

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теоретичної кібернетики

Кваліфікаційна робота

на здобуття ступеня бакалавра

за спеціальністю 122 Комп'ютерні науки

на тему:

**ЗАСТОСУВАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ
АНАЛІЗУ ДАНИХ (DATA MINING) І ПРОГНОЗУВАННЯ
ПРОЦЕСІВ**

Виконав студент 4-го курсу
Попов Клим Олександрович

(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Трохимчук Ростислав Миколайович

(підпис)

Засвідчую, що в цій роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент

(підпис)

Роботу розглянуто й допущено до
захисту на засіданні кафедри
теоретичної кібернетики

«____»_____

_____2021 р., протокол № __

Завідувач кафедри

Ю. В. Крак

(підпис)

РЕФЕРАТ

Обсяг роботи 41 сторінка, 32 ілюстрації, 20 джерел посилань.

КЛАСИФІКАЦІЯ, НЕЙРОННА МЕРЕЖА, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, ЗГОРТКОВА НЕЙРОМЕРЕЖА, СЕПАРАБЕЛЬНИЙ ШАР, ПОКАЗНИКИ ЕФЕКТИВНОСТІ, МАТРИЦЯ НЕВІДПОВІДНОСТЕЙ.

Об'єктом роботи є інтелектуальний аналіз даних за допомогою нейронної мережі, що класифікує знімки легень та ставить відповідний діагноз.

Предметом роботи є згорткова нейронна мережа та використання її результатів.

Метою роботи є дослідити ефективність нейронних мереж у медичній сфері та побудова найкращої моделі для задачі класифікації знімків.

Методи розроблення: інтелектуальний аналіз даних, згорткова нейронна мережа, архітектура Xception, метод оберненого поширення помилки.

Інструменти розроблення: TensorFlow 2.4.1, Anaconda Navigator, Jupiter Lab, мова програмування Python 3.

Результати роботи: виконано загальний огляд інтелектуального аналізу даних та нейронних мереж. Описаний алгоритм градієнтного спуску та зворотного поширення помилки. Побудована модель для класифікації та проведений аналіз її точності.

Отриманий продукт не втрачає актуальність та може бути використаний для подальшого аналізу знімків.

ЗМІСТ

СКРОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	4
ВСТУП	5
РОЗДІЛ 1 КЛАСИФІКАЦІЯ ЯК ВИД DATA MINING	7
1.1 Поняття data mining	7
1.2 Задача класифікації.....	9
1.3 Показники ефективності	9
РОЗДІЛ 2 НЕЙРОННА МЕРЕЖА ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ	12
2.1 Поняття нейронної мережі.....	12
2.2 Градієнтний спуск та метод зворотного поширення помилки	15
2.3 Згортова нейронна мережа.....	18
2.4 Архітектура Xception.....	25
РОЗДІЛ 3 КЛАСИФІКАЦІЯ РЕНТГЕНІВСЬКИХ ЗНІМКІВ	26
3.1 Підготовка та аналіз вхідних даних.....	26
3.2 Реалізація моделі.....	30
3.3 Навчання та тестування нейронної мережі.....	32
РОЗДІЛ 4 ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ЗАХВОРЮВАНОСТІ НА ПНЕВМОНІЮ	36
ВИСНОВКИ	39
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	40

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

SQL – structured query language, мова для створення та модифікації даних;

OvA – один проти всіх;

OvO – один проти одного;

MNIST – набір зображень, що включає рукописні цифри;

ANN – artificial neural networks, штучна нейронна мережа;

CNN – convolutional neural network, згорткова нейронна мережа;

ReLU – rectified linear unit, активаційна функція, що визначена як додатна частина аргументу;

MSE – mean squared error, середня квадратична помилка;

PNA – пневмонія.

ВСТУП

Оцінка сучасного стану об'єкта дослідження. Нейронні мережі – це сучасний і ефективний інструмент, що використовують у різних сферах нашого життя: у фінансовій сфері, мультимедійній, ігровій тощо. Вони передбачають курси валют, аномальні банківські переводи, покращують графіку, оживляють фото, розпізнають зображення та голоси. Особливо важливим інструментом ця технологія виступає у медичному напрямку. За допомогою цієї технології роблять висновки про якість медичних препаратів, визначають діагноз за сукупністю симптомів. За останні роки комп'ютерна потужність досягнула достатнього рівня, щоб обчислювати велику кількість операцій, що потребують нейронні мережі. Це дозволяє класифікувати зображення, що є великим кроком прогресу у багатьох сферах. Нейронні мережі дозволяють обчислювати дані великого масштабу та робити висновки, неочевидні для експерта.

Актуальність роботи та підстави для її виконання. Під час пандемії дуже важливо робити правильні рішення щодо карантинних обмежень та необхідності вакцинації. Робота лікарів стала дуже необхідною, тому автоматизація будь-яких функцій дає змогу звільнити їх від надмірного навантаження. Автоматичний діагноз дозволяє відстежувати стан пацієнтів, визначати тип та рівень захворюваності. Також важливим є спостереження за спалахами пандемії на різних територіях, що дозволяє правильно обмежувати рух транспорту та встановлювати відповідні карантинні обмеження.

Мета й завдання роботи. Метою роботи є аналіз вхідних зображень та побудова оптимальної нейронної мережі для класифікації знімків. Для досягнення мети поставлено та виконано такі завдання:

- Дослідити поняття нейронних мереж.
- Дослідити згорткові нейронні мережі для обробки зображень.
- Визначити найкращу структуру для поставленої задачі.
- Підготувати та проаналізувати вхідні дані.
- Побудувати та навчити нейронну мережу.
- Розглянути методи аналізу вихідних результатів.

Об'єкт, методи й засоби розроблення. Об'єктом роботи є інтелектуальний аналіз знімків легень для визначення діагнозу, а саме класифікація зображень. Методи розроблення: інтелектуальний аналіз даних, згорткова нейронна мережа, архітектура Xception, метод оберненого поширення помилки. Інструменти розроблення: TensorFlow 2.4.1, Anaconda Navigator, Jupiter Lab, мова програмування Python 3.

Можливі сфери застосування. Дана система може застосовуватись у лікарнях та госпіталях для відстеження стану хворих та встановлення діагнозу.

РОЗДІЛ 1 КЛАСИФІКАЦІЯ ЯК ВИД DATA MINING

1.1 Поняття data mining

Data mining – це загальна назва множини алгоритмів та методів, для знаходження нових даних, з певної сукупності наявних. [1] Знайдені дані мають бути не очевидними з точки зору експерта, практичними та корисними з точки зору бізнесу. Українською мовою цей термін перекладають як інтелектуальний аналіз даних.

Розглянемо основні задачі, що які потрапляють під визначення задач для Data mining:

- Регресійний аналіз – це сукупність методів, що дозволяють встановити зв'язок між парами значень незалежної та залежної змінних. [2]
- Класифікація – це надання об'єктам мітки класу, список яких є попередньо визначеним. [3]
- Асоціація – виявлення взаємозв'язку між певними подіями. Прикладом для такого зв'язку є аналіз ринкового кошика.
- Кластерний аналіз – сукупність методів, що виконують розбиття сукупності об'єктів на кластери. Всередині кожного кластеру об'єкти схожі один на одного за певними властивостями. [4]
- Пошук аномальних спостережень – виявлення шаблонів, що є не звичними для основної частини даних. Прикладом може слугувати пошук підозрілих фінансових переводів.
- Послідовні шаблони – пошук закономірностей та взаємозв'язків між певними подіями в часі.

Також заведено вважати, що статистичні методи, різного роду статистичні критерії також є частиною data mining. Невіддільною частиною також є штучний інтелект.

Знаходження масштабних закономірностей можливе при наявності великої бази даних. Найчастіше для підготовки необхідні дані витягуються за допомогою SQL.

Тож підсумувавши різні погляди на цей великий розділ аналізу, можна побудувати наступну ілюстрацію:

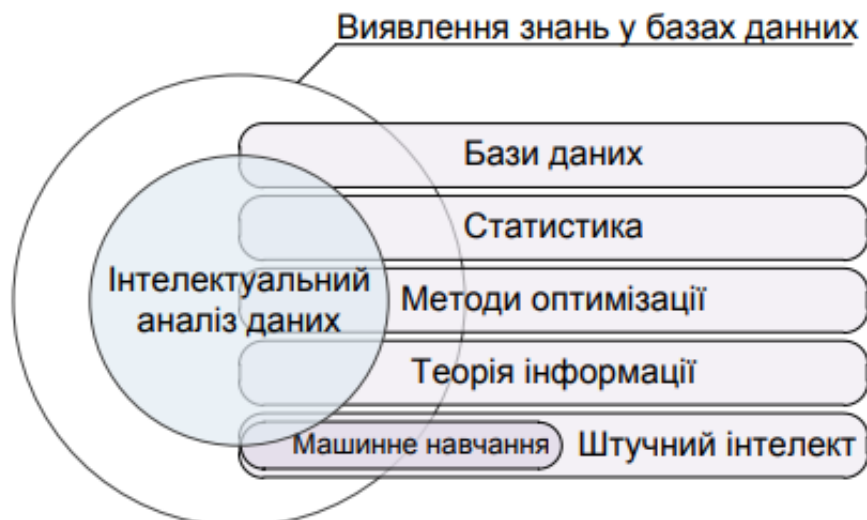


Рисунок 1.1 – Основні складові Data mining

Дані, що використовуються у сфері інтелектуального аналізу, можуть мати різну природу. Вони можуть бути у вигляді графіків, відео, фото, тексту, координат тощо. Організовані за певними правилами та збережені на носії дані формують базу даних.

Інтелектуальний аналіз використовують підприємства, що мають велику кількість активних користувачів. Він допомагає підвищити рівень продажу, краще розуміти аудиторію клієнтів. Розташування продуктів на полицях магазинів та персональна реклама на сайтах є яскравими прикладами реалізації використання інформації, що добута інтелектуальним аналізом історії покупок клієнтів.

Також data mining використовують медичні заклади. Основні задачі – це визначення хвороби людей за симптомами або передбачення росту хворих, що є дуже актуальною задачею під час пандемії COVID-19. [\[5\]](#)

1.2 Задача класифікації

Як ми вже вище визначали класифікація – це надання об’єктам мітки класу, список яких є попередньо визначеним. Короткими словами – прогнозування класів.

Найпростішим видом класифікації є двійкова класифікація, тобто модель може віднести об’єкт до одного з двох заданих класів. Але є різні стратегії, що допомагають перетворити двійковий класифікатор в багатокласовий. Звісно, є методи, що дозволяють відразу створювати модель, здатну на навчання для багатьох класів. Наприклад, класифікатор SGD або наївний баєсів класифікатор.

Але є способи, щоб зробити багатокласовий класифікатор на основі двійкового класифікатора. Одним із таких способів є «Один проти всіх» (OvA). Він полягає в тому, що створює N двійкових класифікаторів, де N – кількість класів, кожен з яких вимірює ймовірність того, що об’єкт належить саме до його класу. Потім обирається той клас, чий класифікатор дав найбільшу ймовірність.

Іншим способом є метод «Один проти одного» (OvO). Його логіка передбачає для кожної пари класів створити двійковий класифікатор. Тобто для N класів створиться $N \times (N-1) / 2$ класів. Після цього обраним класом буде той, який виграв більшість дуелей.

Для деяких алгоритмів стратегія OvO є кращою, адже навчити багато класифікаторів на малих даних швидше, ніж навчити малу кількість класифікаторів на великих даних.^[6]

1.3 Показники ефективності

Основними показниками ефективності є точність і повнота. Для їх визначення введемо такі поняття як:

- TP (true positive) – класифікатор відніс об’єкт до позитивного класу та не помилився.

- FP (false positive) – класифікатор відніс об'єкт до негативного класу та не помилився.
- TN (true negative) – класифікатор відніс об'єкт до позитивного класу, але об'єкт йому не належить.
- FN (false negative) – класифікатор відніс об'єкт до негативного класу, але об'єкт належить позитивному.

FN та TN також називають помилками першого та другого роду відповідно.

