

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації та управління

Спеціальність 126 Інформаційні системи та технології

ЗВІТ

з переддипломної практики бакалаврів
на тему:

„Інформаційна система з підтримки процесу розпізнавання пневмонії за
рентгенівськими знімками”

Місце проходження практики:

ТОВ «Netcracker»

Виконав студент

ІС-71 Алпаєва Юлія Сергіївна
(шифр групи, прізвище, ім'я, по батькові)



(підпис)

Керівник практики
від університету

Телишева Тамара Олексіївна
(прізвище, ім'я, по батькові)

(підпис)

Керівник практики

від підприємства

Білорус Дмитро Сергійович
(прізвище, ім'я, по батькові)



(підпис)

Дата захисту

Оцінка

„_____”

Київ 2021

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Пневмонія залишається гострою проблемою в Україні і світі. Це зумовлено, її поширеністю, високими показниками смертності, а також значними економічними втратами внаслідок цього захворювання.

Пневмонія — гостре інфекційне захворювання, переважно бактеріальної етіології, яке характеризується вогнищевим ураженням респіраторних відділів легень та наявністю внутрішньоальвеолярної ексудації[1].

Зазвичай захворювання виникає через ураження бактеріями, рідше вірусами чи іншими мікроорганізмами, наприклад грибки та найпростіші. В першу чергу хвороба уражає альвеолами, повітряні міхурці.

Вченими виявлено більш 100 видів виявлених збудників запалення легень, лише декілька з них спричиняють більшість випадків пневмонії. При захворюванні у хворого наявні такі симптоми як кашель, біль у грудях, гарячка, ускладнене дихання, хрипіння, озноб.

Серед симптомів частіше спостерігається втома у 90% хворих, кашель та підвищена температура у 75-79% хворих.

Серед хворих на пневмоніюїї переважають чоловіки - 55%. Захворюваність пневмонією збільшується з віком. Найбільш висока летальність спостерігається серед осіб старше 55 років. У групі ризику люди похилого віку, курці, алкоголіки: куріння, також пацієнти, які мають імунodefіцит, хронічну обструктивну хворобу легенів, хронічну хворобу нирок та хворобу печінки.

Фактори ризику:

1. Паралельним захворюванням при гострій пневмонії є запальні захворювання носоглотки та придаткових пазух. Через те що порушується носове дихання, створюються умови для попадання інфікованого секрету в бронхи.

2. Також фактором ризику є бронхіальна астма, хронічні обструктивні бронхіти, локальні обструкції бронхів пухлиною, стороннім тілом, це призводить до порушення перистальтичних скорочень бронхів а

також мукоциліарного транспорт, а це в свою чергу призводить до затримки слизу.

3. У хворих на алкоголізм порушений глотковий рефлекс, що призводить до періодичної аспірації ротоглоткової флори, це теж підвищує ризик захворювання.

4. Через вдихання токсичних речовин у курців страждає війчастий епітелій, розвивається функціональна недостатність альвеолярних макрофагів, вдихання викликає поширені опіки слизової оболонки бронхолегеневого апарату, це підвищує ризик вторинного інфікування. Виникає порушення дренажної функції, це сприяє бронхогенному поширенню інфекції

Для діагностики застосовують рентгенографію легень і бактеріологічний посів мокротиння, аналіз крові.

Рентгенологічна картина залежить від фази розвитку запального процесу. В період вираженої клінічної картини хвороби на рентгенограмі виявляють вогнищеві тіні різних розмірів, часто розміщених локалізовано. У випадку сегментарної пневмонії спостерігається гомогенна тінь з чіткими контурами в межах сегменту. [2]

У стадії захворювання «прилив» на рентгенограмах визначається підсилення легеневого малюнку а також зниження прозорості фону через переповнення кров'ю легневих судин. Якщо у пацієнта на ренгені зона ураження менша за одну долю, діагностика змін стає складнішою. Корінь легені яка є ураженою розширений а також його структура розмита. При ураженні нижньодольових сегментів спостерігається зниження рухомості куполу діафрагм.[3]

У стадії гепатизації є гомогенні інтенсивні затемнення, які за щільністю нагадують ателектаз без зміщення органів середостення у бік ураження. Інтенсивність тіні у напрямку до периферії збільшується. При масивній крупозній пневмонії із залученням цілої долі легені тінь однорідна на всьому протязі. Найбільш частим для крупозної пневмонії є ураження 2-3 сегментів (70% хворих). 1-3 сегменти вражаються тільки у 5% хворих. Пневмонія правосторонньої локалізації спостерігається у 1,7 разів частіше, ніж

лівосторонньої. Ураження ділянок легень уздовж міждольових щілин можна діагностувати тільки рентгенологічно - аускультативні симптоми не виявляються, оскільки фокус розташований дуже глибоко. [3]

1.1.1 Опис процесу діяльності

Дана робота присвячена аналізу захворювань легень за допомогою рентгенівських знімків.

Оскільки це найпоширеніший і доступний спосіб діагностики хвороб легень. З розвитком інформаційних технологій і збільшенням об'ємів інформації постала проблема обробки і аналізу. Оскільки захворювання легень є досить поширеними, а лабораторії в деяких країнах світу не справляються з швидкістю захворюваності – проблема є досить гострою.

Для вирішення її пропонується сервіс розпізнавання пневмонії за рентгенівським знімком.

Раніше до створення даного сервісу знімки міг аналізувати лише експерт – лікар рентгенолог, за рядом ознак він робив висновки щодо захворювання. З появою даного сервісу аналіз може зробити і людина без медичної освіти, для отримання результату потрібно завантажити знімок у систему і алгоритм дасть відповідь. Дане програмне забезпечення може використовуватись експертом для першого аналізу зображення, а бо при великих навантаженнях на систему охорони здоров'я для повноцінного автоматизованого аналізу, та збору статистики про захворюваність. Це пришвидшить процес лікування та діагности захворювання.

Завдяки розробленій системі процес аналізу спроститься і не потребуватиме кваліфікованого лікаря, користувач лише має мати базові знання використання комп'ютера. Це допоможе провезти первинний аналіз перед заключенням лікаря, без участі самого лікаря.

