

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ
СІКОРСЬКОГО»
(КП ім. Ігоря Сікорського)

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
кафедра БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

«До захисту допущено»
В.о. завідувач кафедри БМК

_____ Євген НАСТЕНКО

“ ___ ” _____ 2023р.

Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Комп'ютерні технології в біології та медицині»
спеціальності 122 «Комп'ютерні науки»

на тему: Мобільний додаток для класифікації зображень легень
на здорові та хворі на туберкульоз

Виконав: студент ІV курсу, групи БС-93

СЕРДЕЧЕНКО ІВАН СЕРГІЙОВИЧ

(прізвище, ім'я, по батькові)



(підпис)

Керівник: *проф. каф. біомедичної кібернетики (БМК)*
проф., д.м.н., КОВАЛЕНКО ОЛЕКСАНДР СЕРГІЙОВИЧ

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Консультант з розділів дипломної роботи:

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ім'я, по батькові)


(підпис)

Рецензент: *доцент кафедри біомедичної інженерії, к.т.н.*
Тарасова Лариса Дмитрівна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

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

Студент (-ка) _____
(підпис) 

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет біомедичної інженерії
Кафедра біомедичної кібернетики

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні технології в біології та медицині»

ЗАТВЕРДЖУЮ

В.о. завідувач кафедри БМК

_____ Євген НАСТЕНКО

« 30 » травня 2023 р.

ЗАВДАННЯ
на дипломну роботу студенту
СЕРДЕЧЕНКО ІВАН СЕРГІЙОВИЧ

(прізвище, ім'я, по батькові)

1. Тема роботи Мобільний додаток для класифікації зображень легень на здорові та хворі на туберкульоз

Керівник роботи

Коваленко Олександр Сергійович, професор, д.м.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «31» травня 2023 р. №2106-с

2. Термін подання студентом роботи

06-08 червня 2023р.

3. Вихідні дані до роботи: *згорткова нейронна мережа для класифікації рентгенівських знімків, аналіз зображення в android-додатку, історія аналізу зображень*

4. Зміст роботи: *створення нейронної мережі для класифікації зображень легень на здорові та хворі на туберкульоз, аналіз сучасних підходів до створення android застосунку, проектування та розробка програмного продукту для класифікації знімків комп'ютерної томографії легень за допомогою нейронної мережі*

5. Перелік ілюстративного матеріалу : 15 слайдів

6. Дата видачі завдання **30 травня 2023 року.**

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримати завдання за темою ДР на практику	До 15.02.2023р.	<i>виконано</i>
2	Переддипломна практика	За графіком	<i>виконано</i>
3	Виконання розділів ДР (Вступ, аналітичний огляд літературних джерел, теоретична частина)	До кінця практики	<i>виконано</i>
4	Виконання розділів ДР (практична частина, загальні висновки, список джерел)	Не пізніше 1 тижня до засідання каф-ри	<i>виконано</i>
5	Перевірка ДР науковим керівником	Не пізніше 1 тижня до засідання каф-ри	<i>виконано</i>
6	Подання в електронному вигляді ДР та анотації до неї на перевірку нормоконтролера та плагіат (UNICHECK).	---- « -----	<i>виконано</i>
7	Надання документів на засідання кафедри	За день до засідання	<i>виконано</i>
8	Предзахист ДР та допуск до захисту дисертації	Згідно плану каф.	<i>виконано</i>
9	Подання ДР рецензенту. Отримання рецензії.	До подання пакету документів до ЕК	<i>виконано</i>
10	Подання пакету документів по ДР та супровідних до неї документів до захисту в ЕК ¹	За 5 днів до дати захисту ДР за графіком	<i>виконано</i>
11	Захист ДР в ЕК		

Студент



(підпис)

Іван СЕРДЕЧЕНКО

(ім'я, ПРІЗВИЩЕ)

Керівник ДР

(підпис)

Олександр КОВАЛЕНКО

(ім'я, ПРІЗВИЩЕ)

Нормоконтролер

(підпис)

Галина КОРНІЄНКО

(ім'я, ПРІЗВИЩЕ)

¹ не пізніше ніж за 5 днів до затвердженої дати захисту ДР в ЕК

АНОТАЦІЯ

Дипломна робота за темою «Мобільний додаток для класифікації зображень легень на здорові та хворі на туберкульоз» виконана студентом кафедри біомедичної кібернетики ФБМІ *Сердеченко Іваном Сергійовичем* зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині» та складається зі: вступу; 3 розділів (*Аналітичний огляд літературних джерел, Теоретична частина, Практична частина*), висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 27 джерел. Загальний обсяг роботи 72 сторінок.