На рисунку нижче розглянемо матрицю неточності для класифікатора, що визначає чи об'єкт з набору даних MNIST належить до класу п'ятірок. Зліва зверху вказані приклади істинно негативних класифікацій. Справа зверху хибно позитивні класифікації. Зліва знизу та справа знизу вказані хибно негативні та істинно позитивні класифікації відповідно. [\[7\]](#)

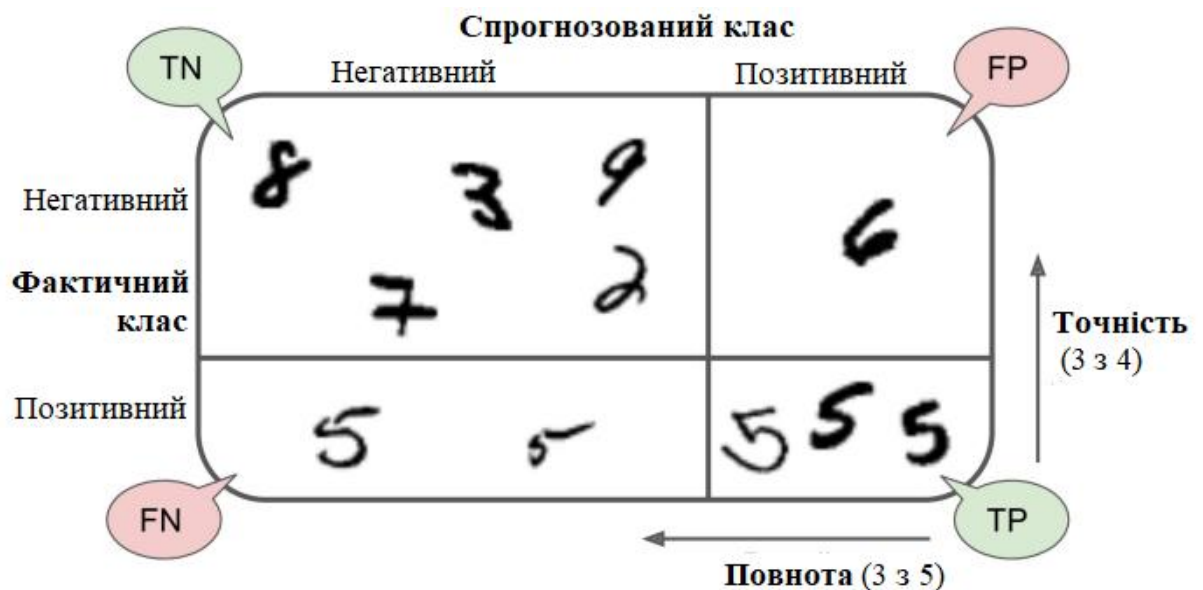


Рисунок 1.2 – Матиця неточності

Точність дорівнює відношенню TP до суми TP та FP.

Повнота дорівнює відношенню TP до суми TP та FN.

В залежності від задачі, класифікатор можуть більше орієнтувати як на точність, так і на повноту. Для прикладу класифікатор, який розроблений для рекомендацій дитячих відеороликів, буде більш орієнтований на точність.

Але для компромісу між цими двома метриками часто використовують так звану міру F_1 .

$$F_1 = \frac{TP}{TP + \frac{FN+FP}{2}}$$

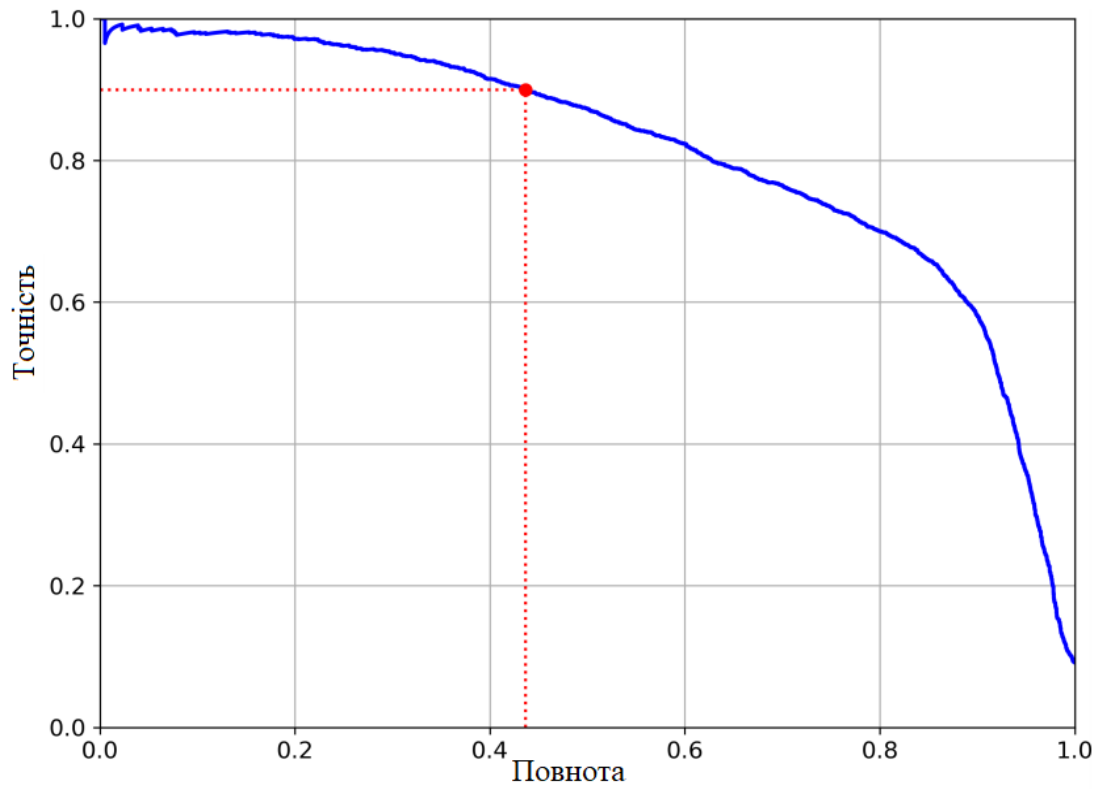


Рисунок 1.3 – Точність в порівнянні з повнотою

РОЗДІЛ 2 НЕЙРОННА МЕРЕЖА ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

2.1 Поняття нейронної мережі

Нейронна мережа – це назва цілого класу методів машинного навчання, в основі яких лежить використання таких одиниць як нейрон, зв'язок та функція активації.

Найпростішим та першим видом була штучна нейронна мережа (artificial neural network - ANN). Розробники запозичили ідею функціонування кори головного мозку та спроектували її у математичну модель. Однією з перших одиниць ANN був штучний нейрон (artificial neuron). Кожен такий нейрон має входи, а сам він дає вихідний сигнал тоді, коли у нього активована певна кількість входів. ^[6]

Розглянемо логічні обчислення за допомогою штучних нейронів. Кожен з них активується тоді, коли у нього більш ніж два активних входи.

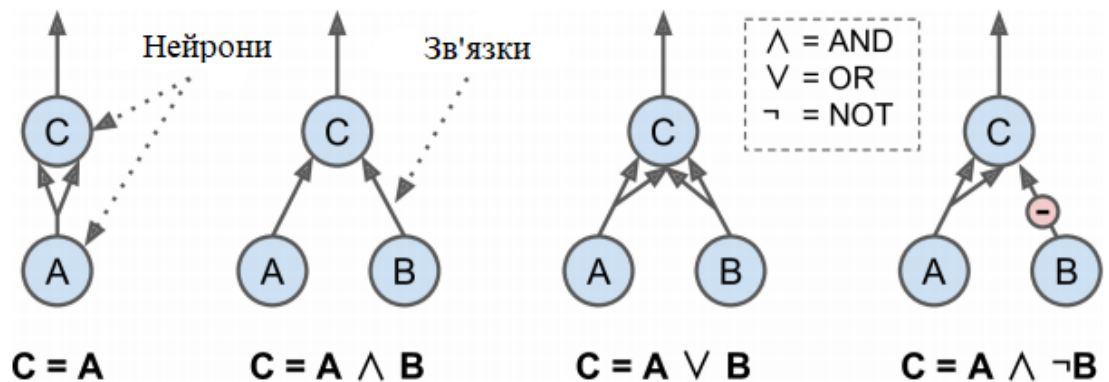


Рисунок 2.1 – Мережі ANN для простих логічних обчислень

Перша мережа – це мережа тотожності. Нейрон C активується тоді, коли нейрон A є активним.

Друга мережа – це логічне так (and). Активація нейрона C відбувається тільки за умови активності A та B.

Третя мережа – це логічне або (or). Активація нейрона C відбувається при активності хоча б одного з вхідних нейронів A та B.

Четверта мережа – це більш складна система, що активує нейрон С у випадку активності А і водночас неактивності нейрона В.

Ще одним з простих архітектур ANN є перцептрон, також його називають пороговим логічним елементом. Відрізняється від свого попередника тим, що після отримання деяких значень на входах кожен перцептрон обчислює зважену суму за формулою:

$$z = w^T x,$$

де w – вектор вагів;

x – вектор вхідних сигналів.

Після цього зважена сума стає параметром функції активації, в залежності від результату якої перцептрон вирішує, який сигнал йому передавати. Найпопулярнішою функцією активації для таких нейронних мереж є функція Хевісайда:

$$h(z) = \begin{cases} 0, & \text{якщо } z < 0 \\ 1, & \text{якщо } z \geq 0 \end{cases}$$

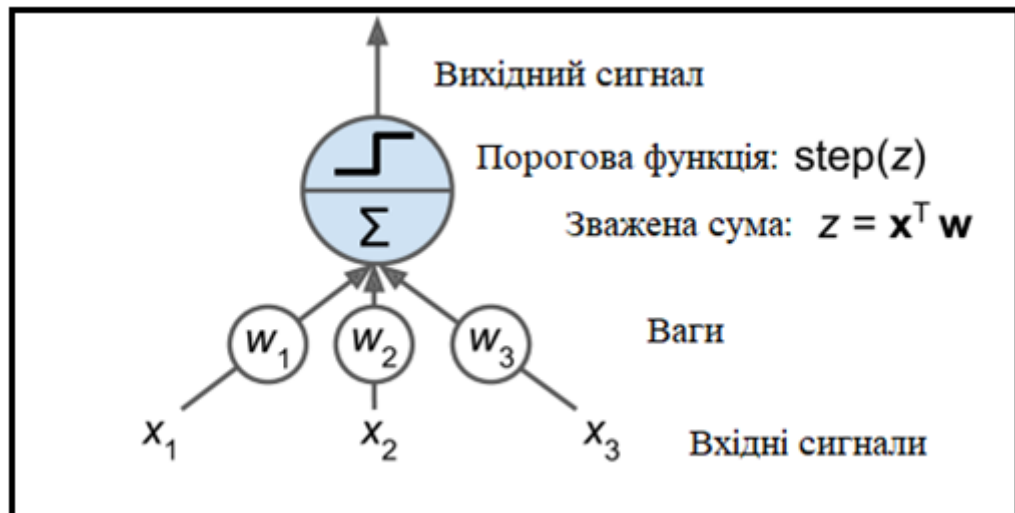


Рисунок 2.2 – Пороговий логічний елемент

Коли всі нейрони зв'язані між собою, така нейронна мережа має назву повно зв'язна (fully connected). Також важливо зазначити, що до шару нейронної мережі додається так званий нейрон зміщення, що завжди рівний 1. Входи перцептрона мають назву – вхідні нейрони. Їхня сукупність формує вхідний шар. Останнім шаром перцептрона є сукупність вихідних нейронів,

що формують вихідний шар. Якщо є шари нейронів, що знаходяться між вхідним та вихідним шаром, вони мають назву приховані.

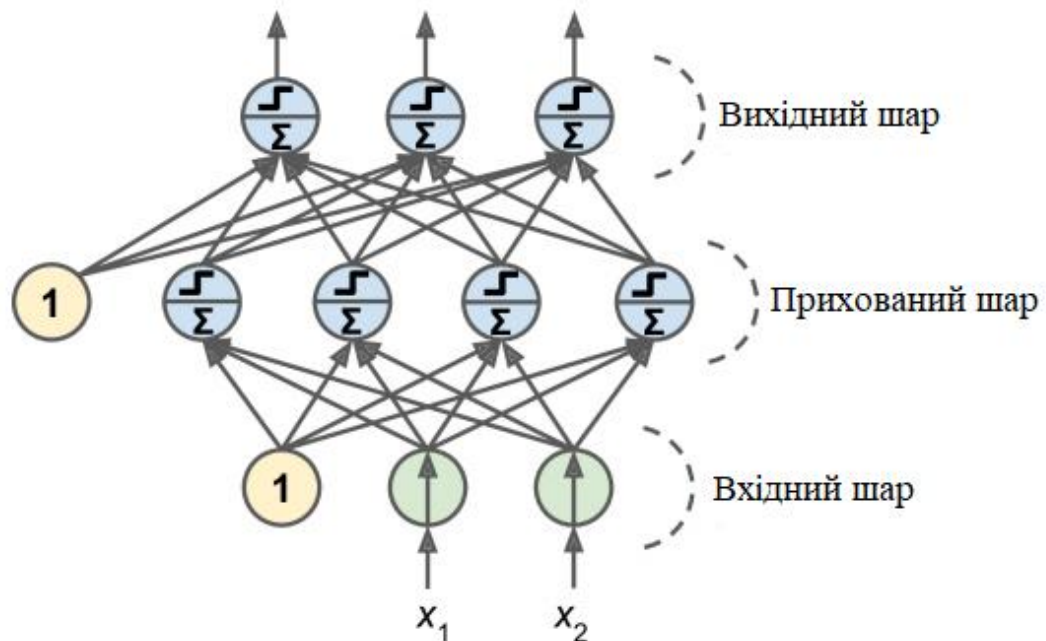


Рисунок 2.3 – Архітектура багатошарового перцептрона

Функція активації – це залежність вихідного сигналу штучного нейрона від його вхідних сигналів. Розглянемо найпоширеніші з них.