Така система може бути корисна якщо наприклад в лікарні немає лікаря рентгенолога і він може працювати лише декілька днів на тиждень, або на декілька тижнів. Проведення рентгенів не буде призупинятися при відсутності фахівця, а також коли лікар розпочне діяльність, він одразу зможе бачити які знімки з наявними змінами і зможе їх дослідити першочергово.

Або така система може бути корисною якщо існує велике навантаження на лікарню, або на систему охорони здоров'я, пацієнти знімки яких система визначила як хворих, можуть бути оглянуті першочергово.

1.1.2 Опис функціональної моделі

Спроекуємо функціональну модель системи, опишемо дійових акторів системи і дії, які вони можуть виконувати з системою.

У системі аналізу рентгенівських знімків є один актор – користувач, який проводить аналіз зображення.

Користувач може проводити аналіз рентгенівського знімку, та переглядати збережені результати, а також логінитись і створювати акаунт, переглядати свій вкаунт.

Розглянемо дії, які може виконати користувач системи, представленні на схемі варіантів використання.

Користувач в системі може:

- авторизуватися: користувач може зайти в існуючий акаунт за допомогою логіну і паролю. Якщо користувач не має акаунта, він може його створити, ввівши адресу електронної пошти, ім'я користувача, і пароль. Без авторизації користувач не матиме змоги проводити аналіз зображення і переглядати збережені результати.

- переглянути свій профіль: користувач має змогу переглянути профіль, дані які використовувались при реєстрації.

- провести аналіз зображення: користувач може завантажити зображення з файлової системи в дозволеному форматі, відправити дані для аналізу, та отримати розгорнутий результат аналізу.

- продивитись історію досліджень: користувач має можливість передивитись історію досліджень, які він проводив у системі.

1.2 Огляд наявних аналогів

В ході проведеного аналізу не було знайдено аналогів програмного забезпечення з представленою функціональністю, але можна порівняти різні підходи розв'язання даної задачі.

1. В роботі розглядається два методи для визначення пневмотораксу за рентгенівськими знімками.

Перший метод ,що розглядається метод опорних векторів. Риси відокремлюються за допомогою Локального бінарного шаблону. Класифікація рентгенівських зображень визначається Методом опорних векторів.

Другий метод досліджений в роботі оснований на сегментації по інтенсивності досліджуваного зображення. Для сегментації аномальних областей видаляється фон та шуми, границі ребер визначаються за допомогою оператора Собеля. Розглядається два класи зображень – знімки у яких змін не виявлено і знімки де наявний пневмоторакс.

70% знімків легенів (36 нормальних випадків та 22 випадки пневмотораксу) використовувались для фази тренувань, тоді як 30% знімків легенів (16 нормальних випадків та 10 випадків пневмотораксу) використовувались для фази тестування. Проведено 5 випробувань отриманий із варіацією точності від 76,9% до 88,4%.[4-5]

2. Дослідження проведено групою Індійських учених. Станом на березень 2011 року Індія є другою за чисельністю населення країною в світі. Більшість населення проживає в сільських районах і не має доступу до медицини, через це частота тестування населення є досить низькою, що призводить до швидкого поширення захворювання COVID-19 на території Індії. Також великою проблемою є те що на 1 000 000 людей лише 3 рентгенологи, що робить майже неможливим дослідження великої кількості знімків.

Тому групою індійських вчених була запропонована автоматизована система з розпізнавання пневмонії , COVID-19 та легень без аномалій за рентгенівськими знімками

Розпізнавання виконується за допомогою KE Sieve алгоритму.

Розглядається три класи зображень: зображення без змін, зображення з пневмонією та з COVID-19 .

Алгоритм KE Sieve є неітераційним і має можливість інкрементально навчатись.

Для випадків пневмонії отримано точність 99%, для випадку COVID-19 - 100%, для легень без змін 96% [6]

3. Розробка COVID-GAPS, фреймворку для дослідження COVID-19 за рентгенівськими знімками. Фреймворк базується на капсульних мережах - CapsNet. Архітектура розпізнавача складається з чотирьох згорткових шарів та трьох капсульних шарів. Вхідними даними є 3D рентгенівські знімки. Розпізнавання виконується за двома класами є COVID-19 чи немає. Для навчання використовувався датасет з чотирма класами зображень: здорові легені, бактеріальна пневмонія, вірусна пневмонія, COVID-19.

Використовуючи набір даних, запропонований COVID-CAPS досяг точності 95,7%, чутливості 90%, специфічності 95,8% та AUC 0,97. [7]

4. У роботі за допомогою машинного навчання автоматизують процес виявлення різних захворювань на рентгенівських знімках грудної клітки. Досліджується ефективність сегментації легенів та виключення тіней для аналізу 2D знімків легень методами глибокого навчання, щоб допомогти рентгенологам виявити підозрілі зміни та новоутворення у хворих на рак легенів. Точність розпізнавання отримана 99%. [8-9]

Отже, після проведеного аналізу аналогів можемо дійти висновку, що існує багато наукових робіт, досліджень в галузі розпізнавання захворювань за рентгенівськими знімками ,і результати отримані дуже хороші, але не існує якогось додатку який міг би використовуватися лікарями для аналізу , отже потрібно розробити систему, яка дозволить зробити аналіз знімку хворого і зробити висновок про наявність або відсутність змін у легенях.

1.3 Постановка задачі

1.3.1 Призначення розробки

Призначення розробки - розпізнавання пневмонії за рентгенівськими знімками збережених в різних форматах.

1.3.2 Цілі та задачі розробки

Метою розробки є полегшення процесу розпізнавання пневмонії, збільшення точності аналізу при аналізі хворих, надання можливості аналізування даних без участі лікаря.

Для розробки даного програмного забезпечення мають бути вирішені такі проблеми:

- Розробити архітектуру нейронної мережі
- Підтримка можливості зручного перегляду історії зображень
- Підтримка можливості завантажити зображення для аналізу
- Виведення результату в зручному і зрозумілому для користувача навіть без медичної освіти форматі.

- Підтримка аналізу зображень різних розмірів та форматів

- Виведення результату в зручному і зрозумілому для користувача навіть без медичної освіти форматі.

- Підтримка можливості увійти в систему
 - Розробка функціоналу створення акаунту
 - Зібрати достатньо великий датасет для навчання та тестування роботи нейронної мережі. Датасет має містити достатню кількість прикладів здорових легень і легень зі змінами.