Мета і завдання роботи.

Метою роботи є покращення послуг за допомогою мобільного додатку для класифікації зображень легень на здорові та хворі на туберкульоз. Такий додаток надасть можливість швидкого та точного виявлення туберкульозу, що сприятиме ранньому діагностуванню та ефективному контролю цієї небезпечної хвороби.

Для досягнення мети, були поставлені наступні задачі:

1. Аналіз джерел.
2. Підготовка моделі класифікації.
3. Імплементация моделі.
4. Реалізація мобільного застосунку.

Використані методи. Використані методи включали підготовку зображень для класифікації та застосування глибинного навчання з використанням нейронних мереж. Було проведено тренування нейронної мережі на великому наборі зображень легень, що включали як здорові, так і заражені туберкульозом. Тренування включало в себе подачу зображень до мережі, вагову оптимізацію та налаштування параметрів, щоб забезпечити оптимальну класифікацію.

Отримані результати. Було реалізовано усі поставлені задачі, в тому числі підготовка моделі класифікації, її імплементация та реалізація мобільного застосунку для класифікації зображень легень на здорові та хворі на туберкульоз.

Публікації. За результатами виконаної дипломної роботи публікації не передбачаються.

Ключові слова. нейромережа, захворювання, комп'ютерна томографія, android, програмна реалізація, архітектура застосунку, класифікація.

Бібліографічний опис ДР

Сердеченко І. С. Мобільний додаток для класифікації зображень легень на здорові та хворі на туберкульоз : дипломна роб. бакалавра : 122 Комп'ютерні науки / Сердеченко Іван Сергійович. – Київ, 2023. – 72 с.

ABSTRACT

Bachelor's diploma thesis on the topic "Data Processing for Classification of X-ray Images of Healthy Lungs and Tuberculosis Patients" was carried out by Ivan Serdechenko, a student of the Department of Biomedical Cybernetics at the Faculty of Biomedical Engineering and Medical Informatics. The thesis is completed within the specialization 122 "Computer Science" under the educational-professional program "Computer Technologies in Biology and Medicine." It consists of the following sections: introduction, 3 chapters (Analytical Review of Literary Sources, Theoretical Part, Practical Part), conclusions for each of these chapters, general conclusions, a list of references comprising 27 sources. The total volume of the thesis is 72 pages.

Objective and tasks of the thesis.

The *objective* of the thesis is to improve services using a mobile application for classifying lung images into healthy and tuberculosis patients. Such an application will enable rapid and accurate detection of tuberculosis, which will contribute to early diagnosis and effective control of this dangerous disease. Its achievement involves solving the following *tasks*:

1. Analysis of sources.
2. Preparation of the classification model.
3. Implementation of the model.
4. Implementation of a mobile application.

Methods used. The methods used included preparing images for classification and applying deep learning using neural networks. A neural network was trained on a large set of lung images that included both healthy and tuberculosis patients. Training included feeding images to the network, weight optimization, and parameter tuning to ensure optimal classification.

Results obtained. All set tasks were implemented, including the preparation of a classification model, its implementation and the implementation of a mobile application for the classification of images of lungs in healthy and tuberculosis patients.

Publications. No publications are planned based on the results of this work.

Keywords: neural network, diseases, computer tomography, android, software implementation, application architecture, classification..