Ступінчата функція (Step):

$$f(x) = \begin{cases} -1, & \text{якщо } x < 0 \\ 1, & \text{якщо } x \geq 0 \end{cases}$$

Логістична функція (Sigmoid):

$$f(x) = \frac{1}{1 + e^{-x}},$$

розробники внесли цю функцію, бо вона є диференційованою в кожній точці, що дає змогу реалізувати градієнтний спуск для навчання нейронної мережі.

Випрямлений лінійний вузол (ReLU):

$$f(x) = \max(0, x),$$

ця функція, як можемо бачити на рисунку 2.4, не є диференційованою в кожній точці, проте на практиці вона працює дуже добре та має високу

швидкість обчислення, що зробило її стандартним варіантом функції активації.

Гіперболічний тангенс (Tanh):

$$f(x) = \frac{e^x + e^{-x}}{e^x - e^{-x}}$$

ця функція, як і логістична, має S-подібну форму, але її вихідні значення знаходяться в діапазоні від мінус 1 до 1, що робить вихід кожного шару більш зосередженим навколо 0 під час навчання. Це робить процес навчання швидшим.

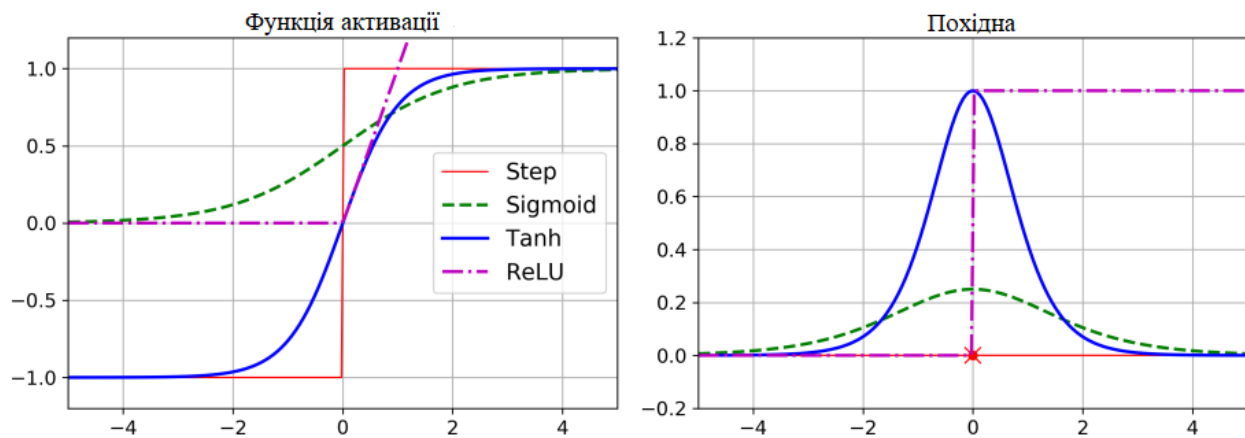


Рисунок 2.4 – Функції активації та їх похідні.

2.2 Градієнтний спуск та метод зворотного поширення помилки

Градієнтний спуск являє собою алгоритм оптимізації, що здатний знаходити оптимальні рішення дуже великого діапазону задач. Основна ідея градієнтного спуску в тому, щоб покроково змінювати вхідний параметр функції похибки таким чином, щоб її значення досягло мінімуму. Першим кроком є випадкова ініціалізація вхідного вектора, потім поступово ми покращуємо його, змінюючи значення в сторону антиградієнта. [\[8\]](#)

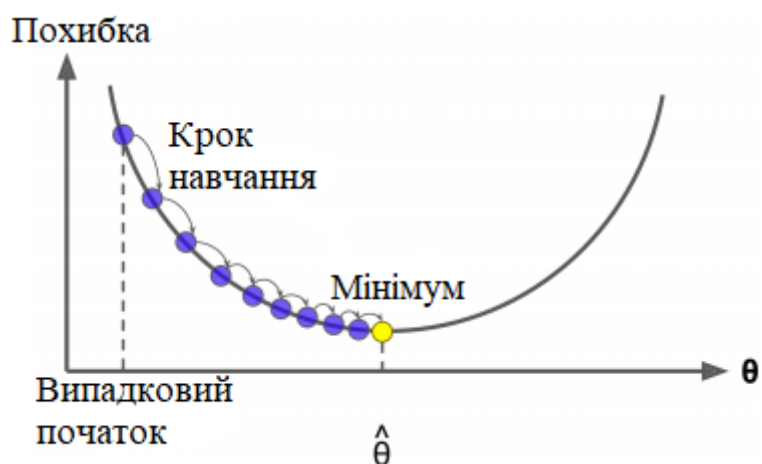


Рисунок 2.5 – Ілюстрація градієнтного спуску

Дуже важливим параметром градієнтного спуску є розмір кроку, який називають швидкістю навчання. Якщо швидкість навчання дуже мала, то алгоритму прийдеться пройти дуже багато ітерацій, що потребує багато часу.

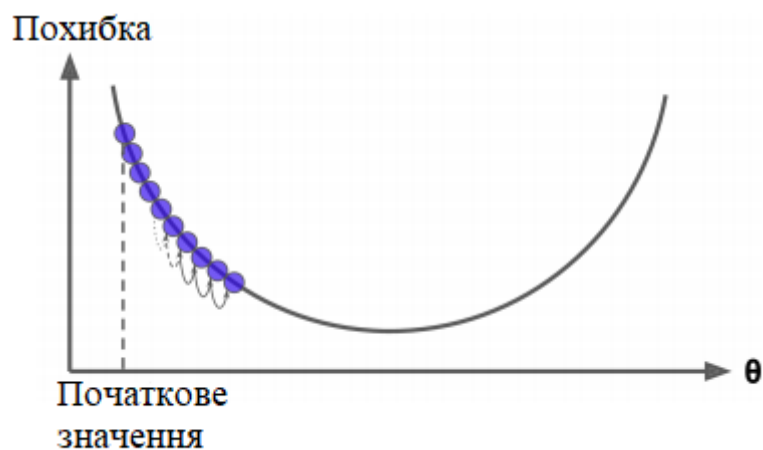


Рисунок 2.6 – Градієнтний спуск при низькій швидкості навчання

З іншої сторони, якщо швидкість навчання буде занадто високою, тоді можна перескочити значення мінімуму й алгоритм не зійдеться.

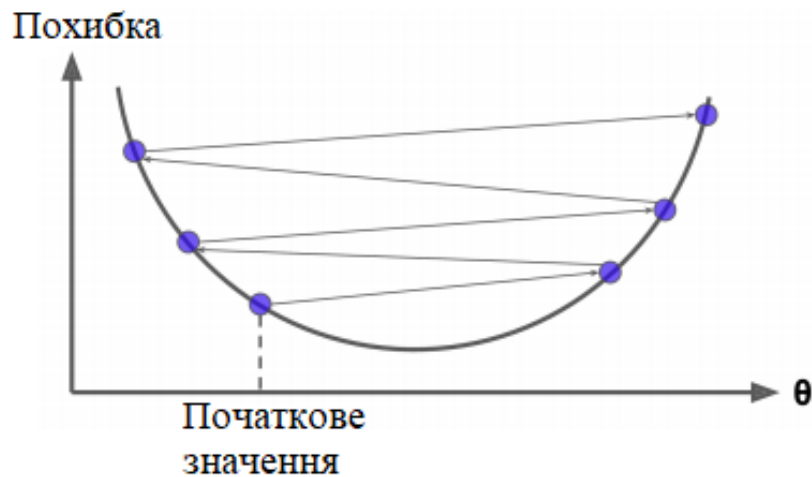


Рисунок 2.7 – Градієнтний спуск при великій швидкості навчання

Крок градієнтного спуску вираховується як:

$$\theta^{i+1} = \theta^i - \eta \nabla_{\theta} MSE(\theta^i),$$

де θ – вхідний параметр;

η – швидкість навчання;

$\nabla_{\theta} MSE(\theta^i)$ – вектор градієнт в поточній точці.

Алгоритм зворотного поширення помилки – це алгоритм, що базується на методі градієнтного спуску. Він вираховує градієнт помилки мережі відносно кожного параметра моделі. ^[9] Розглянемо більш детально:

- Він обробляє по одному мініпакету за раз, кожен з яких складається наприклад з 32 зразків, і багаторазово проходить по повному навчальному набору. Кожен прохід називається епохою.
- Кожен мініпакет передається вхідному шару мережі, який відправляє наступному прихованого шару, потім алгоритм обчислює вихід всіх нейронів в цьому шарі, і результат передається наступному. Процес продовжується до тих пір, поки не буде отриманий вихід останнього шару. Це називається прямий прохід, він схожий на вироблення прогнозу, за винятком того, що всі проміжні результати зберігаються, оскільки вони будуть необхідні для зворотного проходу.
- Потім алгоритм вимірює вихідну похибку мережі.

- Наступним кроком, він обчислює розмір вкладу в помилку кожного вихідного зв'язку. Це робиться аналітично, за допомогою використання ланцюгового правила, що робить даний крок швидким і точним.
- Потім алгоритм вимірює, скільки таких вкладів в помилку надходить від кожного зв'язку попереднього шару, потім знову використовує ланцюгове правило і працює у зворотному напрямку до тих пір, доки не потрапить до вхідного шару. Такий зворотний прохід ефективно вимірює градієнт похибки по всім вагам мережі.
- Нарешті, алгоритм виконує крок градієнтного спуску для побудови всіх параметрів у мережі, з урахуванням тільки що вихованих градієнтних помилок.

2.3 Згорткова нейронна мережа

Згорткові нейронні мережі (convolutional neural network – CNN) з'явилися в результаті вивчення зорової кори головного мозку і використовувались при розпізнаванні зображень, починаючи з 1980-х років.^[10] Ідея архітектури цих мереж була запозичена з принципів розпізнавання біологічного мозку, а саме те, що деякі нейрони краще розпізнають зображення горизонтальних ліній, деякі – вертикальних. Також було помічено, що інші нейрони можуть розпізнавати більш спутані образи. Це вказало на думку, що входи цих нейронів є виходами примітивних нейронів. Така архітектура спроможна розрізняти всі види образів у будь-якій області поля зору.

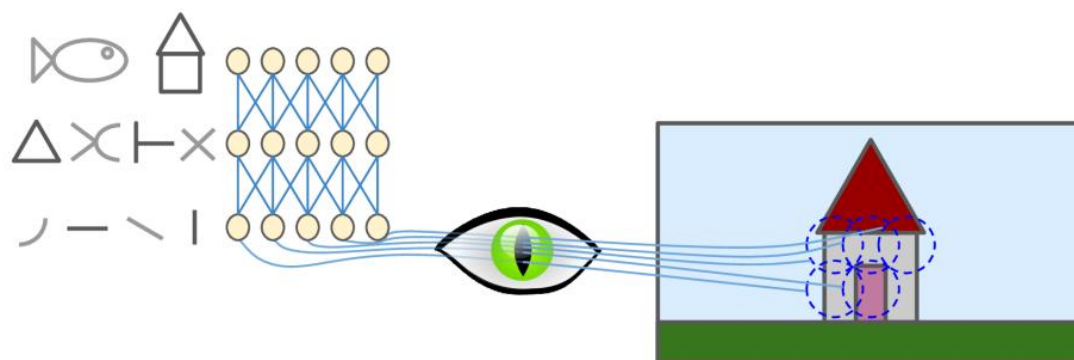


Рисунок 2.8 – Принцип передачі сигналів у зоровій корі головного мозку для пошуку образів

Нейрони в першому зародковому шарі зв'язані не з кожним одиничним пікселем у вхідному зображенні, а тільки з пікселями у власних рецепторних полях. Зі свого боку кожен нейрон у другому шарі зв'язаний тільки з нейронами, що знаходиться всередині невеликого прямокутника в першому шарі. Така архітектура дозволяє мережі зосередитись на маленьких низькорівневих признаках в першому прихованому шарі, потім скомбінувати їх в більші високорівневі признаки в наступному прихованому шарі і так далі. Ієрархічна структура такого роду розповсюджена в реальних зображеннях, що є однією з причин, чому мережі CNN наскільки гарно працюють при розпізнаванні зображень.