- Навчання нейронної мережі на прикладах

Висновок до розділу

На даному етапі було описано предметне середовище , досліджені його особливості. Досліджене поняття пневмонії як захворювання, його розповсюдженість і небезпечність для різних категорій населення. Була описана функціональна модель, описані актори, які можуть використовувати систему, були описані дії ,які вони можуть виконувати з системою. Використовуючи цю інформацію були досліджені можливі підходи для розв’язання такої проблеми та аналоги. В ході проведеного дослідження стало зрозуміло що аналогів подібних до даної системи немає. Отже була досліджена актуальність програмного продукту, сформульовано призначення розробки, і досліджені проблеми, які може вирішувати система.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

В системі існує два типи вхідних даних : дані для навчання нейронної мережі, та дані користувача. До першої категорії відноситься датасет з зображеннями легень з мітками, на якому навчалася нейронна мережа. До другої категорії відносяться данні користувача, які він вносить при реєстрації а саме : ім'я користувача в системі, адреса електронної пошти, пароль для входу в систему.

Опишемо докладніше першу категорію даних.

Для навчання нейронної мережі потрібен досить великий набір даних, які заздалегідь промарковані. При навчанні нейрона мережа знає до якої категорії відноситься те чи інше зображення. В даній роботі використовувалось навчання нейронної мережі – з учителем.

Проблема збору даних є чи найважливішою при створенні класифікатора, оскільки від кількості та якості даних залежить точність роботи нейронної мережі.

Для тренування нейронної мережі було використано

- Зображення здорових легень: 1341 приклади
- Зображення легень з пневмонією: 3875 приклади

Для валідаційного датасету було використано:

- Зображення здорових легень: 8 прикладів
- Зображення легень з пневмонією: 8 прикладів

Вибірка включає в себе достатньо багато прикладів для навчання з обох категорій, але прикладів пневмонії значно більше, що робить датасет незбалансованим, при навчанні нейронної мережі це було враховано. Набір даних містить багато зображень різної якості та розмірів, це допоможе отримати точні результати класифікатора при роботі з даними різної якості кінцевого користувача.

2.2 Вихідні дані

Вихідними даними є об'єкт, що містить значення ймовірності з якою зображення належить до того чи іншого класу.

Такий формат дозволяє оцінити з якою ймовірністю пацієнт має пневмонію.

2.3 Опис структури бази даних

Розглянемо структуру бази даних. База даних містить такі таблиці:

- Images: таблиця містить результати аналізу зображення.
- User: таблиця містить дані користувача.

Опишемо структуру таблиці Images, наведено в таблиці 2.1.

Таблиця 2.1 – Опис таблиці Images

Назва поля	Тип даних	Опис
Id	INTEGER	Унікальний Id запису, первинний ключ в таблиці
result	VARCHAR	Результат перевірки зображення
conclusion	VARCHAR	Загальні висновки щодо отриманого результату
probability	VARCHAR	Вірогідність належності зображення до класу пневмонія
Назва поля	Тип даних	Опис
Email	VARCHAR	електрона адреса користувача, що проводив аналіз зображення
identifier	VARCHAR	Унікальний ідентифікатор перевірки
date	VARCHAR	Дата і час аналізу
status	VARCHAR	Статус аналізованого зображення, в залежності від

		вірогідності.
--	--	---------------

Опишемо структуру таблиці Users, наведено в таблиці 2.2.

Таблиця 2.2 – Опис таблиці Users

Назва поля	Тип даних	Опис
id	INTEGER	Унікальний Id запису, первинний ключ в таблиці
email	VARCHAR	Електронна адреса користувача
password	VARCHAR	Пароль користувача
Назва поля	Тип даних	Опис
name	VARCHAR	Ім'я користувача в системі

Висновок до розділу

В даному розділі були описані вхідні дані, їх джерела. Оскільки для класифікації буде використовуватися нейрона мережа, були описані вимоги, яким мають відповідати шукані дані. Також були описані проблеми з які наявні в знайдених даних.

Наряду з вхідними даними, були описані вихідні дані, їх форма, та інформація, яку вони будуть нести кінцевому користувачу при аналізі.

Наприкінці розділу була описана структура бази даних, вміст таблиць та зв'язки між ними.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Існує набір даних великого розміру з рентгенівськими знімками. Зображення мають різний розмір та якість. Кожне зображення має ярлик, що означає наявність або відсутність пневмонії на знімках. Необхідно на основі навчальної вибірки створити алгоритм, що зможе за зображенням легенів класифікувати знімок або до класу з наявною пневмонією, або з відсутньою.

3.2 Математична постановка задачі

Для розв'язку поставленої задачі була використана штучна нейронна мережа. Вона складається із шарів нейронів пов'язаних між собою. Мережа навчається за допомогою ітеративного алгоритму навчання, в кінці кожної ітерації навчання обчислюється функція витрат, що характеризує наскільки результат розпізнавання мережею відхиляється від результатів вказаних на ярликах. Після ітерації навчання коригується значення ваг мережі (алгоритмом градієнтного спуску), результатом навчання вважається такий набір ваг при якому функція витрат приймає мінімальне значення.

Введемо позначення, які описують нейронну мережу:

W – матриця ваг

X – матриця значень сили імпульсів, які приходять на входи

Y – матриця значень функції активації

dW – матриця градієнтів

$E = \frac{1}{2}(\text{label}_j^{(n)} - y_j^{(n)})^2$ – функція витрат

Нейрони, які утворюють один шар мережі з'єднані між собою зв'язками, які передають сигнали. Штучний нейрон, який отримав сигнал, обробляє його за допомогою функції активації, й потім передає його іншим нейронам, з'єднаним з ним.

Розглянемо найпростішу модель нейрона, наведено на рисунку 3.2.

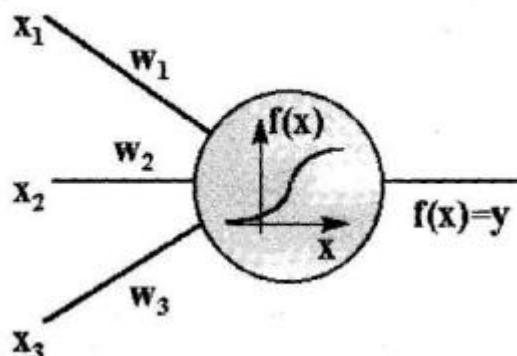


Рисунок 3.2 - Модель найпростішого нейрона [10]

Введемо позначення:

w_i - вага i – того входу.

x_i - сила імпульсу i – того входу.