ЗМІСТ

<i>ВСТУП.....</i>	<i>10</i>
<i>РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....</i>	<i>13</i>
1.1. Аналіз літературних джерел.....	13
Висновки до розділу 1	18
<i>РОЗДІЛ 2 ТЕОРЕТИЧНА ЧАСТИНА.....</i>	<i>19</i>
2.1. Туберкульоз: діагностика та класифікація за допомогою рентгену 19	19
2.2. Нейронні мережі для класифікації зображень	20
2.2.1. Згортова нейронна мережа.....	21
2.2.2. Рекурентна нейронна мережа	22
2.3. Традиційні алгоритми машинного навчання.....	23
2.4. Переваги використання мобільних пристроїв в медицині.....	24
2.5. Актуальні тренди в розробці Android застосунків	26
2.6. Сучасні архітектури мобільних застосунків.....	28
2.6.1. Архітектура Model-View-Controller.....	28
2.6.2. Архітектура Model-View-Presenter	29
2.6.3. Архітектура Model-View-ViewModel	30
2.7. Clean Architecture, Dependency Injection та Reactive Programming.....	31
Висновки до розділу 2	31
<i>РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА</i>	<i>32</i>
3.1. Тренування нейронної мережі	33

3.2.	Результати тренування нейронної мережі	34
3.3.	Аналіз фреймворків для інтеграції нейромережі у android	
додаток	35	
3.3.1.	PyTorch	36
3.3.2.	Tensor Flow Lite	38
3.4.	Конвертація моделі нейромережі для використання у мобільному додатку	40
3.5.	Архітектура застосунку MVVM.....	41
3.5.1.	Model (Data Layer).....	42
3.5.2.	View (Presentation Layer).....	43
3.5.3.	ViewModel (Domain Layer)	44
3.5.4.	Repository	45
3.5.5.	Клас ViewModel.....	46
3.6.	Зберігання даних. SQLite та Room	48
3.7.	Material Design You	51
3.8.	Динамічні кольори	52
3.9.	Демонстрація роботи програмного застосунку	53
3.10.	Розрахунок економічного ефекту	67
	Висновки до розділу 3	68
	<i>ЗАГАЛЬНІ ВИСНОВКИ.....</i>	<i>69</i>
	<i>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</i>	<i>70</i>

ВСТУП

У сучасному світі здоров'я людини займає особливе місце в суспільстві. З розвитком технологій і наукових досягнень медицина набула значного прогресу в діагностиці та лікуванні різних захворювань. Рентген є одним з ключових методів обстеження, який дозволяє отримати детальні зображення внутрішніх органів та тканин людини. Проте, інтерпретація і аналіз цих зображень можуть бути складними та вимагати великого зусилля та досвіду медичних спеціалістів.

Туберкульоз є однією з найпоширеніших інфекційних хвороб у світі, зокрема в країнах з низьким рівнем розвитку та обмеженими ресурсами у галузі охорони здоров'я. За даними Всесвітньої організації охорони здоров'я, щорічно захворюють на туберкульоз мільйони людей, а близько 1.5 мільйона пацієнтів помирають від цієї хвороби. Туберкульоз також є однією з основних причин смерті серед людей, інфікованих вірусом імунодефіциту людини (ВІЛ).

Однією з проблем, пов'язаних з діагностикою туберкульозу, є складність та тривалість процесу виявлення хвороби. Існує потреба у відповідних методах та інструментах для ранньої та точної діагностики. Традиційні методи, такі як мікроскопія смуг або культури бактерій, мають свої обмеження в термінах точності, швидкості та вартості. Крім того, з'являються резистентні штами бактерій, які важко лікувати стандартними антибіотиками, ускладнюючи проблему лікування туберкульозу.

Застосування сучасних технологій машинного навчання та обробки зображень може значно полегшити діагностику та класифікацію туберкульозу. Розробка мобільного додатку для класифікації рентгенівських знімків здорових легень і легень, поразених туберкульозом, є інноваційним підходом, який може забезпечити швидке та надійне виявлення цієї хвороби. Такий додаток може стати ефективним інструментом для медичних працівників, особливо в областях з обмеженим доступом до спеціалізованих медичних закладів.