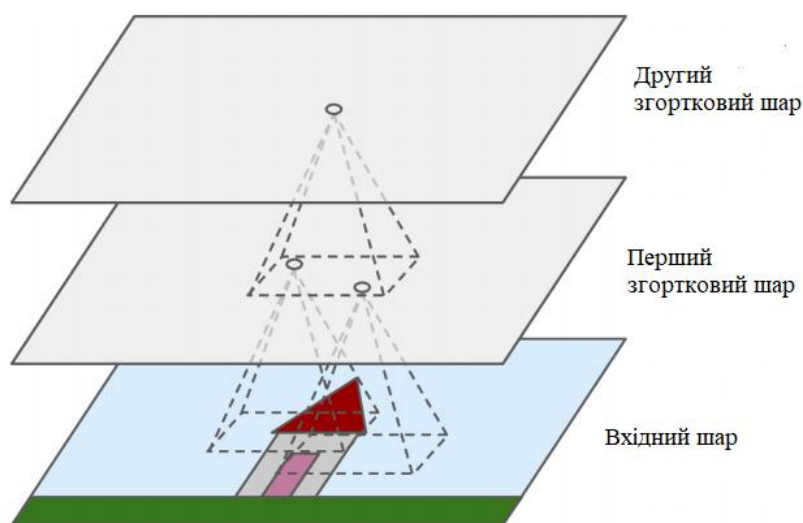


Рисунок 2.9 – Шари CNN з прямокутними локальними рецепторними полями

Для збереження розмірності шару, навколо входів додають нулі, як зображено на рисунку нижче, де f_h і f_w – висота і ширина рецепторного шару відповідно. Це називається доповнення нулями.

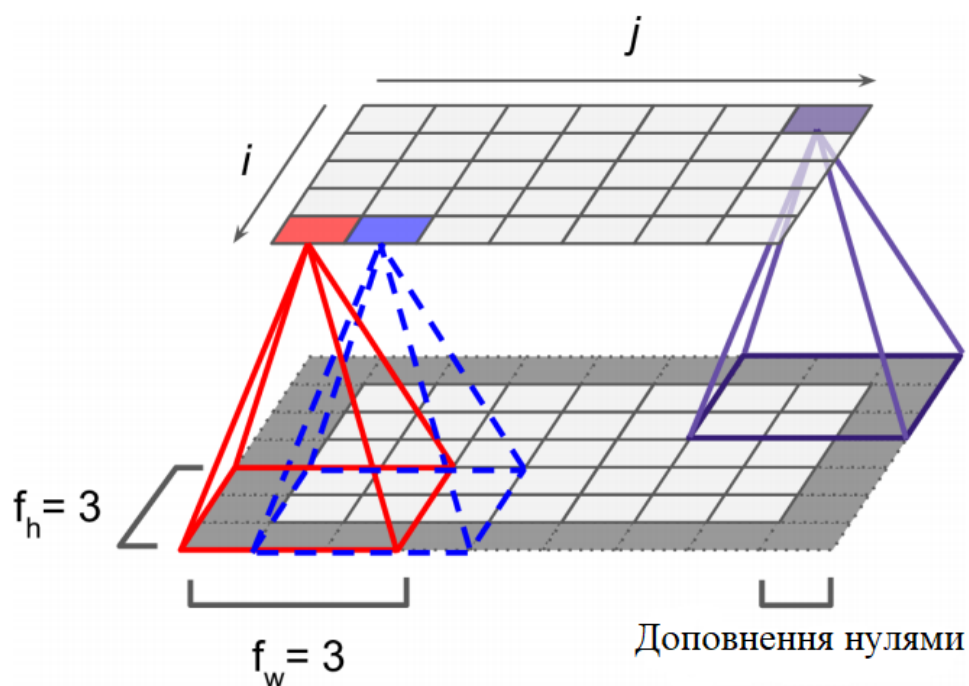


Рисунок 2.10 – Зв'язки між шарами та доповнення нулями

Якщо вхідний шар має велику розмірність, то для збільшення швидкості обробки зображення використовують страйд: це зміщення від одного рецепторного шару до наступного.

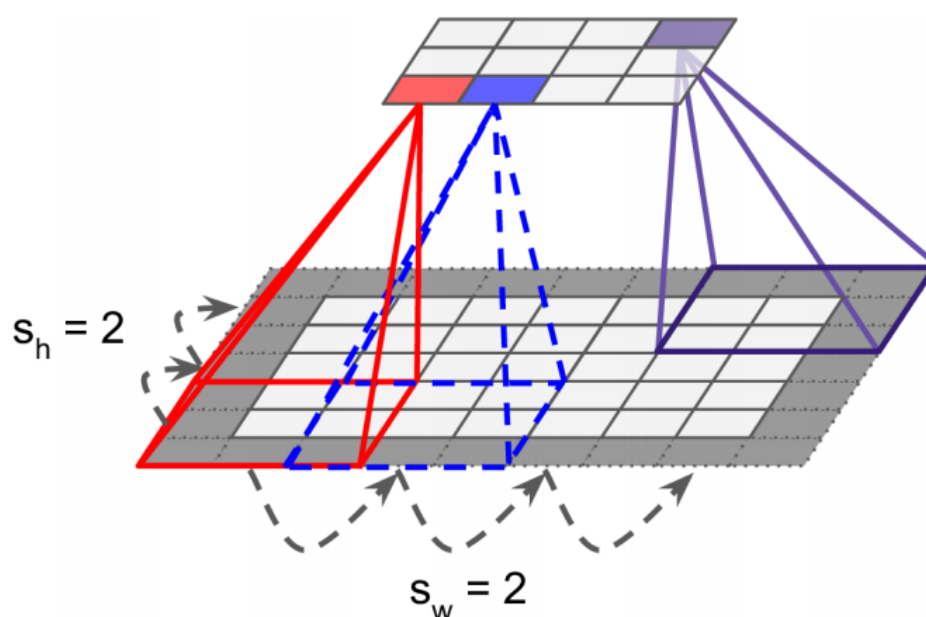


Рисунок 2.11 – Пониження розмірності з використанням страйда

Для пошуку більш примітивних частин зображення використовують фільтри. Це спеціальні набори вагів, що представлені у вигляді невеликого зображення розміром рецепторного поля. Вхідне зображення, що пройшло через фільтр, називається картою ознак.

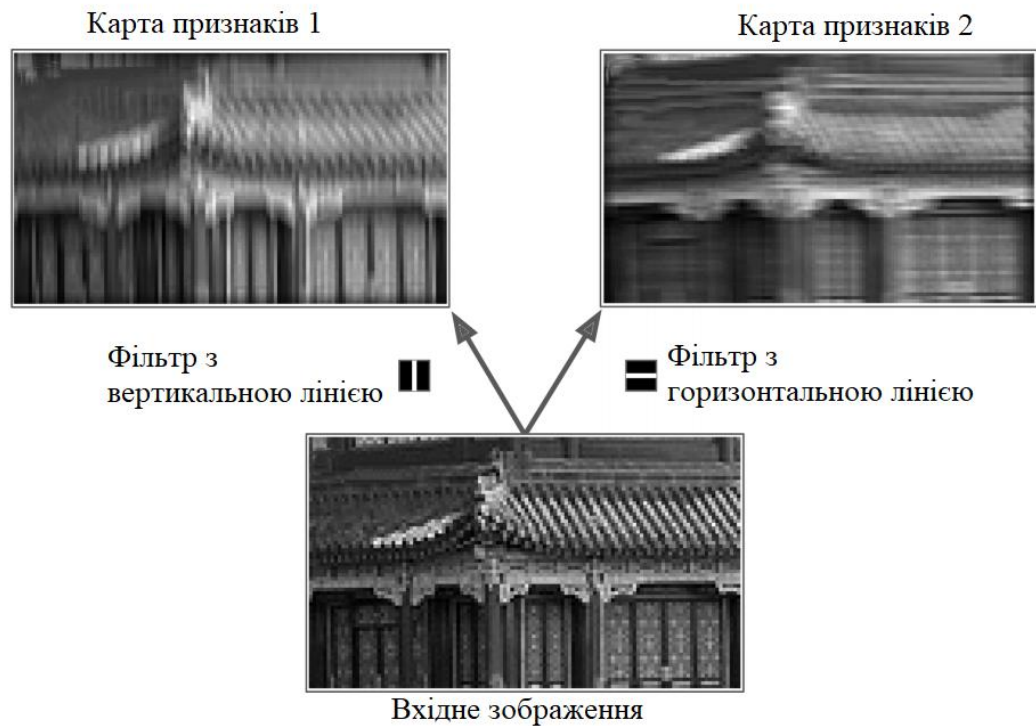


Рисунок 2.11 – Отримання карт ознак завдяки різним фільтрам

Згортковий шар може мати довільну кількість карт ознак. Це дозволяє мережі розпізнавати різні ознаки по всьому зображенню. Також зображення складаються з великої кількості підшарів. На один кольоровий канал один підшар. Зазвичай, це червоний, синій та зелений. Але може бути і більше: наприклад, при фіксації інфрачервоного чи ультрафіолетового випромінювання, що фіксують спеціальні прилади. Нижче зображено шаблон архітектури, що працює по принципах зорової кори мозку. Спочатку мережа робить акцент на примітивних об'єктах, потім поєднує їх для розпізнавання комбінацій, а потім знаходить конкретний патерн.

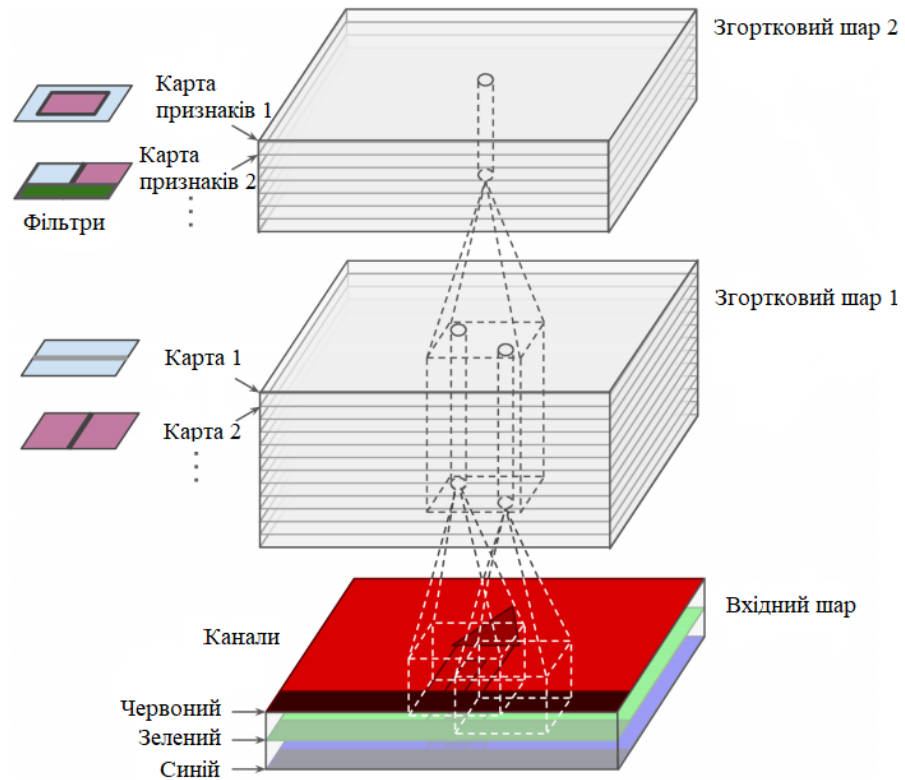


Рисунок 2.12 – Згорткові шари з картами ознак і кольорових каналів

Обчислення виходу нейрона в згортковому шарі:

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_{n'}-1} x_{i',j',k'} \times W_{u,v,k',k} ,$$

де

- $\begin{cases} i' = i \times s_h + u \\ j' = i \times s_w + u \end{cases}$;
- $z_{i,j,k}$ – вихід нейрона, що розташований в рядку i та стовпці j в карті ознак згорткового шару номер k ;
- s_h, s_w – вертикальний та горизонтальний страйд;
- $f_h, f_w, f_{n'}$ – висота і ширина рецепторного шару, кількість карт ознак попереднього шару відповідно;
- $x_{i',j',k'}$ – вихід нейрона для рядка i' , стовпця j' , карти ознак k' ;
- b_k – член зміщення для карти ознак k ;
- $W_{u,v,k',k}$ – вага зв'язків між штучними нейронами для рядка u та стовпця v .