$f(x)$ - функція активації

На рисунку 3.2 зображена модель простого нейрону з трьома входами, кожний вхід має вагу w_1, w_2, w_3

Нехай до зв'язків надходять імпульси силою - x_1, x_2, x_3 , тоді відповідно до нейрона надходять імпульси - w_1x_1, w_2x_2, w_3x_3 .

Тоді сумарний імпульс, який отримає нейрон

$$x = w_1x_1 + w_2x_2 + w_3x_3$$

Сила вихідного імпульсу дорівнює

$$y = f(x) = f(w_1x_1 + w_2x_2 + w_3x_3)$$

Вибір функції активації $f(x)$ залежить від архітектури нейронної мережі.

Для розпізнавання буде використовуватись мережа з згортковими шарами.

Опишемо операцію згортки і результат функції активації для нейрону згорткового шару.

Обчислимо результат згортки $(I * K)$,

де

K - матриця розмірності $(m \times n)$, фільтр згортки або рецептивне поле, побудована таким чином, що кодує в собі якусь ознаку, яка може бути наявною на шуканому класі зображень.

I - двовимірне зображення.

Обчислюється згортка $(I * K)$, накладаючи на зображення I фільтр K всіма можливими способами. Проілюструємо операцію згортки на рисунку 3.3

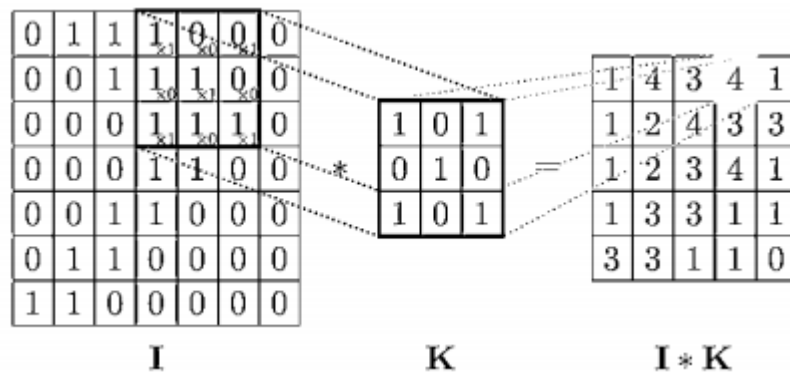


Рисунок 3.3 – Процес згортки

В результаті операції отримаємо

$$(I * K)_{xy} = \sum_{i=1}^m \sum_{j=1}^n K_{ij} \times I_{x+i-1, y+j-1}$$

Розмір фільтру K обирається в залежності від задачі.

Отже, вхідне значення для нейрону згорткового шару l обчислюється за формулою

$$x_{ij}^{(l)} = f(b + \sum_{r=0}^n \sum_{c=0}^n w_{r,c} x_{i+r, j+c}^{(l-1)})$$

Розв'язком задачі є такий набір ваг мережі w_i при якому значення функції витрат мережі є мінімальним

$$E = \frac{1}{2} (label_j^{(n)} - y_j^{(n)})^2 \rightarrow \min$$

3.3 Обґрунтування методу розв'язання

Штучна нейрона мережа — це математична модель, що представляє собою систему з'єднаних і взаємодіючих між собою простих процесорів (штучних нейронів).[10]

Такі системи навчаються на прикладах конкретних даних розв'язувати проблему, при використанні не потребують спеціального програмування під конкретну задачу.

Приклад архітектури наведено на рисунку 3.1

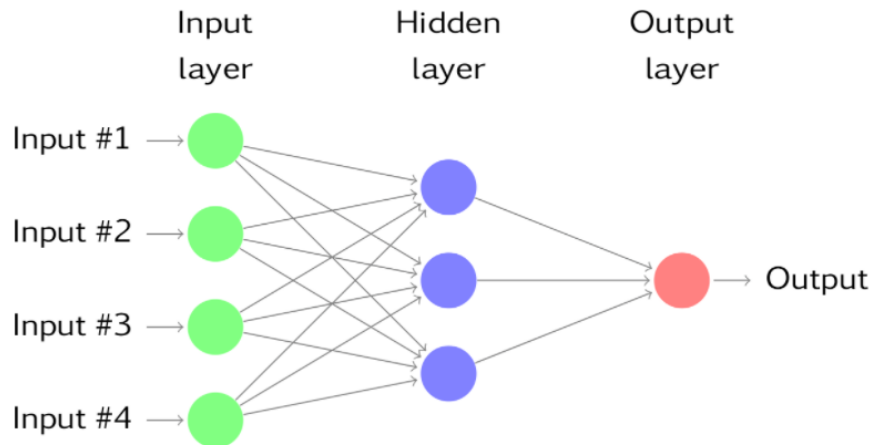


Рисунок 3.1- Приклад архітектури нейронної мережі

На рисунку 3.1 показано нейронну мережу з чотирма предикторами.

Згорткові нейронні мережі — це клас штучних мереж прямого поширення, особливість цієї архітектури полягає в тому що для розв'язку задачі використовуються згорткові шари. Ці шари нейронної мережі застосовують до матриці вхідного зображення операцію згортки. Потім передаючи результат згортки до наступного шару мережі. Операція згортки імітує виокремлення ознаки зображення.

Успішність застосування цього алгоритму для розпізнавання образів пов'язана з тим, що згорткові мережі використовують фільтри для розпізнавання окремих ознак. А також, те що обробка конкретної частини зображення проходить незалежно від місця її знаходження.

Операція згортки повторює процес виокремлення деталей оком людини, за допомогою фільтрів, які використовуються для операції згортки.

Шар згортки може мати будь-яку кількість фільтрів різного розміру. Фільтри у процесі навчання підбираються під конкретну задачу класифікації. Навчання згорткової мережі - це навчання цих фільтрів, які є вагами для мережі. Таким чином, мережа може повністю розкрити деталі вхідного зображення, які є важливими для розпізнавання.

Також алгоритм не потребує такої сильної попередньої обробки зображення, як інші алгоритми розпізнавання зображень.

