такі риси обличчя людини, як очі, брови, ніс, рот та лінію підборіддя. При цьому використовувався штучний спосіб створення набору даних. Маски просто накладалися на обличчя людей. Тому через те, що такі дані використовувати просто ризиковано, довелося відмовитися від їх використання. І в результаті набір даних складатиметься із зображень людей у масках та без масок, що буде більш ефективним та безпечним (рис. 3.5). Зрештою, було отримано набір даних, який включав 5521 зображення людей масках і 5521 зображення людей без масок.

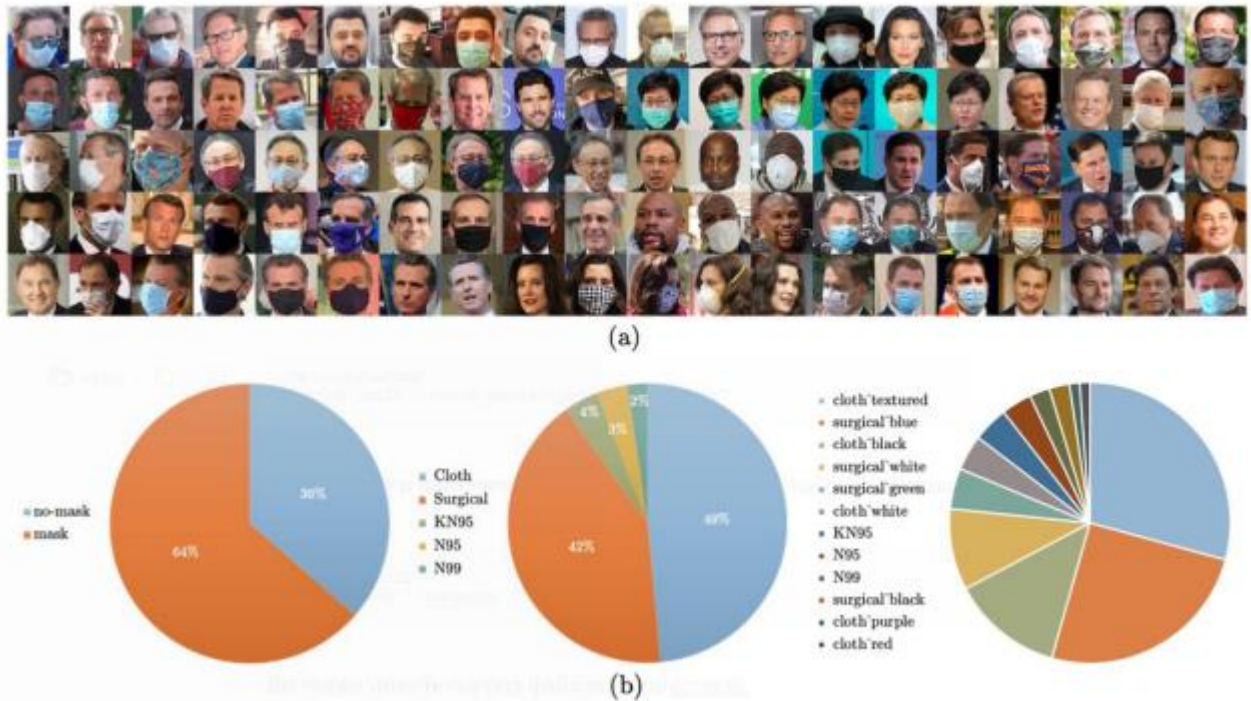


Рисунок 3.5 – Набір даних з PyImageSearch

Однак через особливості алгоритму, обробка 11042 зображень вимагає досить великої обчислювальної потужності. Тому було прийнято рішення скоротити цей набір даних, і було використано десяту частину від загальної кількості зображень.

### 3.5 Виявлення обличчів із використанням OpenCV

Для того, щоб розпізнати наявність взагалі будь-яких обличчів на зображенні, можна використовувати вбудовану модель від OpenCV. Як базова модель використовується архітектура «ResNet-10». Детектор Single Shot Multi-box схожий на техніку YOLO, яка робить лише один знімок для виявлення декількох об'єктів, присутніх на зображенні, за допомогою Multibox. Це значно швидший та точніший алгоритм виявлення об'єктів. Зображення, якими навчається модель, не розголошуються.

OpenCV надає дві версії моделі [29]:

- CaffeImplementation;
- реалізація TensorFlow;

Caffe - це середовище для глибокого навчання, розроблене як швидша, потужна та ефективна альтернатива в порівнянні з іншими методами виявлення об'єктів. Вона створена та управляється Berkeley AI Research (BAIR) та учасниками спільноти.

Для використання Caffe файли Caffemodel та prototxt завантажуються за допомогою методу readNet або методу readNetFromCaffe бібліотеки OpenCV. Після застосування моделі виявлення осіб ми отримуємо кількість виявлених осіб, розташування їх рамок, що обмежують, і оцінку достовірності цих прогнозів. Ці вихідні дані потім використовуються як вхідні дані для класифікатора наявності або відсутності маски на обличчі. Використання цього підходу виявлення осіб дозволяє виявляти особи реальному часі без значного використання ресурсів. Він також може виявляти особи в різній орієнтації, тобто ліворуч, праворуч, зверху та знизу, з хорошою точністю та виявляти захисні маски різних розмірів.