Шари об'єднання або шари підвибірки – це спеціальні структури, що дозволяють прорідити вхідне зображення для зменшення обчислювального навантаження, використання пам'яті та кількості параметрів. Але замість набору вагів, як в попередньому випадку, тут застосовують агрегатні функції, такі як середнє, або максимум.

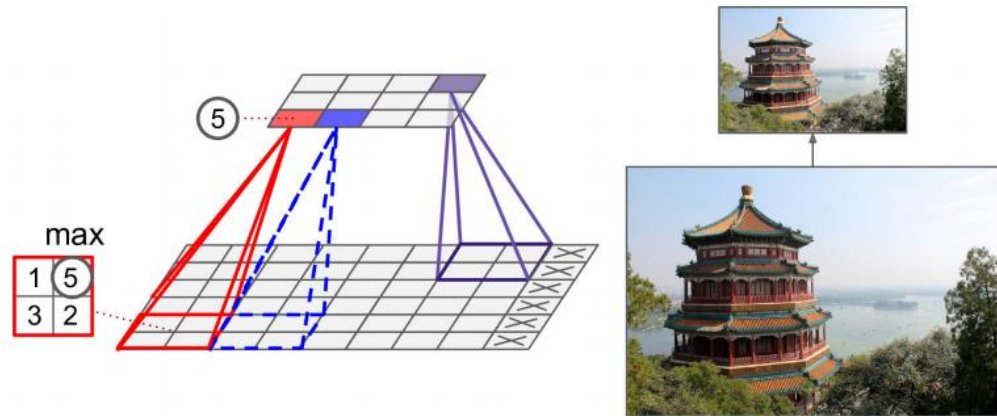


Рисунок 2.13 – Шар об'єднання по максимуму

Окрім зменшення об'єму обчислень та зменшення кількості параметрів, об'єднання по максимуму вносить деяку степінь інваріантності. Ми вважаємо, що яскраві пікселі мають значення нижче, ніж темні. На рисунку 2.14 розглянуто три зображення, з яких можемо бачити, що для зображення А та В вихід однаковий: це має назву трансляційна інваріантність. Для зображення С вихід відрізняється зміщенням пікселів вправо, але інваріантність на 75% досі присутня. Вставляючи шар об'єднання по максимуму через кожні декілька шарів мережі CNN, можна отримати деяку степінь трансляційної інваріантності у великому масштабі.

Окрім того, об'єднання по максимуму пропонує невелику величину ротаційної інваріантності та легку масштабну інваріантність. Вона може бути корисною у випадках, коли прогноз не має залежати від цих деталей, як в задачах класифікації.

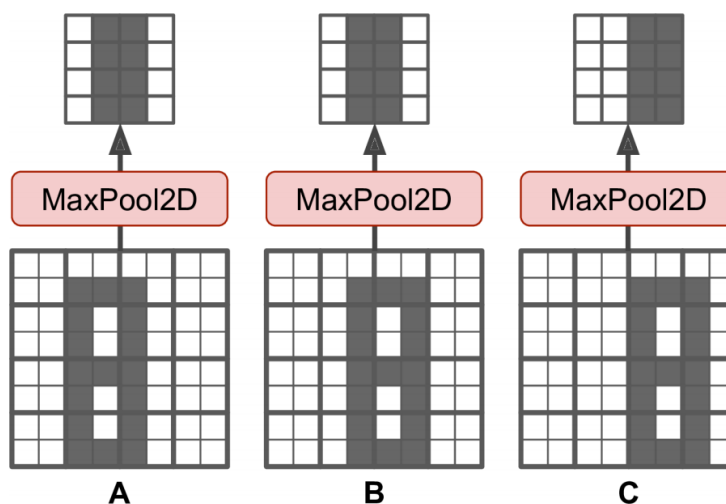


Рисунок 2.14 – Привнесення інваріантності у невеликі трансляції

Недоліками такого об'єднання є втрата великої кількості вхідних значень. Логічно було б вважати, що агрегація по середньому принесе менше втрат, але, як показує практика, вона робить менший акцент на важливих пікселях і не зберігає основні ознаки зображення.

Доповнення даних – це ще один з методів, що дозволяє розширити набір для навчання, якщо кількість об'єктів замала. Це зменшує ризик перенавчання, тому можна вважати, що це також свого роду регуляризація. Правильне доповнення передбачає, що людина не здатна точно сказати чи доповнені ці дані, чи ні. Для прикладу фотографії, що має класифікувати майбутня модель, можна повернути, збільшити чи зменшити розширення, змінювати рівень контрастності тощо.

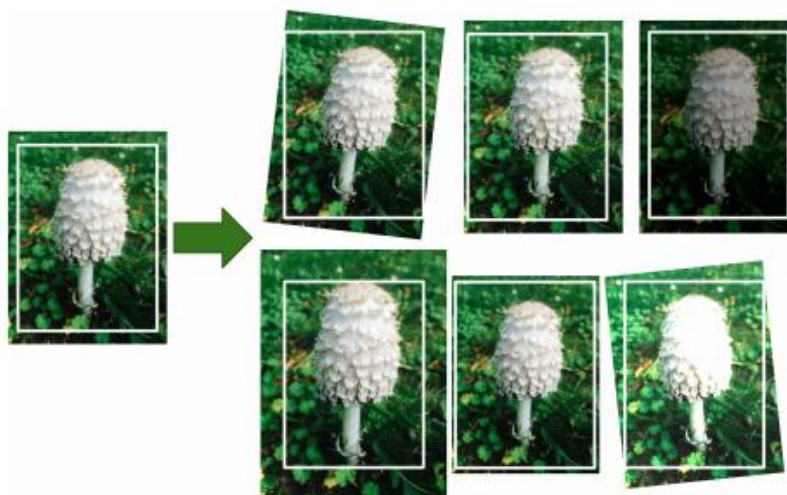


Рисунок 2.15 – Генерування нових навчальних об'єктів

2.4 Архітектура Xception

Це один із варіантів архітектури, що розробила компанія Google. В буквальному сенсі переводиться як екстремальний початок. Вона використовує шари спеціального типу, що мають назву сепарабельний шар згортки в глибину або просто сепарабельний згортковий шар. [\[11\]](#)

Звичайні згорткові шари намагаються одночасно захопити просторові образи (наприклад, овал) і міжканальні образи (наприклад ніс + рот + очі = обличчя), а сепарабельний згортковий шар робить припущення про те, що просторові і міжканальні образи можуть моделюватись окремо. Таким чином, він складається з двох частин: перша з яких застосовує фільтр для кожної вхідної карти ознак, а друга частина шукає виключно міжканальні образи, а це звичайний згортковий шар.

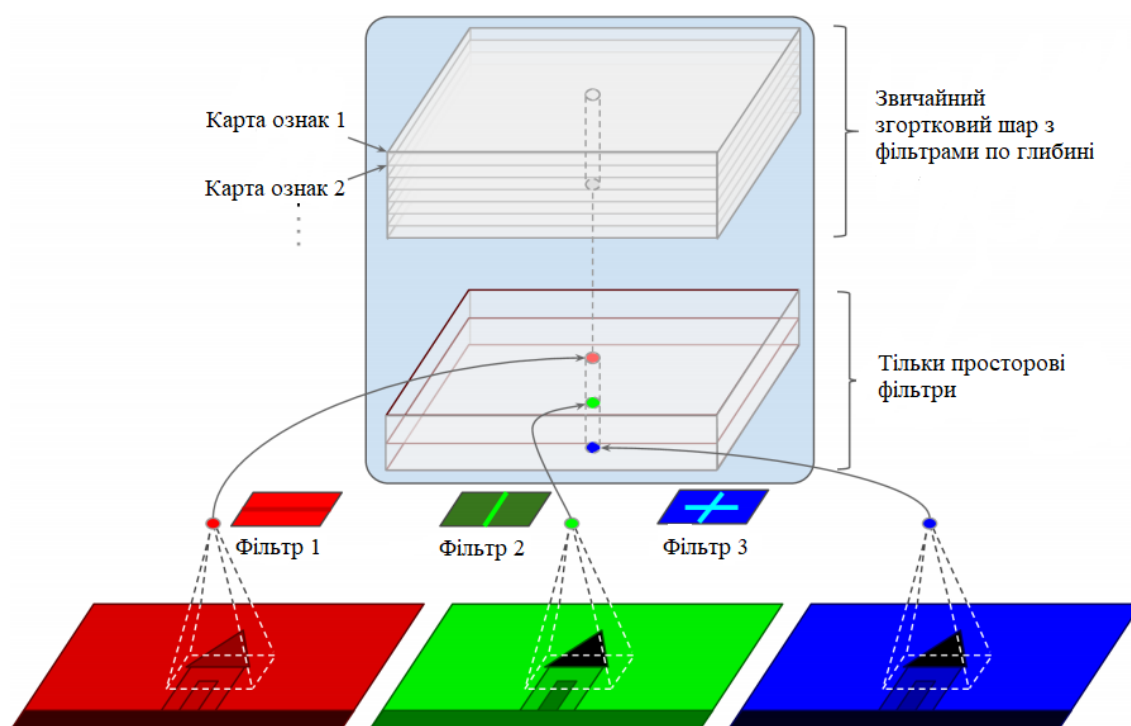


Рисунок 2.16 – Сепарабельний шар згортки в глибину

РОЗДІЛ 3 КЛАСИФІКАЦІЯ РЕНТГЕНІВСЬКИХ ЗНІМКІВ

3.1 Підготовка та аналіз вхідних даних

Об'єктом класифікації будуть рентгенівські знімки легень здорових людей та тих, що хворі на пневмонію. [12]

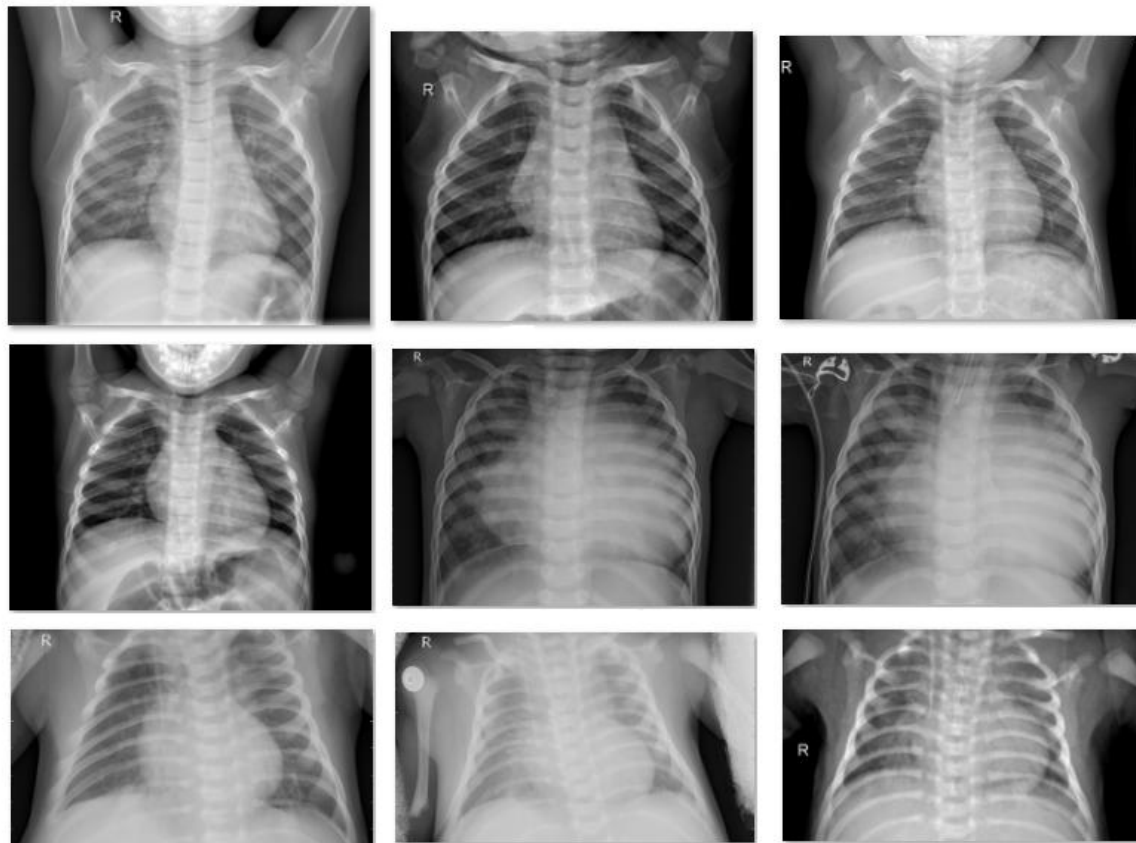


Рисунок 3.1 – Приклад вхідних даних

Завантажений набір даних уже розбитий на три набори: для тренування, валідації та тестування. Нижче буде проведено його аналіз.