Саме ці відмінності роблять згорткові мережі точними для задачі розпізнавання образів.

3.4 Опис методів розв'язання

Алгоритм навчання нейронної мережі -

1. Ініціалізувати всі ваги $\{w_{ij}^l\}$ рандомним чином ,малими числами,

Де $\{w_{ij}^l\}$ - це вага на шляху між i - нейроном $(l-1)$ шару та j – нейроном l – шару.

2. Алгоритм прямого розповсюдження

(a) На вхід 0 шару мережі надати приклад з датасету, проставити вхідне значення для кожного нейрону, і мітку для вихідного шару мережі.

(b) Розрахувати загальне вхідне значення від 0 шару до внутрішнього кожного нейрону наступного шару мережі x_j^l , розрахувати вихідне значення y_j^l за допомогою функції активації. Повторити процес для інших шарів мережі. Для них вхідним значенням буде вихідне значення передуючого шару.

$$X_j^l = Y^{l-1} W^{l-1} \quad W^l \in R(d(l-1) \times d(l))$$

$$Y_j^l = f(X^l) \quad X^l \in R(d(l) \times 1)$$

3. Алгоритм зворотного розповсюдження

(a) Розрахувати dW матрицю з $n-1$ шару до n отже $(l=n)$. Функція витрат

$$E = \frac{1}{2} (\text{label}_j^{(n)} - y_j^{(n)})^2$$

Розрахуємо градієнт для кожної ваги між останнім та передостаннім шаром мережі.

$$\delta^{(n)} = -(E - Y^{(n)}) f'(X^{(n)}), \delta^{(n)} \in R(d(n) \times 1)$$

$$dW^{(n-1)} = (Y^{l-1})^T \delta^{(n)}$$

(b) Розрахувати градієнт для передостаннього шару $(l=n-1)$

$$\delta^{(l)} = W^{(l)} \delta^{(l+1)}$$

$$dW^{(l-1)} = (Y^{(l-1)})^T \delta^{(l)}$$

(c) Оновити значення ваг для кожної ваги нейронної мережі

$$W^{(l)} = W^{(l)} - \eta dW^{(l)}$$

де $0 < \eta < 1$ - швидкість навчання

4. Повернутися на крок 2 і повторювати алгоритм стільки разів скільки задано ітерацій навчання.

Для розв'язання заданої задачі розроблялась і навчалась модель нейронної мережі. Нейронна мережа складається з трьох згорткових шарів, двох pooling шарів, та одного dropout шара.

Архітектура мережі наведена на рисунку 3.4.

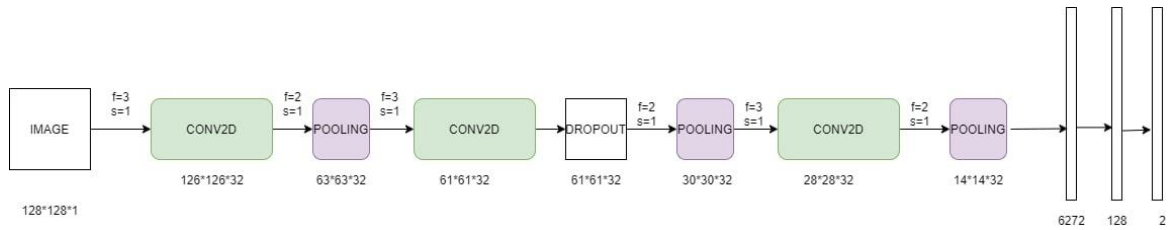


Рисунок 3.4 – Архітектура нейронної мережі.

Результати навчання на тестовому та валідаційному датасеті:

Графік залежності функції помилок від кількості епох навчання наведено на рисунку 3.5.

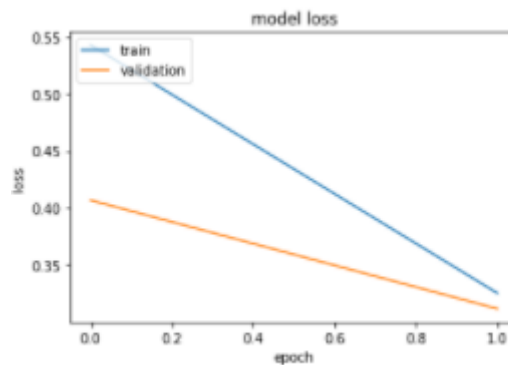


Рисунок 3.5 - Графік залежності функції помилок від кількості епох навчання

Графік залежності точності від кількості епох навчання наведено на рисунку 3.6.

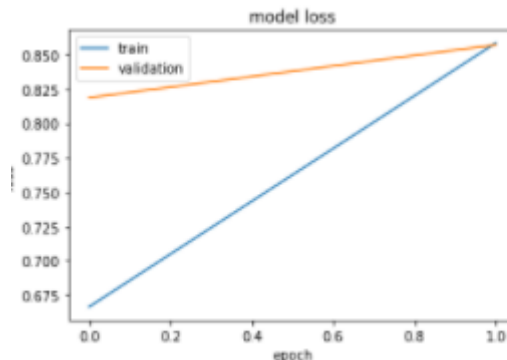


Рисунок 3.6 - Графік залежності точності від кількості епох навчання.

Точність розпізнавання моделі та значення функції витрат наведено на рисунку 3.7:

Accuracy of the Model is 85.73718070983887
Loss of the Model is 0.31190183758735657

Рисунок 3.7 - Значення точності моделі і значення функції витрат.
Матрицю помилок наведено на рисунку 3.8:


```
Found 624 images belonging to 2 classes.  
Confusion Matrix  
[[105 129]  
 [168 222]]
```

Рисунок 3.8 - Матриця помилок.

Звіт класифікації наведено на рисунку 3.9.

```
Classification Report
```

	precision	recall	f1-score	support
Pneumonia	0.38	0.45	0.41	234
Neutral	0.63	0.57	0.60	390
micro avg	0.52	0.52	0.52	624
macro avg	0.51	0.51	0.51	624
weighted avg	0.54	0.52	0.53	624
samples avg	0.52	0.52	0.52	624

Рисунок 3.9 - Звіт класифікації.

Як бачимо точність моделі складає 85%. При навчанні інших моделей було отримано вищу точність, це досягалось за рахунок кращого розпізнавання нормального стану легенів, але було прийняте рішення зупинитись саме на цій моделі, оскільки для задачі важливіше точніше класифікувати випадки з пневмонією.