### **3.6 Алгоритм програми для навчання нейромережі**

Першим кроком до реалізації програми для розпізнавання масок є вибір даних для навчання. Спочатку малося на увазі використання всіх зібраних даних, проте, як виявилось, для цього потрібно дуже багато обчислювальної потужності. Як було сказано раніше, була взята десята частина від більш ніж 11 тисяч зображень. Цього виявилось достатньо навчання мережі і це заощадило обчислювальну потужність.

Другим і, можливо, найважливішим кроком є навчання ШНМ.

Насамперед завантажуються і перетворюються зображення, а також оголошуються змінний, які відповідають за навчання мережі. Це швидкість навчання, кількість епох та розмір підвибірки. Потім зображення поділяються на тренувальні та тестові. Кількість тестових даних було взято 10%.

Після цього необхідно оголосити передбачену мережу MobileNetV2 як основну, до якої потім додадуться нові шари. Необхідно вимкнути параметр, який відповідає за навчання цієї мережі, оскільки вона вже навчена. Потім до

мережі додається шар пулінга, для зменшення зображень. В якості функції вибірки була вибрана функція середнього значення, однак, можна було скористатися і функцією максимального значення. На малюнках 3.5 та 3.6 видно різниця в роботі двох цих функцій. Структура мережі змінюється на повнозв'язну. Для цього додається шар Flatten, який перетворює зображення, що вийшло у вектор. Далі додається шар нейронів з ФА ReLU, потім виключає шар (для запобігання перенавченню мережі) та шар нейронів ФА SoftMax, яка добре підходить для вирішення задачі класифікації. Загальний алгоритм програми навчання мережі показаний на рис. 3.7.



Рисунок 3.7 – Алгоритм роботи програми навчання мережі

Після того, як структура мережі зібрана, можна розпочинати її навчання. На даному етапі, акцентуємо увагу на завантаженні нашого набору даних з пам'яті комп'ютера для навчання моделі виявлення маски на обличчі (з використанням TensorFlow), а потім зберігаємо модель з кращими показниками. На рис. 3.8 представлена модель CNN.

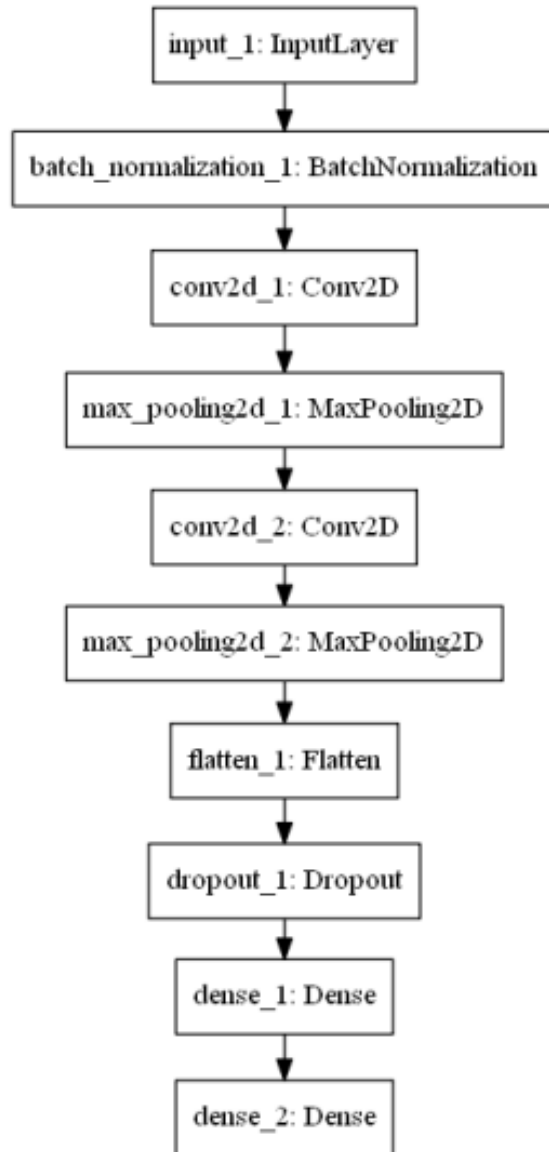


Рисунок 3.8 – Модель CNN

Розгортання: після того, як неймережева модель для детектування маски навчена, переходимо до завантаження детектора маски з диска для застосування етапу виявлення осіб, а потім класифікації обличчя в захисній масці або без неї.

Крок 1. Завантажити та обробити дані для тренування CNN.

Крок 2. Реалізувати архітектуру CNN (вхід: 224\*224\*3).

Крок 3. Навчити модель на тренувальній вибірці з за допомогою бібліотек Tensorflow, Numpy та Matplotlib та зберегти модель для подальшого застосування.

Крок 4. Розгорнути навчену модель для обробки кадрів та для оцінок ознак у відеопотоку у реальному часі.

Крок 5. Протестувати модель відеопотоку в реальному часі.

Вихідний код процесу навчання та перевірки моделей наведений в Додатку А.

### 3.7 Результати тестування моделі

Для того, щоб вбудувати видимі риси обличчя у 128-вимірність, нам потрібно обрізати виявлені обличчя, і для цього ми вибрали InceptionV3 за легку архітектуру та точність [30]. Він реалізований за допомогою Keras – відкритої нейромережевої бібліотеки, що працює на базі бібліотеки TensorFlow, розробленої для реалізації завдань CV [31]. Такий підхід є ефективним, оскільки нейромережа намагається визначити наявності чи відсутності маски, лише у тому фрагменті зображенні, де є особи. Це дозволяє покращити швидкість розпізнавання маски на обличчі.