Досягнення у сфері медичних додатків та машинного навчання мають потенціал змінити підхід до діагностики та лікування туберкульозу. Ця дипломна

робота спрямована на розробку та реалізацію такого мобільного додатку, який забезпечить медичному персоналу надійні та точні засоби для класифікації та виявлення туберкульозу на рентген знімках. Це має велике значення для покращення контролю та управління цією небезпечною хворобою, а також для збереження життя та поліпшення якості життя пацієнтів, постраждалих від туберкульозу.

В процесі розробки додатку будуть використані передові методи машинного навчання та глибокого навчання, зокрема сверточні нейронні мережі, що дозволять навчити модель розпізнавати характерні ознаки, що вказують на наявність туберкульозу на рентгенівських знімках. Для досягнення максимальної точності та надійності класифікації будуть проведені етапи навчання моделі на великому наборі зображень, включаючи як здорові зразки, так і зразки з підтвердженим туберкульозом.

В результаті роботи ми отримаємо мобільний застосунок, який буде виявляти туберкульоз на знімках з високою точністю та швидкістю. Це дозволить медичному персоналу отримувати надійну підтримку у процесі діагностики туберкульозу та приймати обґрунтовані рішення щодо лікування пацієнтів. Крім того, розроблений додаток може бути використаний у місцях з обмеженим доступом до медичної діагностики, де необхідно оперативно виявляти туберкульоз та починати лікування.

Ця дипломна робота має велике значення, оскільки вона поєднує передові досягнення в області рентгену та машинного навчання з потребами медицини. Розроблений мобільний додаток може стати цінним інструментом для ранньої діагностики туберкульозу, покращення контролю цієї хвороби та збереження життя людей. Дана робота є важливим кроком у поліпшенні діагностики туберкульозу та внесенні внутрішніх змін у медичну практику.

Мета і завдання роботи

Метою роботи є покращення послуг за допомогою мобільного додатку для класифікації зображень легень на здорові та хворі на туберкульоз. Такий додаток

надасть можливість швидкого та точного виявлення туберкульозу, що сприятиме ранньому діагностуванню та ефективному контролю цієї небезпечної хвороби.

Для досягнення мети, були поставлені наступні задачі:

1. Аналіз джерел.
2. Підготовка моделі класифікації.
3. Імплементация моделі.
4. Реалізація мобільного застосунку.

Використані методи. Використані методи включали підготовку зображень для класифікації та застосування глибинного навчання з використанням нейронних мереж. Було проведено тренування нейронної мережі на великому наборі зображень легень, що включали як здорові, так і заражені туберкульозом. Тренування включало в себе подачу зображень до мережі, вагову оптимізацію та налаштування параметрів, щоб забезпечити оптимальну класифікацію.

Отримані результати. Було реалізовано усі поставлені задачі, в тому числі підготовка моделі класифікації, її імплементация та реалізація мобільного застосунку для класифікації зображень легень на здорові та хворі на туберкульоз.

Структура роботи

Дипломна робота за темою «Мобільний додаток для класифікації зображень легень на здорові та хворі на туберкульоз» виконана студентом кафедри біомедичної кібернетики ФБМІ *Сердеченко Іваном Сергійовичем* зі спеціальності 122 «Комп'ютерні науки» за освітньо-професійною програмою «Комп'ютерні технології в біології та медицині», побудована за класичним типом та викладена на 72 сторінках машинописного тексту. Вона складається зі: вступу; 3 розділів (*Аналітичний огляд літературних джерел, Теоретична частина, Практична частина*), висновків до кожного з цих розділів; загальних висновків; списку використаних джерел, який налічує 27 джерел. В роботі представлено 42 рисунка і 2 таблиці.

РОЗДІЛ 1

АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1.1. Аналіз літературних джерел

World Health Organization (WHO). (2021). Latent tuberculosis infection: updated and consolidated guidelines for programmatic management [1]

Цей документ, підготовлений Всесвітньою організацією охорони здоров'я, є оновленими і узагальненими рекомендаціями щодо програмного управління латентною формою туберкульозу. Оскільки латентна інфекція туберкульозом є важливим епідеміологічним фактором, її ефективне управління має вирішальне значення для зменшення поширення захворювання.