Приклад розпізнавальних даних наведено на рисунку 3.10.

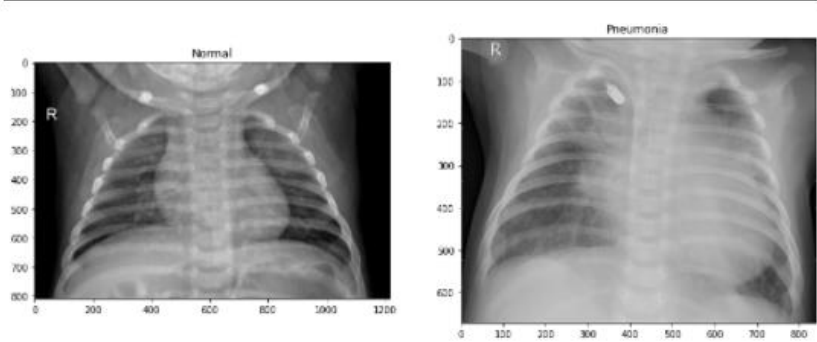


Рисунок 3.10 – Приклад аналізованих даних

Кількість прикладів кожного класу наведено на рисунку 3.11.

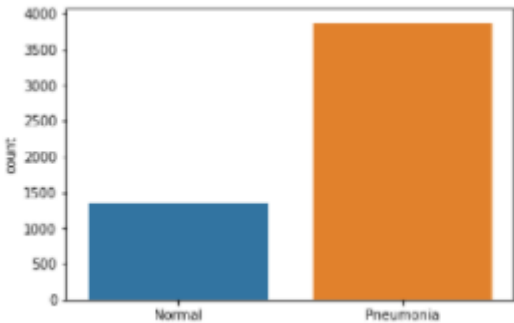


Рисунок 3.11 – Приклад аналізованих даних

Як бачимо з графіку зображеного на рисунку 3.11 датасет є незбалансованим, прикладів пневмонії значно більше ніж прикладів норми. Ця проблема була розв’язана за допомогою надання різної ваги для кожного класу

при навчанні. Для рішення проблеми перенавчання була проведена аугументація даних.

Висновок до розділу

В даному розділі було сформульовано змістовну постановку задачі. Окреслена проблема, яку має вирішувати класифікатор. Визначена форма вхідних даних, на яких буде навчатися класифікатор. В розділі Математична постановка задачі було визначено поняття нейронної мережі, і її складових. Описано модель штучного нейрона, як складової нейронної мережі. Також було сформульовано алгоритм навчання мережі. Наприкінці пункту було визначено поняття згортки та згорткового шару мережі. В розділі огрунтування методів розв'язання було проаналізовано згорткові мережі, як метод розпізнавання образів. Було наведено переваги цього методу над іншими для розв'язання задачі розпізнавання. Ми дійшли висновку, що така особливість згорткових шарів, як фільтри в згорткових шарах може суттєво вплинути на точність роботи класифікатора, а також великою перевагою стало те що даний алгоритм потребує меншої обробки даних. В Розділі Опис методу розв'язання була описана архітектура нейронної мережі, яка використовувалась для задачі розпізнавання випадків з пневмонією та нормальних випадків. Були наведені результати експериментів на тестовому датасеті, а саме: графік витрат моделі, матриця помилок, звіт класифікації та обраховано значення точності моделі та значення функції помилок.

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Для реалізації даного програмного продукту були використані такі технології:

- Python
- Flask
- Jinja2
- SQL lite
- Kaggle
- HTML
- CSS

Python— це потужна мова програмування, розроблена в 1990 році Гвідо ван Россумом. Він має простий і зрозумілий синтаксис, отже можна швидко розробити код. Мова має ефективні структури даних високого рівня та ефективний підхід до об'єктно-орієнтованого програмування. Пайтон має динамічну обробку типів і динамічну симантику, через те що мова є інтерпритовою надає можливість легко писати скрипти. Важливим є те що мова є кросплатформена, отже додаток буде працювати на будь-яком середовищі. Також має багато бібліотек та модулів для розробки додатків та математичних досліджень. Досить популярними є Numpy,Pillow,Requests.

Flask — мікрофреймворк для веб-додатків. Він створений з використанням мови програмування Python на основі бібліотеки Werkzeug, та має двіжок для шаблонізатора Jinja2. Flask має багато властивостей,що направлені полегшити розробку веб додатків, а саме: має сервер для розробки та відлагоджувач, вбудовану підтримку юніт-тестів,управління запитами RESTful,зручне використання шаблонів Jinja2 ,має підтримку безпечної сесії на стороні клієнта. А також має докладну документацію, що робить його зручним для вивчення.

Jinja2 - Python-бібліотека для рендеринга шаблонів, що є стандартом при написанні веб-додатків на Flask . Не дивлячись на прив'язку до мови, Jinja2 позиціонує себе як інструмент для дизайнерів і верстальників, це спрощує верстку і відокремлює її від розробки, це дозволяє розробляти фронтенд швидко і не залежити від програмного коду бекенду.

SQLite - компактна вбудована реляційна база даних. БД є кросплатформеною, а також безпечною для зберігання даних, станом на 2009 рік покриття тестами складає 100%, як заявляють розробники. SQLite не використовує парадигму клієнт-сервер, БД не є окремим процесом, з яким взаємодіє програма, а надає бібліотеку,

яка вбудовується в програму і БД стає складовою частиною програмного коду ,що означає що в якості протоколу обміну використовуються виклики функцій бібліотеки. Такий підхід зменшує накладні витрати, час відгуку і спрощує програму.

Kaggle - система організації конкурсів з дослідження даних. Система організована як публічна веб-платформа, на якій користувачі та організації можуть публікувати набори даних, досліджувати набори даних ,створювати моделі машинного навчання. В системі розміщені датасети, надаються хмарні інструменти для обробки даних і машинного навчання.