Даний метод полегшує завдання розпізнавання, оскільки для порівняння ознак, які використовують попередньо навчену модель глибокого навчання (див. рис. 3.9), потрібно багато часу, і нам потрібно виконати це завдання за мілісекунди.

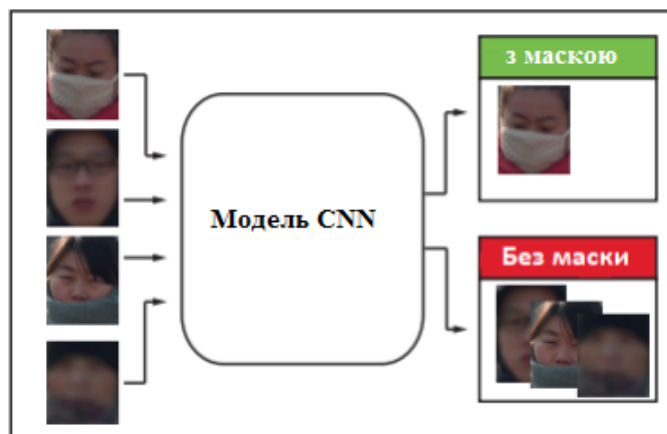


Рисунок 3.9 – Зображення обличчя для виявлення масок

На етапі кодування ми нормалізували вбудоване обличчя (див. рис. 3.10), що означає масштабування значень, виміряних в іншому діапазоні, до

стандартної шкали та нормалізацію наших векторів, було використано бібліотеку scikits-learn.

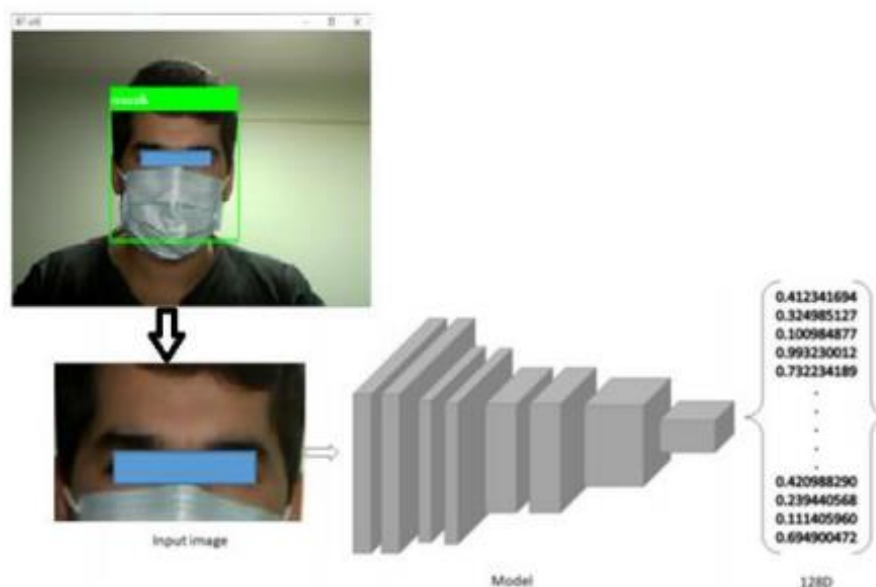


Рисунок 3.10 – Перетворення зображення обличчя на 128-вимірний вектор

Після виконання попередньої роботи з отримання та підготовки даних для навчання моделі, була реалізована архітектура та навчена модель CNN.

Результати навчання наведено на рис. 3.11. Незавжно помітити, що модель добре навчилася і досягла точності розпізнавання, що дорівнює 98%.

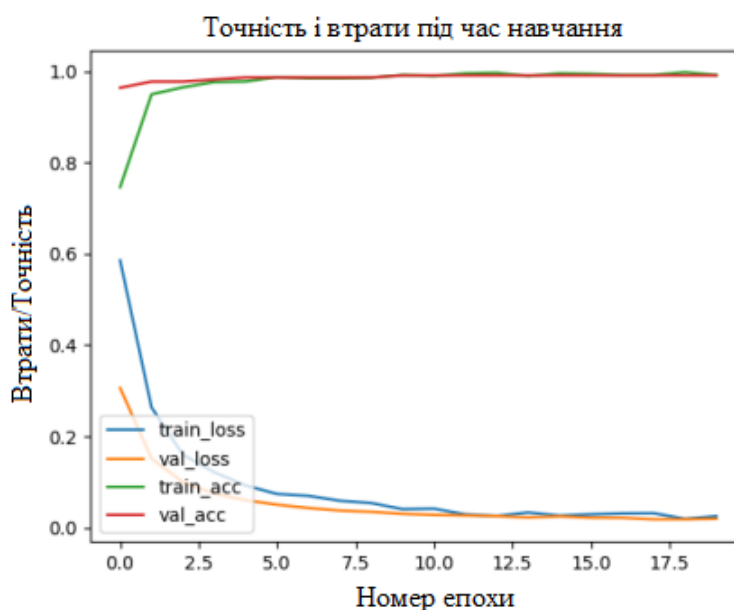


Рисунок 3.11 – Результат проведення навчання моделі

Також результати проведених експериментів можна переглянути в додатку А.

Останнім кроком буде використання навченої моделі. Для використання був написаний простий графічний інтерфейс з використанням бібліотеки python tkinter. Діаграму класів можна побачити на рис. 3.12.