На рисунку 3.2 зображено розподіл розмірів зображень за допомогою функції `plot()`, а нижче, на рисунку 3.3, значення їх середніх.

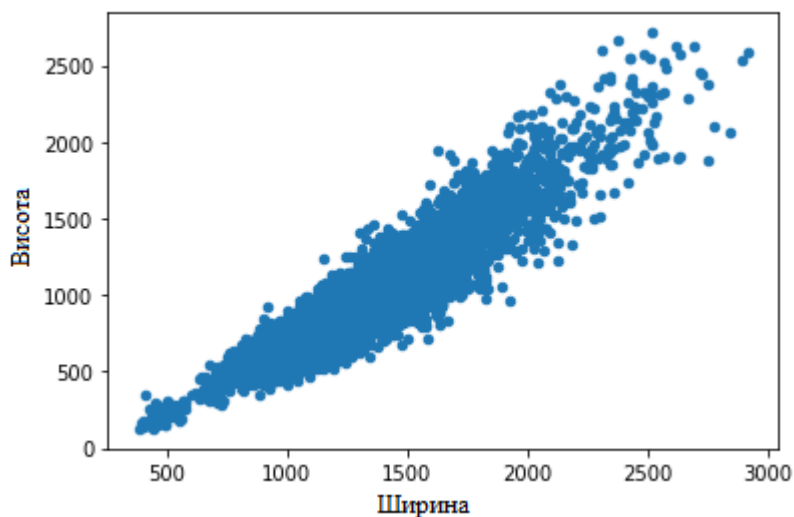


Рисунок 3.3 – Розподіл розмірів вхідних зображень

```
df.loc[:, ['width', 'height']].mean()
```

```
width    1327.880806
height   970.689037
dtype: float64
```

```
df.loc[:, ['width', 'height']].median()
```

```
width    1281.0
height   888.0
dtype: float64
```

Рисунок 3.4 – Значення середніх розмірів зображень

Детальніше про розбиття вхідного набору. Набір має складатись з трьох частин: для тренування, для валідації (перевірки) та тестовий. Перший набір має мати найбільший розмір, щоб врахувати якнайбільшу кількість особливостей. Набір валідації створений для перевірки моделі та підбору її параметрів. Тестовий набір вимірює фінальну точність моделі. Об'єкти з нього не попадають раніше фінального етапу.

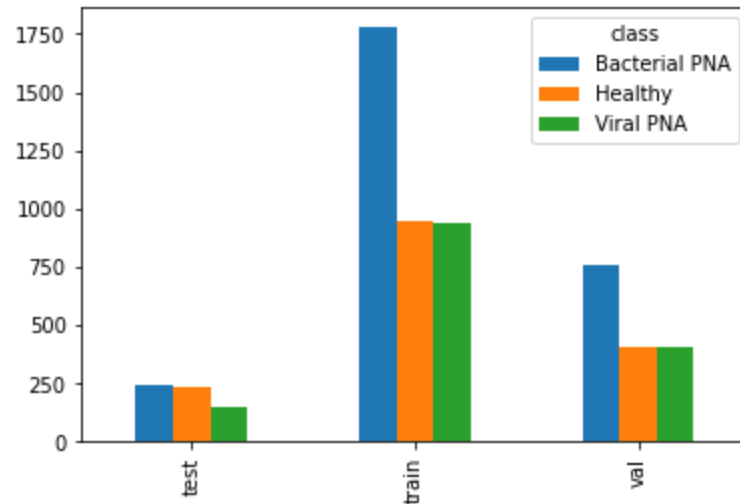


Рисунок 3.5 – Гістограми розподілу класів по наборах

Для ілюстрації об'єктів, необхідно написати метод, що виведе по декілька зразків з підписами міток. Для прикладу використаємо тренувальний набір.

```
def plot_image_grid(files, title, color_mode=COLOR_MODE,
                    interpolation=INTERPOLATION):

    square = math.ceil(math.sqrt(len(files)))

    fig, axes = plt.subplots(nrows=square, ncols=square, figsize=(10, 10))

    fig.suptitle(title, fontsize="xx-large")
    axes = axes.flatten()

    for (title, filename), ax in zip(files, axes):
        ax.set_title(f"{title}", fontsize="x-large")
        ax.imshow(
            preprocessing.image.load_img(filename, color_mode=color_mode),
            interpolation=interpolation,
        )

    for ax in axes:
        ax.axis("off")

    plt.tight_layout(pad=4)
    plt.show()
```

```

def get_sample_files(df, stage, target, n, random_state):
    return list(
        df.loc[(df["stage"] == stage) & (df["class"] == target),
              ["class", "filename"]]
            .sample(n=n, random_state=random_state)
            .itertuples(name=None, index=False)
    )

def plot_image_grid(
    get_sample_files(df, STAGE_TRAIN, CLASS_VPNA, 3, rs)
    + get_sample_files(df, STAGE_TRAIN, CLASS_H, 3, rs)
    + get_sample_files(df, STAGE_TRAIN, CLASS_BPNA, 3, rs),
    "3 Sets Of Random Training Samples",
)

```

3 Sets Of Random Training Samples

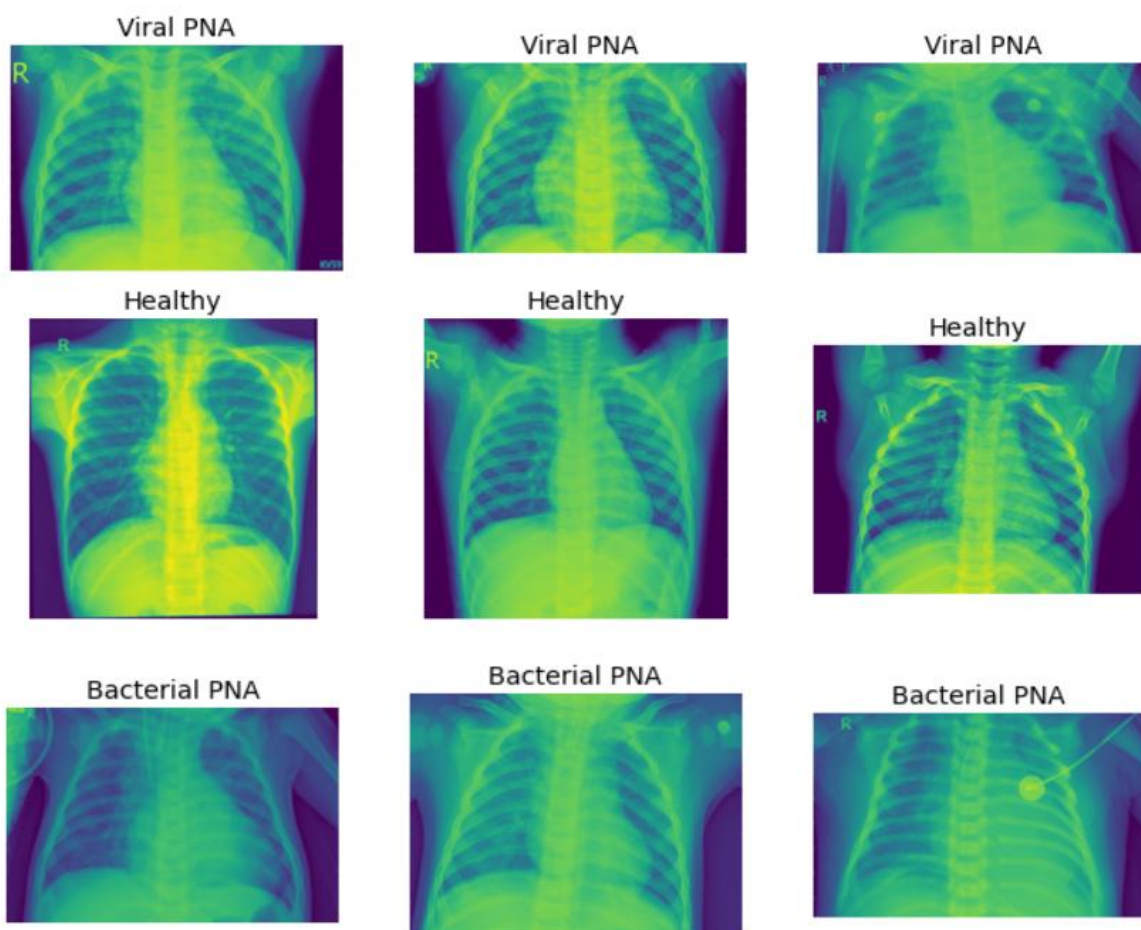


Рисунок 3.6 – Вхідні дані з мітками класів

3.2 Реалізація моделі

Архітектура моделі буде розроблена на основі Xception, але з деякими змінами. Xception вводиться без верхніх шарів та ініціалізується за допомогою зважених ваг мережі ImageNet. ^[13] Вихідні дані мають відтінки сірого, але оскільки Xception вимагає трьох каналів, для створення вхідних функцій застосовується один шар згортки з регуляризацією. Форма за замовчуванням для Xception (299, 299, 3), тому входи були відкалібровані, щоб бути кратним цим розмірам. Коефіцієнт масштабування та максимальний шар об'єднання використовуються для зменшення зображень в цю форму.

Побудована модель на основі Xception отримала найвищу продуктивність з цією архітектурою при застосуванні до рентгенівських даних. Ця модель була застосована до зображень у градаціях сірого, тому необхідні 3 канали створюються із вхідного рівня згортки. Ініціалізація за допомогою ImageNet є доцільною для класифікації медичних зображень та адаптується до вхідних даних з градаціями сірого. Частина задніх шарів розморозується для налаштування. Оскільки доступні навчальні параметри швидко зростають, то розморозження відбувається у невеликій кількості.

Нижче описана функція, що створює вищеописану модель.

```
def construct_model():

    x = input = layers.Input(shape=IMAGE_SHAPE)

    x = layers.Conv2D(
        3, SCALE_DIM, padding="same", kernel_regularizer="L2", activation="relu"
    )(x)

    if SCALE_DIM > 1:
        x = layers.MaxPooling2D(SCALE_DIM)(x)

    Xception = applications.Xception(
        weights="imagenet", include_top=False,
        input_shape=(XCEPT_DIM, XCEPT_DIM, 3)
    )

    freeze = round(FREEZE_LAYERS_FRAC * len(Xception.layers))
    for i, layer in enumerate(Xception.layers, start=1):
        layer.trainable = i > freeze
```

```

if VERBOSE:
    trainable_flags = [layer.trainable for layer in Xception.layers]
    print(trainable_flags)
    print(sum(trainable_flags) / len(Xception.layers))

x = Xception(x)
x = layers.GlobalMaxPool2D()(x)
x = output = layers.Dense(3, activation="softmax")(x)

model = models.Model(input, output)

return model

```

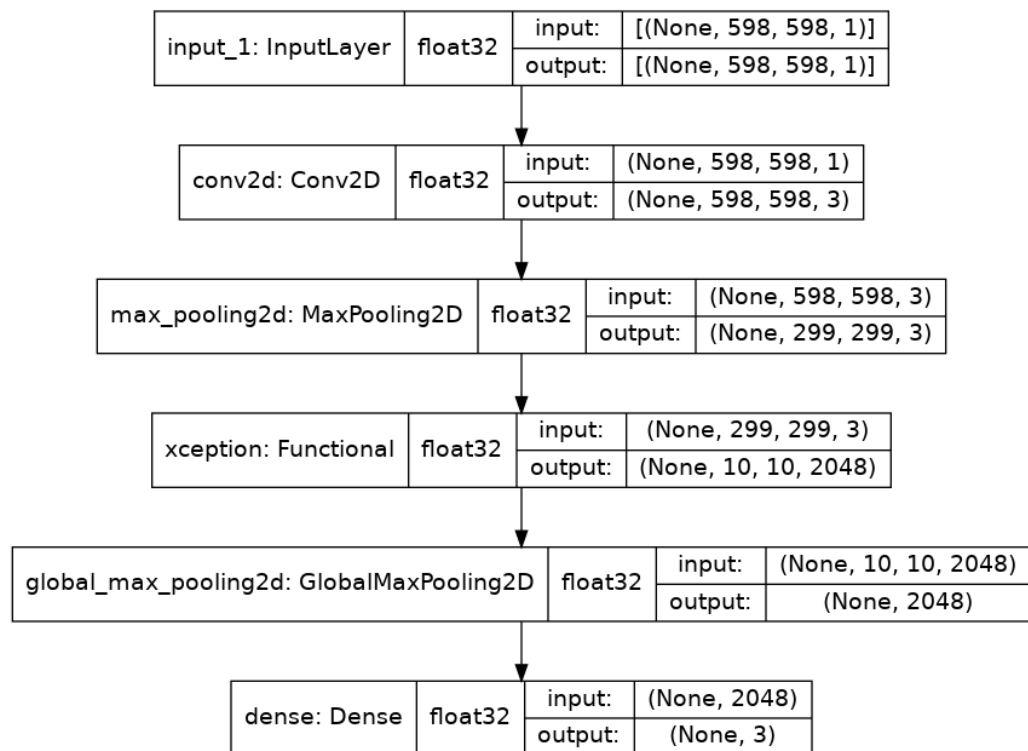


Рисунок 3.7 – Візуалізація порядку шарів описаної нейронної мережі

Розберемо кожен шар окремо:

- InputLayer – вхідний шар;
- Conv2D – згортковий шар з трьома каналами;
- MaxPooling2D – шар об'єднання по максимуму;
- Xception – сепарабельний згортковий шар;
- GlobalMaxPooling2D – шар об'єднання по глобальному максимуму;
- Dense – повно зв'язний шар з трьома виходами;

3.3 Навчання та тестування нейронної мережі

Нижче описано, як саме підібрані параметри: кількість епох, швидкість навчання. Також вказані умови закінчення навчання.

```
def fit_model(model, train_iterator, validation_iterator):

    def lr_scheduler(epoch, lr):
        DROP = 0.94
        EPOCHS_DROP = 2
        rate = INIT_LEARN_RATE *
            math.pow(DROP, math.floor((1 + epoch) / EPOCHS_DROP))
        return rate

    schedule_lr = callbacks.LearningRateScheduler(lr_scheduler)

    stop_early = callbacks.EarlyStopping(
        patience=PATIENCE, monitor="val_accuracy", restore_best_weights=True
    )

    weights = class_weight.compute_class_weight(
        "balanced",
        list(train_iterator.class_indices.values()), train_iterator.classes
    )

    weight_dict = {i: weights[i] for i in train_iterator.class_indices.values()}

    if VERBOSE:
        print("Loss weights:", weight_dict)
    return model.fit(
        train_iterator,
        validation_data=validation_iterator,
        class_weight=weight_dict,
        callbacks=[schedule_lr, stop_early],
        epochs=EPOCHS,
        verbose=VERBOSE,
    )
```

Аналізом моделі є представлення точності та похибки, матриці невідповідностей, вибірки неправильно передбачених зображень та графік неправильно передбачених розмірів зображень. [\[14\]](#)

Розглянемо значення похибки та точності моделі під час кожної епохи.

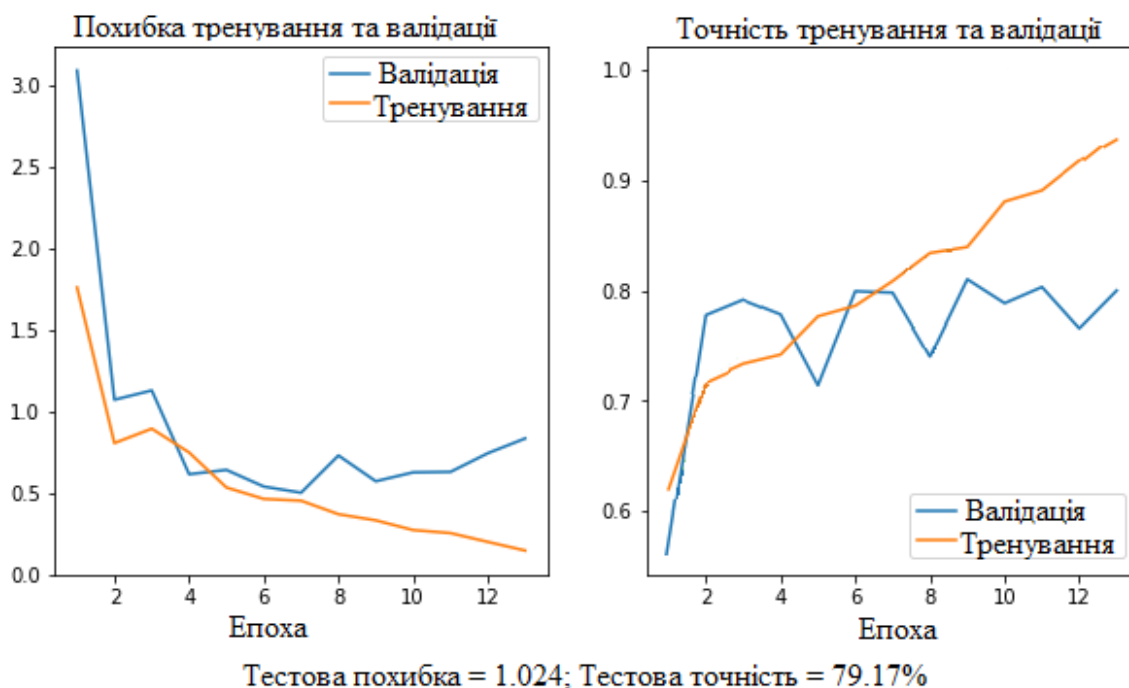


Рисунок 3.8 – Показники моделі під час навчання

У матриці невідповідностей на рисунку 3.9 видно, що бактеріальна пневмонія в більшості випадків точно класифікується. Вірусну пневмонію можна часто розпізнати, але її також плутають з бактеріальною пневмонією. Здорові зразки важче розрізнити, і помилкові класифікації часто розподіляються рівномірно між двома класами пневмонії. [15]

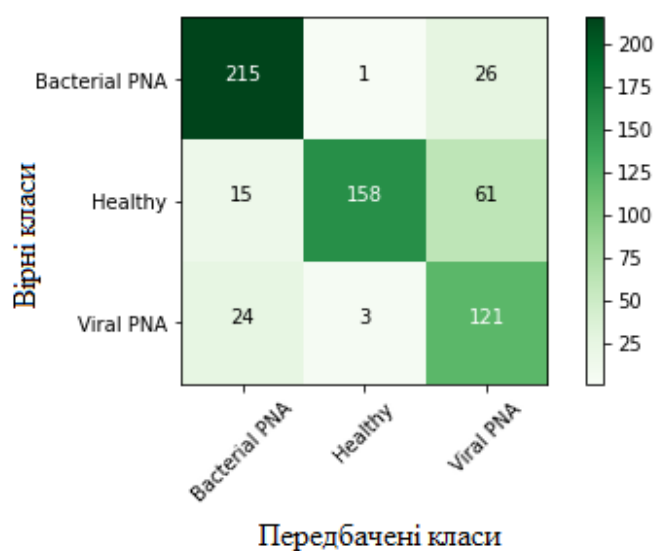


Рисунок 3.9 – Матриця невідповідностей

Більш розмита природа вірусної пневмонії, схоже, спричиняє появу рентгенівських променів, які більше нагадують здорові зразки. На неправильно передбачених зображеннях можна спостерігати поширеність вірусних випадків, що передбачені як бактеріальні або здорові випадки. Бактеріальні хибні прогнози менш вірогідні, тому можуть не з'являтися у випадковому відборі.

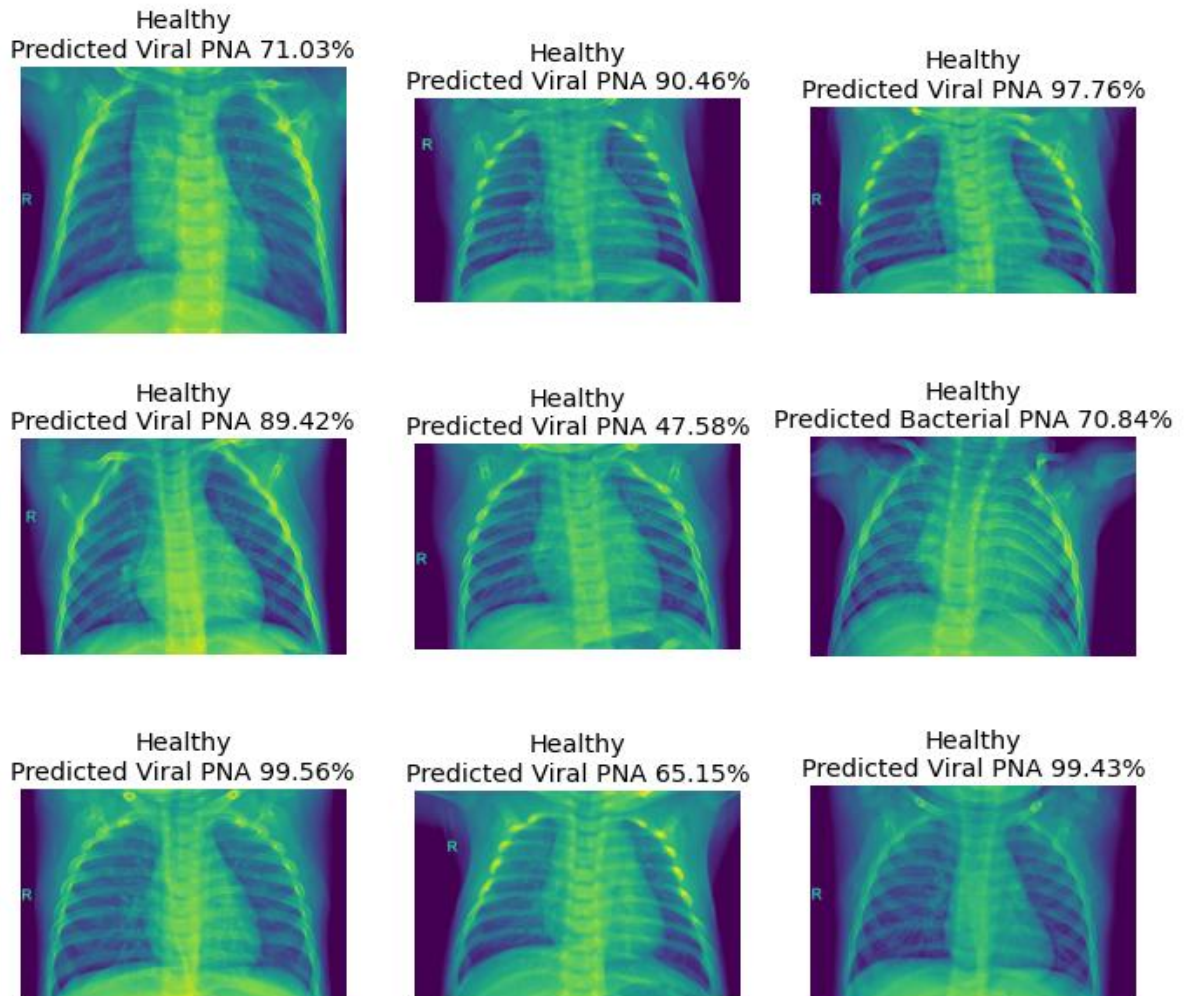


Рисунок 3.10 – Невірно передбачені знімки

Також на точність може впливати початковий розмір зображення, тому для аналізу представлений рисунок 3.11, з якого видно, що менші зображення розпізнаються більш точно. Це зв'язано з нерівномірністю навчального набору.