Для навчання нейронної мережі використовувались такі бібліотеки:

- Keras — відкрита бібліотека, написана мовою Python. Була спроектована для швидких експериментів з мережами глибинного навчання. Важливими перевагами її є те що вона зручна в користуванні, модульна та розширювана.
- Scikit-learn це безкоштовна програмна бібліотека машинного навчання. Бібліотека надає функціональність для створення та тренування різноманітних алгоритмів машинного навчання, наприклад лінійна регресія, random forest, градієнтний бустинг. Scikit-learn є однією з найбільш популярних бібліотек машинного навчання
- Python Imaging Library - бібліотека мови Python, призначена для роботи з зображеннями.
- matplotlib – безкоштовна бібліотека для побудови графіків функцій та різних діаграм;
- os- модуль для роботи з файловою системою.
- NumPy — бібліотека з відкритим вихідним. Надає можливості для роботи з багатовимірними масивами та підтримує високорівневі математичні функції для багатовимірних масивів

4.2 Вимоги до технічного забезпечення

4.2.1 Загальні вимоги

Особливих вимог до технічного забезпечення для запуску програмного продукту не висувається, потрібен лише комп'ютер зі встановленим браузером.

4.2.2 Архітектура програмного забезпечення

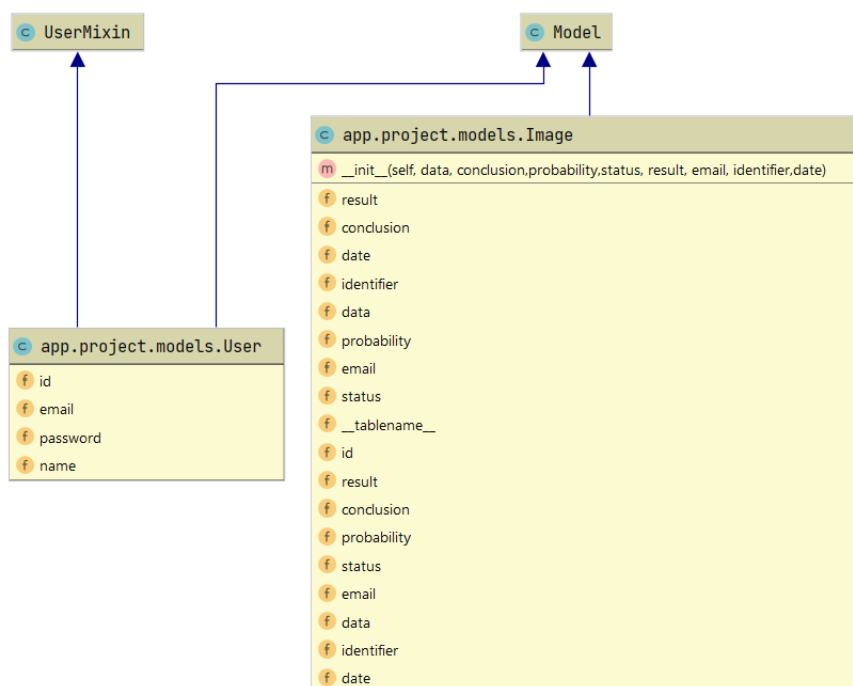
Для розробки даного забезпечення була обрана Трирівнева архітектура. Трирівнева архітектура - це модульна клієнт-серверна архітектура, яка складається з рівня представлення, рівня бізнес-логіки і рівня даних. Рівень даних забезпечує зберігання інформації, рівень бізнес-логіки обробляє логіку, а рівень представлення являє собою графічний інтерфейс користувача , який

взаємодіє з двома іншими рівнями відправляючи дані до користувача. Ці три рівня є логічними, а не фізичними, і можуть працювати як на одному фізичному сервері, так і на різних машинах. В випадку даного застосунку всі рівні будуть знаходитися на одному сервері.

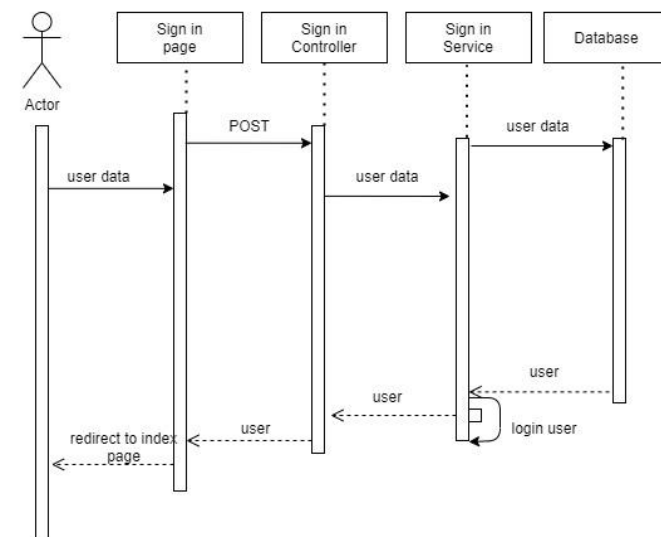
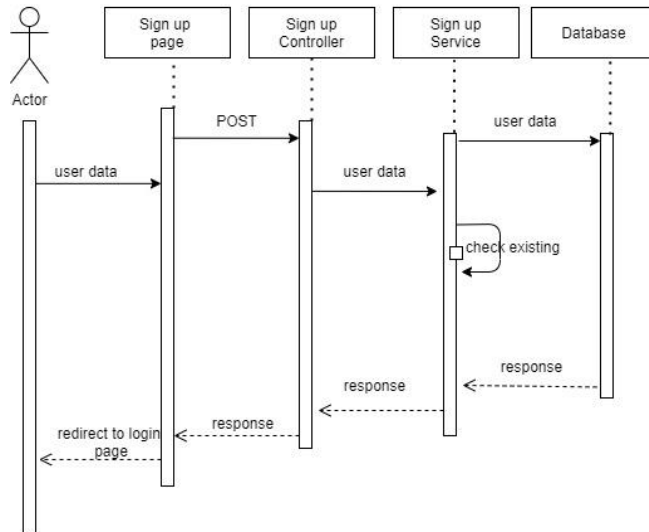
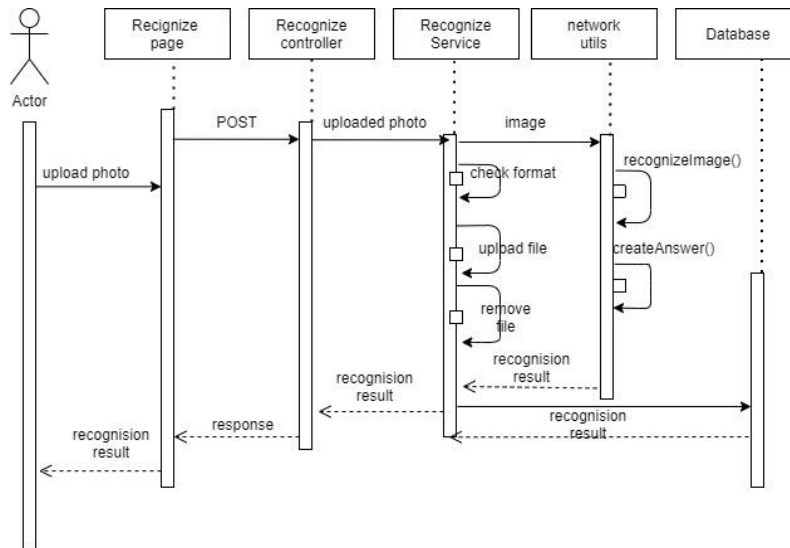
Великою перевагою такої архітектури є те що вона допомагає реалізувати інкапсуляцію а саме - заміні шару будуть порушені тільки ті шари, які працюють безпосередньо з зміненим. Обмежуючи залежності шарів один від одного, можна зменшити наслідки внесення змін, в результаті чого одиничне зміна не впливатиме на всі шари додатку.