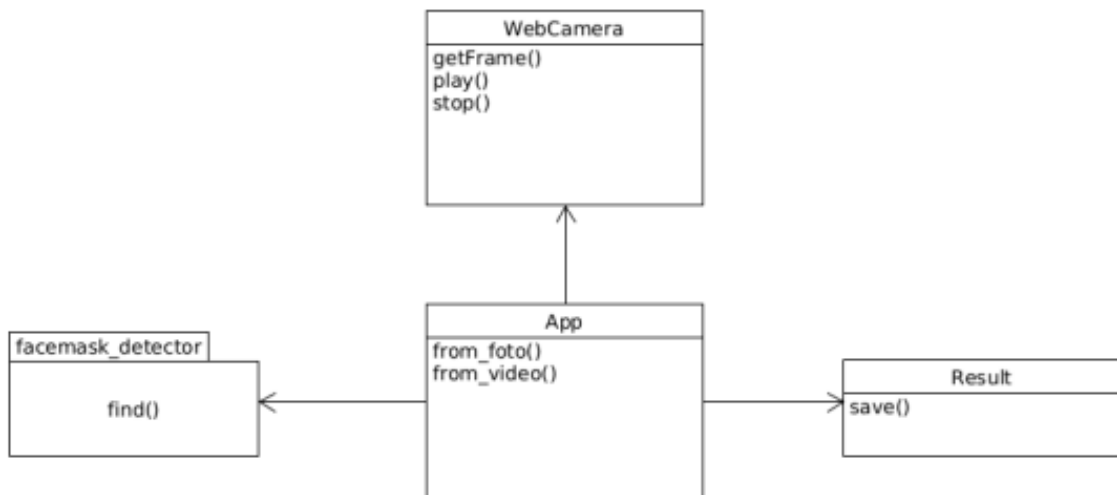


Рисунок 3.12 – Діаграма класів програми, яка використовує навчену модель

Під час запуску програми та отримання доступу до камери з'явиться спеціальне вікно, де виводяться результати РО. Після етапу розпізнавання осіб проводиться детектування захисної медичної маски на обличчі, а результат детектування виводиться на екрані (рис. 3.13).

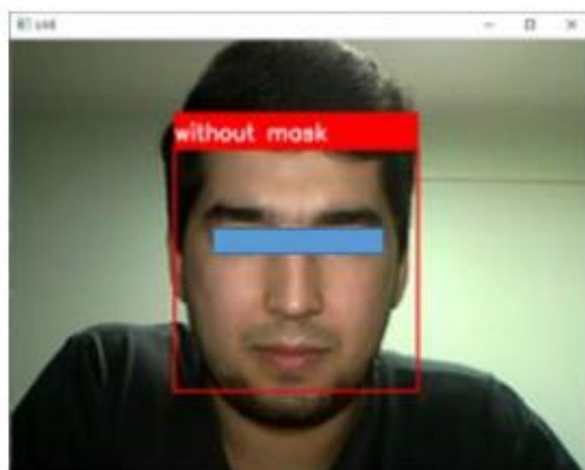


Рисунок 3.13 – Результат тестування програми для осіб без захисної маски



Переконавшись у працездатності програми для РО без захисної маски, було проведено тестування на обличчі людини в захисній масці (рис. 3.14).



Рисунок 3.14 – Результат тестування програми для осіб із захисною маскою

При проведенні тестування були одержані очікувані результати, що полягають у тому, що застосування навчених CNN дозволяє виявляти маски на розпізнаних обличчях. Досягнута точність розпізнавання масок на рівні 98%, що дозволяє досить добре визначити чи носить людина захисну маску чи ні.

## РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Долікарська допомога при ураженні електричним струмом

Удар електричним струмом є поширеною травмою і часто може закінчитися летальним випадком. Ураження електричним струмом відбувається під час контакту тіла людини з джерелом електричної енергії під напругою. Це реакція організму людини на проходження електричного струму через тіло. Вона проявляється по різному, від легких потрясінь до небезпечних здоров'ю травм, які можуть вплинути на тканини в організмі.

Шкода завдана електричним струмом залежать від кількох факторів: наскільки висока була напруга, яка область тіла була вражена і від виду струму. Фізичні наслідки для людини можуть коливатися від опіків частин тіла до серйозних вражень внутрішніх органів [32].

Часто люди стикаються з підвищеним ризиком ураження високою напругою. Низька напруги не наносить серйозних травм для людини, а з іншої сторони висока напруги, яка має більше 500 В, може призвести до серйозного пошкодження тканин. У дітей або підлітків при ураженні електричним струмом в діапазоні від 110 до 200 В можуть виникнути значні травми. Зазвичай це трапляється під час порушення техніки безпеки при роботі з електричними приладами, які є в побуті. Це можуть бити електричні шнури, подовжувачі, несправні розетки та багато чого іншого.

Виділяють чотири основні фактори від яких залежить вплив електричного струму на організм: величина струму, що протікає через тіло; органи, через які проходить струм; час, протягом якого струм вражає тіло; частота струму.

Фізичні наслідки на пряму залежать від величини струму, яка вражає тіло людини. Струм менший за 1 мА не завдає жодної шкоди людині фізичного ефекту. При 1 мА можуть відчуватися слабкі поколювання, а при 5 мА – легкий поштовх. Однак при струмі величезною від 6 до 25 мА людина може відчувати больовий шок і деяку втрату контролю над м'язами.