Документ надає вичерпний огляд сучасних знань про латентну інфекцію туберкульозом, включаючи опис принципів діагностики, оцінки ризику, лікування та моніторингу пацієнтів. Він також містить рекомендації щодо впровадження програм з управління латентною інфекцією в національні системи охорони здоров'я.

Документ наголошує на важливості індивідуального підходу до кожного пацієнта з латентною інфекцією туберкульозом, враховуючи ризики розвитку активного захворювання та користь від профілактичного лікування. Рекомендовані підходи базуються на наукових доказах та кращій клінічній практиці, з орієнтацією на забезпечення ефективності, безпеки та доступності лікування.

Оновлені і узагальнені рекомендації в цьому документі є цінним джерелом для медичних фахівців, дослідників та організацій, які займаються контролем туберкульозу. Вони допомагають покращити програми управління латентною інфекцією туберкульозом, сприяючи зниженню навантаження захворювання на суспільство та покращенню результатів лікування пацієнтів з туберкульозом.

Zenner, D., Beer, N., Harris, R. J., Lipman, M. C., & Stagg, H. R. (2020). Treatment of latent tuberculosis infection: An updated network meta-analysis. [2]

Джерело "Treatment of latent tuberculosis infection: An updated network meta-analysis" є статтею, опублікованою в журналі *Annals of Internal Medicine* у 2020 році. У цьому дослідженні автори здійснили оновлений мета-аналіз лікування латентної форми туберкульозу.

Основна мета цього дослідження полягала в порівнянні ефективності різних методів та режимів лікування латентної туберкульозної інфекції. Для досягнення цієї мети автори використали метод мережевого мета-аналізу, що дозволяє об'єднати дані з різних клінічних досліджень та зробити комплексне порівняння.

У статті розглядаються різні протоколи лікування, включаючи одиночні антибіотикові режими, комбіновані режими та протоколи з використанням ізоніазиду (Isoniazid) протягом тривалого періоду. Автори враховували ефективність лікування, побічні ефекти та ризики реактивації захворювання.

Результати дослідження свідчать про те, що комбіновані режими лікування, такі як режими з використанням ізоніазиду та рифампіцину (Rifampicin), показують найвищу ефективність у запобіганні розвитку активного туберкульозу. Також було виявлено, що тривалість лікування впливає на ефективність, і більш тривалі режими мають перевагу перед короткотерміновими.

Додатково, автори статті звернули увагу на побічні ефекти лікування та ризики розвитку реактивації захворювання. Вони розглянули дані щодо гепатотоксичності, невідповідності лікуванню та інших факторів, що можуть вплинути на успішність лікування латентної туберкульозної інфекції.

Узагальнюючи, стаття "Treatment of latent tuberculosis infection: An updated network meta-analysis" пропонує оновлену інформацію щодо лікування латентної форми туберкульозу. Результати мета-аналізу надають важливі вказівки щодо найефективніших методів лікування та режимів, а також допомагають усвідомити побічні ефекти та ризики, пов'язані з цими методами. Ця інформація може бути корисною при розробці стратегій лікування латентної туберкульозної інфекції та запобіганні розвитку активного захворювання.

Sreelakshmi, P. R., & Jayasankar, R. (2020). A review on computer-aided diagnosis of tuberculosis using chest radiographs. *Journal of Medical Imaging and Health Informatics* [3]

Основна мета цього огляду полягала в аналізі та узагальненні існуючих досліджень, що стосуються застосування комп'ютерної діагностики для виявлення туберкульозу на основі зображень рентгенівських знімків. Автори розглядали різні аспекти, такі як автоматичне виявлення патологічних ознак на знімках, класифікація зображень, оцінка стану захворювання та прогнозування розвитку хвороби.

У статті автори описують різні підходи, що використовуються для комп'ютерної діагностики туберкульозу. Це включає використання методів обробки зображень, штучного інтелекту та машинного навчання для автоматичного виявлення та класифікації знімків грудної клітки. Також розглядаються різні показники та ознаки, що використовуються для оцінки та прогнозування стану захворювання.