А також архітектура надає можливість швидкої розробки та масштабованості.

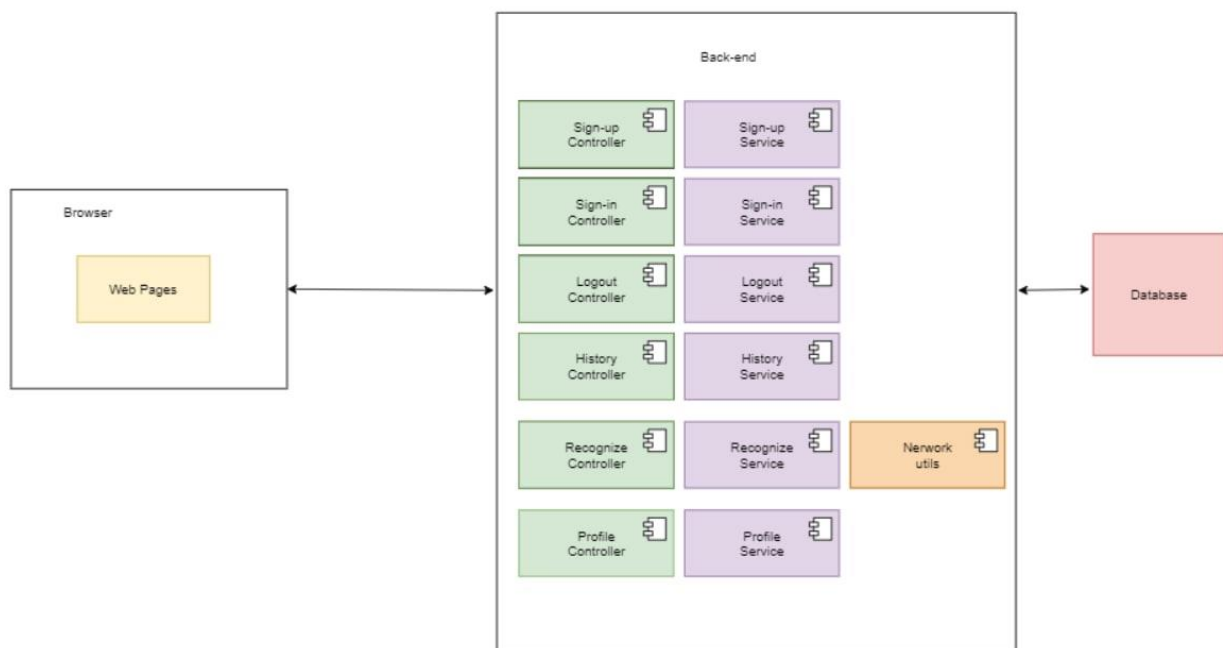
4.2.3 Діаграма класів



4.2.4 Діаграма послідовності



4.2.5 Діаграма компонентів



4.2.6 Специфікація функцій

Назва	Опис
@auth.route('/login') def login():	Контролер для рендерингу сторінки логіну
@auth.route('/login', methods=['POST']) def login_post():	Контролер, що обробляє форму логіну, отримує з форми логін та пароль користувача та передає його сервісу для перевірки
@auth.route('/signup') def signup():	Контролер для рендерингу сторінки реєстрації
@auth.route('/signup', methods=['POST']) def signup_post():	Контролер, що обробляє форму реєстрації, отримує данні користувача з форми та надсилає їх сервісу для обробки.
@auth.route('/logout') @login_required	Контролер, що виконує вихід

<code>def logout():</code>	з системи
<code>@main.route('/') def index():</code>	Контролер для рендиригу початкової сторінки застосунку
<code>@main.route('/profile') def profile():</code>	Контролер для рендиригу початкової сторінки профілю користувача, отримує дані користувача з сервісу .
<code>@main.route('/recognize', methods=['POST']) def recognize_post():</code>	Контролер для рендиригу сторінки розпізнавача.
<code>@main.route('/history') def history():</code>	Контролер, що отримує дані про минулі перевірки користувача і виводить їх на сторінку історії.
<code>def signup_service(email,name,password):</code>	Сервіс,що отримує дані з контролера реєстрації і обробляє сценарій реєстрацію користувача в системі, а саме – валідує введений пароль і логін і передає його для запису в базу даних.
<code>def login_service(email,password,remember):</code>	Сервіс,що отримує дані з контролера входу в систему і обробляє сценарій входу користувача в систему, а саме – порівнює введений логін та пароль з існуючими користувачами в системі.
<code>def recognize_service(uploaded_file):</code>	Сервіс що ,отримує зображення для розпізнавання від контролера і передає його методу для аналізу зображення на предмет наявності пневмонії.

def history_service(rows):	Сервіс,що отримує дані про попередні перевірки користувача.
def recognize_image():	Метод, що за допомогою збереженої навченої нейронної мережі розпізнає зображення.
def create_answer(y_pred):	Метод, що в залежності від ймовірності наявності пневмонії генерує відповідь для користувача.