При ураженні електричним струмом від 50 до 150 мА може спричинити у людини сильний біль, м'язове скорочення і навіть призвести до зупинки дихання. У певних випадках можливий летальний результат для людини. Струм від 1000 до 4300 мА призводить до ймовірної смерті, тому що дана напруга спричиняє пошкодження нервових зав'язків, м'язових скорочення та порушення ритму серця. До 10 000 мА електричний струм спричиняє важкі опіки шкіри людини та зупинку серця. Висока ймовірність смерті.

Найпоширеніші ознаки та симптоми враження електричним струмом включають:

- втрата свідомості;
- ускладнення або зупинка дихання;
- опіки, які виникають там, де струм входить і виходить з тіла;
- зупинка серця;
- слабкий і непостійний пульс або його припинення.

Перш за все, що потрібно зробити при першій допомозі, це відключити джерело живлення. Вимкнути електропостачання, від'єднати електроприлад від джерела електричного струму або вимкнути блок запобіжників, якщо він знаходиться неподалік. Не потрібно намагатися підходити близько до жертви, якщо не переконані, що це безпечно і живлення вимкнено.

Потрібно бути обережним у вологих місцях, тому що вода є електричним провідником і рятівник може стати теж жертвою. Якщо людина не впевнена щодо вологості поверхні, потрібно відключити основне електропостачання будинку. У випадку коли це неможливо, використати підручний предмет, який не є провідником і відокремити людину від джерела струму. Це може бути дерев'яна або пластмасова річ.

Після того як постраждалого відокремили від джерела електрики, потрібно викликати швидку і надати першу медичну допомогу. Далі потрібно визначити стан жертви. Перевірити, чи людина у свідомості і дихає. У складних випадках у жертви може бути слабкий пульс або його відсутність. Можливо, що дихання зупинилося. Якщо людина втратила свідомість і перестала дихати, потрібно почати серцево-легеневу реанімацію. Руки розташовуємо в центрі грудної, одна

на іншу. Сильно і швидко натиснути 30 раз приблизно до третини діаметра грудної клітки. Після кожного натискання на грудну клітку робиться два рятувальні вдихи. Потрібно відкинути голову потерпілого назад і підняти підборіддя. Затиснути ніс і створити повне ущільнення. Далі подути потерпілому в рот і подивитися, чи підніметься грудна клітка. Потрібно продовжувати робити натискань на грудну клітку та вдих, поки не прибуде медична допомога або людина не почне сама дихати. Якщо потерпілий живий, перемістити його у зручне йому положення подальше від небезпеки. Можна запобігти шоку, поклавши людину рівно на землю, з головою трохи нижче тіла [33].

Якщо людина при свідомості, нормально дихає і на тілі є опіки, потрібно накрити їх звичайною харчовою плівкою або іншою неклеюю пов'язкою, але без мазі чи лосьйону. Якщо кровотеча у потерпілого, може знадобитися компресія та джгут.

При роботі з соціальною мережею користувач повинен знати і вміти як правильно поводитися з ПК, тому що людина перебуває у непосредньому контактi з джерелом напруги. Удар електричним струмом є потенційно смертельною травмою. Негайна медична допомога важлива, щоб запобігти серйозним травмам і смерті. Для запобігання уражень електричним струмом при роботі за ПК слід встановити додаткові захисні пристрої, що забезпечують недоступність токопровідних частин для дотику. Для зменшення небезпеки використовувати розділовий трансформатор для розв'язки з основною мережею.

Удар електричним струмом є потенційно смертельною травмою. Негайна медична допомога важлива, щоб запобігти серйозним травмам і смерті.

## **4.2 Вимоги ергономіки до організації робочого місця оператора ПК**

Робоче місце – це ділянка простору, яка облаштована необхідним обладнанням, відповідно до трудової діяльності, для виконання поставлених завдань.

Правильно побудоване робоче місце повинне забезпечувати:

- найкраще розміщення обладнання і предметів праці;

- не допускати дискомфорту;
- підвищувати продуктивність праці;
- зменшувати втому працівника.

Розмір робочого місця повинен бути таким, щоб людина не виконувала лишніх рухів і не відчувала дискомфорту під час виконання роботи. Також для працівника важливо мати змогу змінити робочу позу, наприклад, положення тулуба, рук або ніг. Потрібно мінімізувати або звести до нуля всі незручності положення тіла [34].

Різні дослідження заявляють, що при правильному проектуванні робочого місця продуктивність людини може зрости від 15-25%. Такі фактори як рівень освітлення, вологість повітря, температура, шум, вібрація, токсичність, мають значний вплив на умови життєдіяльності і працездатності людини.

Антропометричні вимоги визначають відповідність робочого місця до фізіологічних параметрів тіла людини як зріст і розміри тіла. Індикатором цього є правильна робоча поза, відсутність дискомфорту, оптимальні зони досягнення, раціональні рухи. Психофізіологічні та фізіологічні вимоги формують відповідність обладнання і робочого місця можливостям співробітника щодо розуміння, обробки даних, пошук і реалізації рішень.

Організація робочого місця передбачає наступні пункти:

- раціональне положення робочого місця у приміщенні;
- вибір робочих меблів відповідно до фізіологічних характеристик працівника;
- правильне компонування і розміщення обладнання на робочих місцях;
- урахування особливостей та характеру професійної діяльності.