Результати огляду свідчать про потенційну ефективність комп'ютерної діагностики туберкульозу з використанням рентгенівських знімків. Автори виявили, що розроблені алгоритми та моделі здатні досягати високої точності в автоматичному виявленні та класифікації туберкульозу на знімках грудної клітки. Такі системи можуть бути корисними для швидкої та надійної діагностики, що допомагає вчасно розпочати лікування та покращити прогнози хвороби.

Отже, стаття "A review on computer-aided diagnosis of tuberculosis using chest radiographs" є цінним джерелом інформації, яке зосереджується на застосуванні комп'ютерної діагностики для виявлення туберкульозу на основі рентгенівських знімків. Вона надає огляд різних методів та підходів, що використовуються в цій галузі, а також вказує на потенційну ефективність таких систем в діагностиці та класифікації туберкульозу.

Jin, C., Chen, X., Wei, L., Liu, X., Zhang, Z., & Li, Y. (2020). Artificial intelligence accurately predicts tuberculosis in chest radiographs of HIV-negative and HIV-positive patients. [4]

У цій статті автори досліджують ефективність штучного інтелекту в точному прогнозуванні туберкульозу на рентгенівських зображеннях легень у пацієнтів зі ШІВ. Вони ставлять перед собою завдання розробити модель, яка б могла надати точний прогноз захворювання незалежно від наявності ВІЛ-інфекції.

Для досягнення цієї мети автори використовували набір даних, що містить рентгенівські зображення легень пацієнтів зі ШІВ, які були підтверджені наявністю або відсутністю туберкульозу. Вони використовували згорткові нейронні мережі для автоматичного виявлення ознак, характерних для туберкульозу, на зображеннях легень.

Результати дослідження показали, що запропонована модель штучного інтелекту досягла високої точності в передбаченні наявності туберкульозу на рентгенівських знімках легень пацієнтів зі ШІВ. Модель продемонструвала аналогічну ефективність як для ВІЛ-негативних, так і для ВІЛ-позитивних пацієнтів, що свідчить про її потенційну придатність для використання в клінічній практиці.

Ця стаття дає цінний внесок у розуміння можливостей використання нейронних мереж для прогнозування туберкульозу на рентгенівських зображеннях легень, зокрема у пацієнтів зі ШІВ. Дослідження надає підстави для подальшого розвитку та вдосконалення таких моделей з метою покращення діагностики та моніторингу захворювання.

TensorFlow. TensorFlow Lite: Deploy machine learning models on mobile and IoT devices. [5]

Джерело "TensorFlow Lite: Deploy machine learning models on mobile and IoT devices" є офіційною документацією TensorFlow Lite, яка надає детальну інформацію про використання TensorFlow Lite для розгортання моделей машинного навчання на мобільних та IoT-пристроях. Ця документація є

надзвичайно цінним ресурсом для розробників, які прагнуть імплементувати нейронну мережу класифікації зображень в андроїд-додатку на базі TensorFlow Lite.

Документація надає загальний огляд TensorFlow Lite, пояснюючи його призначення та переваги в контексті розгортання моделей машинного навчання на пристроях з обмеженими ресурсами. Документація надає докладні посібники щодо установки TensorFlow Lite на розробницьку машину та налаштування середовища розробки для андроїд-додатків. Вона також надає керівництва щодо інтеграції TensorFlow Lite у ваш додаток, включаючи завантаження та використання моделей TensorFlow Lite.

Крім того, документація охоплює керування моделями TensorFlow Lite, включаючи їх оптимізацію та розгортання на мобільних та IoT-пристроях. Вона також надає приклади коду та поради щодо ефективного використання TensorFlow Lite у вашому андроїд-додатку.

Ця документація є офіційним ресурсом TensorFlow Lite, що гарантує актуальну та достовірну інформацію щодо розгортання нейронних мереж класифікації зображень у ваших андроїд-додатках за допомогою TensorFlow Lite.

TensorFlow Lite Android Image Classification Example [6]

Дане джерело є прикладом реалізації класифікації зображень в андроїд-додатку з використанням TensorFlow Lite. Репозиторій містить вихідний код, який демонструє, як використовувати TensorFlow Lite для завантаження та виконання попередньо навченої моделі класифікації зображень на мобільному пристрої.