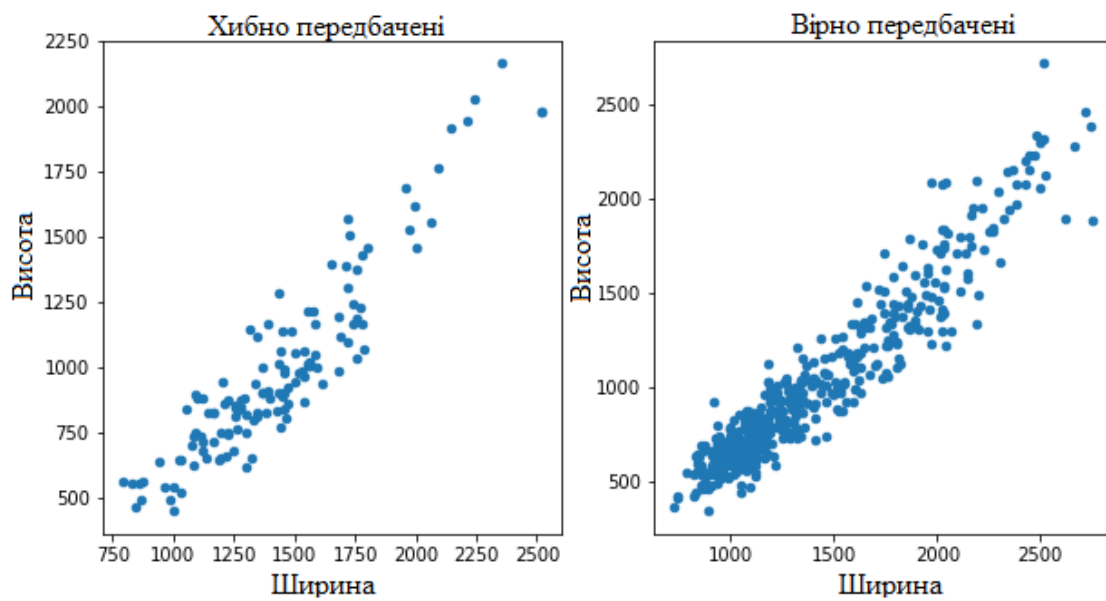


Рисунок 3.11 – Істинність класифікації в залежності від розміру знімка

РОЗДІЛ 4. ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ЗАХВОРЮВАНOSTІ НА ПНЕВМОНІЮ

Починаючи з 2019 року і по теперішній час, люди по всій планеті проживають у певних рамках, що постійно мають регулюватись. Адже на нас діють багато факторів: світова економіка, захворюваність, емоційна та фінансова складова. Попри вакцинацію, карантинні обмеження, тренд смертності від пневмонії, викликаний COVID-19, не йде на спад. [\[16\]](#)

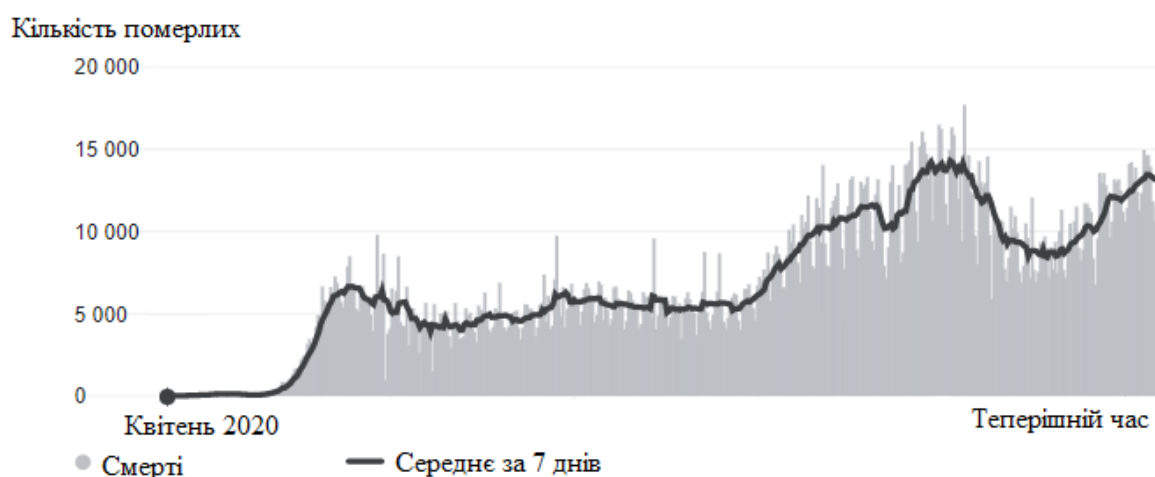


Рисунок 4.1 Значення смертності від COVID-19

Території ділять на рівні, в залежності від ситуації:

- Зелена зона – відсутні сплески захворюваності та більшість місць в лікарнях вільні;
- Жовта зона – більш ніж 5 днів показники захворюваності зростають, лікарні завантажені на половину;
- Помаранчева зона – завантаження лікарень від 50% до 75%;
- Червона зона – кількість хворих на 100 000 населення більше ніж 320 протягом п'яти днів поспіль. [\[17\]](#)


Region	Рівень епіднебезпеки
Цільовий показник	
Дата	
КИЇВ	
Вінницька	
Волинська	
Дніпропетровська	
Донецька	
Житомирська	
Закарпатська	
Запорізька	
Івано-Франківська	
Київська	
Кіровоградська	
Луганська	
Львівська	
Миколаївська	
Одеська	
Полтавська	
Рівненська	
Сумська	
Тернопільська	
Харківська	
Херсонська	
Хмельницька	
Черкаська	
Чернівецька	
Чернігівська	

Рисунок 4.2 – Області України станом на 16.05.2021

Завдяки навченому класифікатору хворих легень, можна робити неочевидні та корисні висновки. Люди розробили багато вакцин та ліків, ефективність яких можна спостерігати завдяки навченій моделі. Пацієнт робить знімок до лікування та після прийняття ліків. Оскільки навчена модель повертає ймовірність хвороби, то по зміні ймовірності можна робити висновки щодо ефективності препарату.

Масові джерела щодня нам розповідають про статистику захворюваності, тим самим підвищують нашу обережність та відповідальність перед оточенням. Але при занадто великому акценті на цю тему, підвищується рівень паніки населення. Якщо багато людей приходить на обстеження, маючи здоровий стан, необхідно зменшити інформування, адже чим менше контактів, тим краще. [\[18\]](#)

COVID-19 часто мутує, і прослідкувати розповсюдження кожного виду – нетривіальна задача. Якщо на деякій території в людей запалення легень почало швидше розвиватись, то необхідно надати цій території статус червоної зони, щоб зменшити розповсюдження нового захворювання. [\[19\]](#)

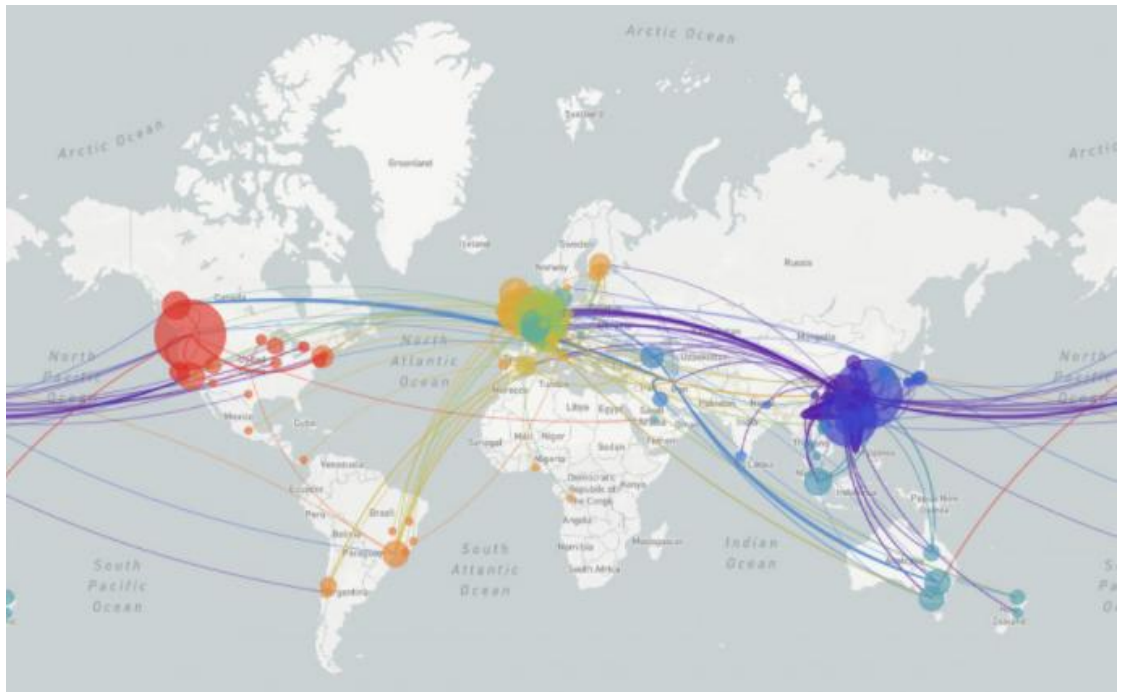


Рисунок 4.3 – Розповсюдження COVID-19

Також важливим є те, що комп'ютери здатні ставити діагноз, призначати ліки, визначати схильність кожної людини до певного захворювання. Це спрямовує до автоматизації лікування, стабільності медицини та підвищення її рівня. [\[20\]](#)

ВИСНОВКИ

Отже, під час пандемії прослідковується багато інформації про стан пацієнтів. Нейронні мережі здатні аналізувати її, визначати діагноз за знімком легень, завдяки чому можна зробити висновки про стан хвороби, динаміку її розвитку та дію препаратів.

Було проведено аналіз великої кількості знімків, та побудована згорткова нейронна мережа, що дозволила провести класифікацію всіх зображень. Досліджувалась загальна структура нейронних мереж та їх видів. Проводився процес побудови, навчання, тестування та вимірювання точності описаної моделі.

У зв'язку з пандемією необхідно застосовувати якнайбільше алгоритмів та проводити аналіз великої кількості даних, щоб ефективно боротися з цим нелегким періодом. Моніторинг стану пацієнта без людини значно спрощує та покращує процес лікування у масштабах медичних закладів.

Зібрані результати у великому об'ємі дають змогу робити висновки щодо розповсюдження захворювання, що дає можливість раціонально накладати обмеження на проблемні території.

У роботі було реалізовано згорткову нейронну мережу зі спеціальними шарами Xception, яка після навчання показала гарні результати з точністю 79.17%. Така нейронна мережа зможе класифікувати не тільки захворювання легень, а й інші знімки людських недуг.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. [Марченко О.О., Россада Т.В.](#) (2017). [“Актуальні проблеми Data Mining”](#) (PDF).
2. [Elizabeth A. Peck, G. Geoffrey Vining, Douglas C. Montgomery](#) (1982). [“Introduction to Linear Regression Analysis”](#).
3. [Toshnall Meenpal, Amit Verma](#) (2017). [“An Overview to Image Classification”](#) ISBN-13: 978-613-3-99025-8, ISBN-10:6133990252.
4. [Манденль І.Д.](#) (1988). [“Кластерний аналіз”](#) ISBN 5-279-00050-7.
5. Всесвітня організація здоров'я [Електронний ресурс]. – 2019. – Режим доступу до ресурсу <https://covid19.who.int/>.
6. [Орельєн Герон](#) (2017). [“Hands-On Machine Learning with Scikit-Learn and TensorFlow2”](#) ISBN: 9781491962299.
7. Набір даних MNIST [Електронний ресурс]. – 2017. – Режим доступу до ресурсу <https://scientistcafe.com/ids/mnist-dataset.html>.
8. [Марк Пітер Дейзенрот, Чен Сун Онг, А. Aldo Faisal](#) (2020). [“Mathematics for ML”](#) ISBN-13: 978-1108455145; ISBN-10: 110845514X.
9. [Krisnendu Chaudhury](#) (2020). [“Math and Architecture of Deep Learning”](#) ISBN 9781617296482.
10. [Леонардо де Марк](#) (2019). [“Hands-On Neural Networks: Learn how to build and train your first neural network model with using Python”](#) ISBN-13: 978-1788992596; ISBN-10: 1788992598.
11. Хсертіон: глибоке навчання з розділеними по глибині згортками [Електронний ресурс]. – 2017. – Режим доступу до ресурсу <https://arxiv.org/abs/1610.02357>.
12. Виявлення і аналіз COVID-19 на рентгенівських знімках грудної клітини на основі глибокого навчання [Електронний ресурс]. – 2021. – Режим доступу до ресурсу <https://link.springer.com/article/10.1007%2Fs10489-020-01902-1>.

13. Попереднє навчання по ImageNet покращує класифікацію медичних зображень [Електронний ресурс]. – 2018. – Режим доступу до ресурсу https://link.springer.com/chapter/10.1007%2F978-3-030-11024-6_37.
14. Метрики для оцінки якості алгоритму машинного навчання [Електронний ресурс]. – 2018. – Режим доступу до ресурсу <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234>.
15. Різниця між бактеріальною та вірусною пневмонією [Електронний ресурс]. – 2020. – Режим доступу до ресурсу [https://www.cebm.net/covid-19/differentiating-viral-from-bacterial-pneumonia/#:~:text=Community%2Dacquired%20pneumonia%20\(CAP\),it%20can%20be%20life%2Dthreatening](https://www.cebm.net/covid-19/differentiating-viral-from-bacterial-pneumonia/#:~:text=Community%2Dacquired%20pneumonia%20(CAP),it%20can%20be%20life%2Dthreatening).
16. Al-Turjman, Fadi (2021). [“Artificial Intelligence and Machine Learning for COVID-19”](#). ISBN 978-3-030-60188-1.
17. Карантинні заходи [Електронний ресурс]. – 2021. – Режим доступу до ресурсу <https://covid19.gov.ua/karantynni-zakhody>.
18. Вплив соціальних мереж та телебачення на паніку під час пандемії [Електронний ресурс]. – 2020. – Режим доступу до ресурсу <https://core.ac.uk/reader/323058200>.
19. Genomics Institute launches landing page for COVID-19 data resources [Електронний ресурс]. – 2020. – Режим доступу до ресурсу <https://news.ucsc.edu/2020/04/landing-page.html>.
20. Застосування нейронних мереж в медицині [Електронний ресурс]. – 2020. – Режим доступу до ресурсу <https://www.sciencedirect.com/science/article/pii/B97801281894670007X>.