До загальних принципів організації робочого місця відносять:

- робоче місце повинне містити тільки ті предмети, які беруть участь у робочому процесі, але не заважати йому;
- предмети, які часто використовуються у роботі, розміщуються ближче, ніж ті речі, якими користуються рідше;
- предмети, які беруться лівою рукою, повинні розміщуватися зліва, а предмети, які використовуються правою рукою — справа;

- якщо при роботі з предметом працівник використовує дві руки, то він розміщується з урахуванням зручності захоплення його двома руками;
- робоче місце не повинно бути засмічене;
- необхідна оглядовість повинна бути забезпечена при правильній організації робочого місця.

Робоча поза – це найбільш тривале положення тіла працівника протягом робочого дня. При зручній робочій позі забезпечується стійкість положення тулуба, ніг, рук, голови і витрачається мінімальний запас енергії та максимальну продуктивність [35].

Сидячи і стоячи – дві найпопулярніших пози у робочому процесі. При проектуванні робочого місця потрібно враховувати, що з фізичним навантаженням бажана поза стоячи, а при малих зусиллях – сидячи. При роботі стоячи, людина стомлюється більше ніж сидячи. У відсотковому еквіваленті це на 10% більше енергії. При додатковому навантаженні підвищується артеріальний і венозний тиск крові, розширення вен, пошкоджуються ступені та викривляється хребет. У свою чергу при сидячій роботі нижня частина тіла розслаблена, а основне навантаження спрямоване на м'язи шиї, спини, таза, стегон. При неправильній сидячій позі розвивається застій крові у ногах, а якщо пальці виконують багато роботи можливе запалення суглобів.

Організація робочого місця при використанні ПК повинна відповідати усім ергономічним вимогам. Ключові ергономічні вимоги до проектування робочого місця оператора ПК зображені на рисунку 4.1.

При роботі з персональним комп'ютером потрібно:

- зменшувати кількість статичних напружень;
- розподіляти кількість і час статичних напружень;
- змінювати робочі пози під професійної діяльності.

Саме вибір правильної робочої пози визначається від впливу багатьох факторів. Одні з найважливіших це – кількість зусиль яка прикладається, величина робочої зони, відношення висоти робочої поверхні і ростом працівника.

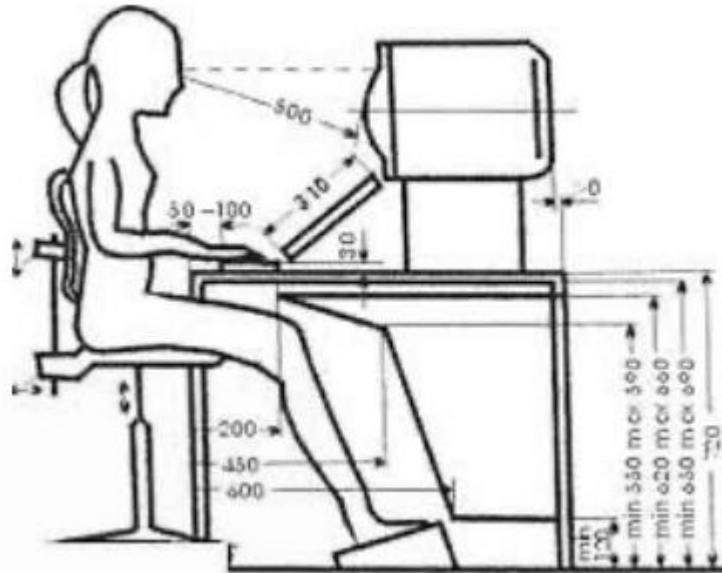


Рисунок 4.1 – Робочий стіл і розміщення користувача ПК

При використанні соціальної мережі користувач повинен дотримуватися вище зазначених правил, які будуть сприяти комфортній і продуктивній роботі. Важливо регулярно робити короткі перерви. Часта зміна заняття – кращий спосіб уникнути можливих неприємностей.

## ВИСНОВКИ

В результаті виконання цієї роботи було проведено дослідження різних методів та алгоритмів виявлення осіб і розпізнавання об'єктів.

Основні результати, отримані в роботі:

- розглянуто проблему виявлення масок, актуальну для додатків моніторингу пандемії COVID-19;
- досліджено та вивчено різні ефективні методи та підходи, що дозволяють визначити правильність одержуваних результатів;
- вивчено недоліки набору даних для виявлення масок, а також проведено експерименти, при проведенні яких було розглянуто: ефективність існуючих методів детектування осіб у медичних масках, реалізації методів розпізнавання, спрямованих на виявлення правильно одягнених масок, якість роботи існуючих моделей виявлення масок,
- запропоновано спосіб розпізнавання осіб та виявлення захисної маски на обличчі у відеопотоці. Він використовує у своїй основі методи CV та МН.

Розроблено програмний засіб, котрий може використовуватися при вирішенні різних завдань відеоаналітики та має практичне застосування в системах контролю доступу та ідентифікації особи, а також для виявлення осіб, які порушують санітарно-епідеміологічні вимоги.

Таким чином, поставлені завдання було повністю вирішено. Шляхом розширення навчальної вибірки якіснішими зображеннями і збільшенням часу навчання CNN, можна досягти високої точності роботи алгоритму. Надалі планується оптимізація моделі, у тому числі шляхом зміни її архітектури.

Відображена в дослідженні модель дає більші можливості з невеликим для моделі розмірами. Це свідчить про можливе подальше її застосування на різних пристроях. Можливим неоптимізованим місцем може бути програма, котра використовує навчену модель.