Аналізуючи це джерело, можна відокремити наступні аспекти:

1. Завантаження попередньо навченої моделі: Приклад надає інструкції про те, як завантажити попередньо навчену модель класифікації зображень, яка була попередньо навчена за допомогою TensorFlow.
2. Інтеграція TensorFlow Lite: Приклад показує, як інтегрувати TensorFlow Lite у ваш андроїд-додаток та як налаштувати середовище для виконання моделі.

3. Обробка зображень: Вихідний код надає приклади коду, як обробляти зображення перед передачею його до моделі, такі як зміна розміру, нормалізація та інші операції підготовки даних.
4. Виконання класифікації: Приклад демонструє, як передати оброблене зображення до моделі TensorFlow Lite та отримати результати класифікації, які показують, до якої категорії належить зображення.
5. Інтерфейс користувача: Джерело також надає приклад реалізації інтерфейсу користувача андроїд-додатку, що включає відображення зображення, результати класифікації та інші елементи інтерфейсу.

Висновки до розділу 1

В рамках аналітичного огляду літературних джерел були використані релевантні роботи та документи, що допомогли здійснити обґрунтовану підготовку до розробки додатку. Джерела включали офіційні документи Всесвітньої організації охорони здоров'я (WHO) щодо управління латентною формою туберкульозу, а також дослідження, які розглядалися питання терапії латентної форми туберкульозу, діагностики на основі зображень та використання штучного інтелекту для прогнозування туберкульозу на рентгенограмах грудної клітки. Крім того, використовувалася документація та приклади TensorFlow Lite для розгортання моделей машинного навчання на мобільних пристроях.

Аналіз цих джерел надав важливі теоретичні знання про управління латентною формою туберкульозу, методи діагностики на основі зображень і застосування штучного інтелекту для класифікації та прогнозування туберкульозу. Він також допоміг отримати уявлення про розробку та розгортання машинного навчання на мобільних пристроях з використанням TensorFlow Lite.

РОЗДІЛ 2

ТЕОРЕТИЧНА ЧАСТИНА

2.1. Туберкульоз: діагностика та класифікація за допомогою рентгену

Туберкульоз залишається одним з найбільш поширених та серйозних захворювань, що впливають на легені. (рис. 2.1) Він є причиною значної загибелі життів та проблем зі здоров'ям по всьому світу. За даними Всесвітньої організації охорони здоров'я, щорічно туберкульозом хворіють мільйони людей, а біля 1.5-2 мільйонів людей помирають від цієї хвороби щорічно.