## ПЕРЕЛІК ДЖЕРЕЛ

1. Коронавірус: що треба знати про захисні маски [Електронний ресурс]. – Режим доступу: <https://www.dw.com/uk/koronavirus-shcho-treba-znaty-pro-zakhysni-masky/a-56300357> (Дата звернення: 26.03.2023).
2. Рекомендації щодо використання масок під час пандемії COVID-19 [Електронний ресурс]. – Режим доступу: [https://doh.wa.gov/sites/default/files/legacy/Documents/1600/coronavirus/ClothFacemasks\\_Ukrainian.pdf](https://doh.wa.gov/sites/default/files/legacy/Documents/1600/coronavirus/ClothFacemasks_Ukrainian.pdf) (Дата звернення: 27.03.2023).
3. Walid Hariri. Efficient Masked Face Recognition Method during the COVID-19 Pandemic / arXiv:2105.03026. – 2021.
4. Кашкаров А.П. Системи відеонаблюдения. Практикум – К.: Феникс, 2014. – 128 с.
5. Розумна система безпеки [Електронний ресурс]. – Режим доступу: <https://www.lsvisionhd.com/uk/products/Smart-Security-System.htm> (Дата звернення: 29.03.2023).
6. Система контролю і управління доступом [Електронний ресурс]. – Режим доступу: <https://vistplus.com/it-poslugi/skud/> (Дата звернення: 31.03.2023).
7. Відстань Гаусдорфа [Електронний ресурс]. – Режим доступу: [https://www.wiki-data.uk-ua.nina.az/Відстань\\_Хаусдорфа.html](https://www.wiki-data.uk-ua.nina.az/Відстань_Хаусдорфа.html) (Дата звернення: 30.03.2023).
8. Jones, M., Viola, P. (2001) Robust Real-Time Face Detection. [Електронний ресурс]. – Режим доступу: <http://www.vision.caltech.edu/html-files/EE148-2005-Spring/pprs/viola04ijcv.pdf> (Дата звернення: 31.03.2023).
9. Open Source Computer Vision [Електронний ресурс]. – Режим доступу: [https://docs.opencv.org/master/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/master/d6/d00/tutorial_py_root.html) (Дата звернення: 31.03.2023).
10. P. Viola and M. Jones, “Robust real-time object detection,” in International Journal of Computer Vision, 2001.
11. Субботін С.О. Нейронні мережі : теорія та практика: навч. посіб. – Житомир : Вид. О. О. Євенок, 2020. – 184 с.

12. Nikhil Ketkar, Jojo Moolayil. Deep Learning with Python. 2nd Ed. – Apress, 2021 – 306 p.
13. Дуда Р., Харт П. Распознавание образов и анализ сцен. – М.: Мир, 1976 – 512 с.
14. Хайкин, Саймон. Нейронные сети: полный курс, 2-е издание. : Пер. с англ. – М. : Издательский дом “Вильямс”. – 2006. – 1104 с.
15. Литвин В.В, Пелещак Р.М., Висоцька В.А. Глибинне навчання.- Львів: ЛП, 2021 – 264 с.
16. Новотарський М.А., Нестеренко Б.Б. Штучні нейронні мережі: обчислення // Праці Інституту математики НАН України. – Т50. – Київ: Ін-т математики НАН України, 2004. – 408 с.
17. A Comprehensive Guide to Convolutional Neural Networks [Електронний ресурс] - Режим доступу: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (дата звертання: 16.05.2023).
18. Кононова К. Ю. Машинне навчання: методи та моделі. – Харків: ХНУ імені В. Н. Каразіна, 2020. – 301 с.
19. Басюк Т.М. Машинне навчання: Навчальний посібник. - Львів: Видавництво «Новий Світ - 2000», 2021. - 315 с.
20. Ciresan U. Meier, J. Masci, L. M. Gambardella, J. Schmidhuber Flexible, High Performance Convolutional Neural Networks for Image Classification. Proceedings of the Twenty-Second international joint conference on Artificial Intelligence. 2011. P. 1237–1242.
21. Порівняльне дослідження методів розпізнавання облич. [Електронний ресурс]. – Режим доступу: <http://matmod.dstu.dp.ua/article/view/246871> (Дата звернення: 04.04.2023).
22. OpenCv [Електронний ресурс] – Режим доступу: <https://opencv.org/about/> (дата звертання 24.11.22).
23. TensorFlow Core [Електронний ресурс] – Режим доступу: <https://www.tensorflow.org/tutorials> (дата звертання: 05.04.2023).
24. NumPy [Електронний ресурс] - Режим доступу: <http://www.numpy.org/>

(дата звертання 07.04.23).

25. Matplotlib: Visualization with Python [Електронний ресурс] - Режим доступу: <https://matplotlib.org/> (дата звертання 07.04.23).

26. MobileNetV2 [Електронний ресурс] - Режим доступу: <https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c> (дата звертання 10.04.23).

27. Medical Mask Dataset [Електронний ресурс] - Режим доступу: <https://www.kaggle.com/datasets/shreyashwaghe/medical-mask-dataset> (дата звертання 12.04.23)

28. The fastest way to learn OpenCV, Object Detection, and Deep Learning [Електронний ресурс] - Режим доступу: <https://pyimagesearch.com/> (дата звертання 12.04.23)

29. Open CV Library [Електронний ресурс] – Режим доступу: <http://opencv.org/downloads.html> (дата звертання: 05.04.2023).

30. InceptionV3 [Електронний ресурс] – Режим доступу: <https://keras.io/api/applications/inceptionv3/> (дата звертання: 15.04.2023).

31. Keras [Електронний ресурс] – Режим доступу: <https://keras.io/> (дата звертання: 15.04.2023).

32. Удар струмом: перша допомога [Електронний ресурс] – Режим доступу: <http://tomrda.gov.ua/news/578646863743857435/> (дата звертання: 15.05.2023).

33. Перша допомога при ураженні електричним струмом [Електронний ресурс]. – Режим доступу: <https://bozhedarivskaselrada.gov.ua/news/1576497483/> (дата звертання: 15.05.2023).

34. Ергономічні вимоги до організації робочих місць [Електронний ресурс] – Режим доступу: [https://pidru4niki.com/14821111/bzhd/ergonomichni\\_vimogi\\_organizatsiyi](https://pidru4niki.com/14821111/bzhd/ergonomichni_vimogi_organizatsiyi) (дата звертання: 15.05.2023).

35. Охорона праці в офісі. Вимоги до робочого місця офісного працівника [Електронний ресурс] – Режим доступу: <https://gc.ua/uk/oxorona-praci-v-ofisi-vimogi-do-robochogomiscya-ofisnogo-pracivnika/> (дата звертання: 15.05.2023).

# ДОДАТКИ

**Фрагмент вихідного коду процесу навчання та перевірки моделей**

```
import tensorflow as tf
from tensorflow import keras
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
# Додаткові бібліотеки
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
# ініціалізуємо початкову швидкість навчання, кількість епох для
# тренування
INIT_LR = 1e-4
EPOCHS = 10
BS = 32
# список даних (тобто зображень) та зображень класів
print("[INFO] loading images...")
imagePaths = list(paths.list_images(args["dataset"]))
data = []
labels = []
# перебираємо шляхи до зображень
for imagePath in imagePaths:
    # добуваємо мітку класу з імені файлу
    label = imagePath.split(os.path.sep)[-2]
    # завантажуюмо вихідне зображення (224x224) та обробляємо його
    image = load_img(imagePath, target_size=(224, 224))
    image = img_to_array(image)
    image = preprocess_input(image)
    # оновити списки даних та міток відповідно
    data.append(image)
    labels.append(label)
# перетворити дані та мітки на масиви за допомогою NumPy
data = np.array(data, dtype="float32")
labels = np.array(labels)
```

```

model = Sequential([
Conv2D(100, (3, 3), activation='relu', input_shape=(150, 150, 3)),
MaxPooling2D(2, 2), Conv2D(100, (3, 3), activation='relu'),
MaxPooling2D(2, 2),
Flatten(),

# Згладжує введення. Не впливає на розмір партії
Dropout(0.5),
Dense(50, activation='relu'),
Dense(2, activation='softmax')
])
model.compile(optimizer='adam',          loss='binary_crossentropy',
metrics=['acc'])

.....

# Створюємо контрольну точку зворотного виклику, щоб зберігати
найкращу модель після кожної епохи під час навчання:
checkpoint = ModelCheckpoint('model2-{epoch:03d}.model',
monitor='val_loss',verbose=0,save_best_only=True,mode='auto')

# Навчання моделі
history = model.fit_generator(train_generator,
epochs=10,
validation_data=validation_generator,
callbacks=[checkpoint])

```

### Візуалізація результату навчання за епохами

```

Epoch 1/10 132/132 [=====] - 171s 1s/step - loss: 0.6702
- a cc: 0.7141 - val_loss: 0.1702 - val_acc: 0.9588
Epoch 2/10 132/132 [=====] - 167s 1s/step - loss: 0.3395
- a cc: 0.8730 - val_loss: 0.0491 - val_acc: 0.9072 121
Epoch 3/10 132/132 [=====] - 166s 1s/step - loss: 0.3382
- a cc: 0.8852 - val_loss: 0.2210 - val_acc: 0.9021

```

Epoch 4/10 132/132 [=====] - 166s 1s/step - loss: 0.2634  
- a cc: 0.8958 - val\_loss: 0.0169 - val\_acc: 0.9742  
Epoch 5/10 132/132 [=====] - 171s 1s/step - loss: 0.2149  
- a cc: 0.9118 - val\_loss: 0.1482 - val\_acc: 0.9588  
Epoch 6/10 132/132 [=====] - 166s 1s/step - loss: 0.2057  
- a cc: 0.9278 - val\_loss: 0.0036 - val\_acc: 0.9691  
Epoch 7/10 132/132 [=====] - 166s 1s/step - loss: 0.2048  
- a cc: 0.9346 - val\_loss: 0.0698 - val\_acc: 0.9588  
Epoch 8/10 132/132 [=====] - 166s 1s/step - loss: 0.1803  
- a cc: 0.9346 - val\_loss: 0.0073 - val\_acc: 0.9845  
Epoch 9/10 132/132 [=====] - 178s 1s/step - loss: 0.1570  
- a cc: 0.9414 - val\_loss: 0.0013 - val\_acc: 0.9845  
Epoch 10/10 132/132 [=====] - 167s 1s/step - loss:  
0.1334 - a cc: 0.9536 - val\_loss: 0.0335 - val\_acc: 0.9536

У процесі навчання моделі відображаються метрики втрати (loss) та точності (аcurracy). Ця модель досягає на тренувальних даних точності рівної приблизно 0.98 (98%).