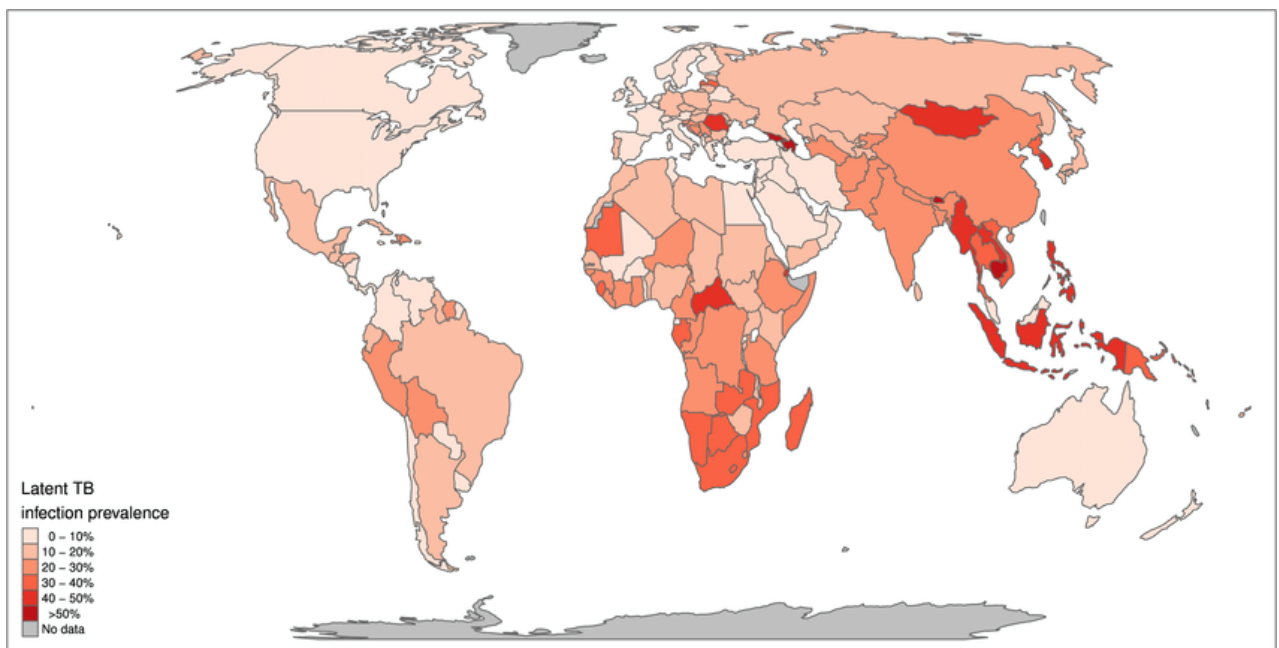


Рисунок 2.1 – Глобальна мапа поширеності латентної туберкульозної інфекції

Туберкульоз є інфекційним захворюванням, що спричинене бактерією *Mycobacterium tuberculosis*. Воно передається повітряно-крапельним шляхом, особливо під час кашлю, чхання або спілкування з хворою особою. Хоча легені є основними органами, які поразяються туберкульозом, хвороба може також впливати на інші органи, включаючи нирки, кістки, спинний мозок та лімфатичну систему.

Діагностика туберкульозу вимагає комплексного підходу. Традиційні методи включають рентгенографію легень, манту-тестування та мікроскопію слизу або культури бактерій. Однак ці методи мають свої обмеження, такі як недостатня точність, довгий час очікування результатів та потреба у висококваліфікованих медичних працівниках.

Останні роки характеризуються значними зусиллями у розробці машинного навчання та штучного інтелекту для діагностики туберкульозу. Використання комп'ютерного зору та алгоритмів машинного навчання дозволяє автоматизувати процес класифікації та виявлення хвороби на знімках рентгену. Такі підходи можуть допомогти в ранньому виявленні туберкульозу, покращуючи шанси на успішне лікування та зменшуючи ризик поширення хвороби. [7]

Проте, необхідно зазначити, що існують певні виклики і обмеження у використанні машинного навчання для класифікації туберкульозу на рентгенівських знімках. Наприклад, недостатня кількість анотованих зображень, які використовуються для тренування моделей, може вплинути на їх точність та універсальність. Крім того, важливо забезпечити конфіденційність та захист особистої інформації пацієнтів, коли використовуються зображення медичного характеру.

Отже, вирішення цих викликів та вдосконалення методів класифікації знімків для виявлення туберкульозу має велике значення для поліпшення діагностики та контролю цієї небезпечної хвороби. Розробка мобільного додатку, який використовує інноваційні підходи та алгоритми для класифікації знімків рентгену, може сприяти покращенню ефективності, швидкості та доступності діагностики туберкульозу.

2.2. Нейронні мережі для класифікації зображень

Нейронні мережі є потужним інструментом у сфері медичного зображення та класифікації рентген зображень легень. Вони дозволяють автоматично

виявляти та класифікувати патологічні зміни з високою точністю та швидкістю.

2.2.1. Згорткова нейронна мережа

Одним із найпопулярніших типів нейронних мереж, які застосовуються для класифікації зображень, є згорткові нейронні мережі (Convolutional Neural Networks, CNN) (рис. 2.2). Вони виявляють локальні особливості зображення шляхом застосування фільтрів, що допомагає розпізнавати унікальні ознаки хвороби на рентгенівських знімках легень. CNN мають велику гнучкість у розпізнаванні широкого спектру патологічних змін, таких як внутрішньолегеві вузли, кавернозні утворення та інші аномалії. [8]

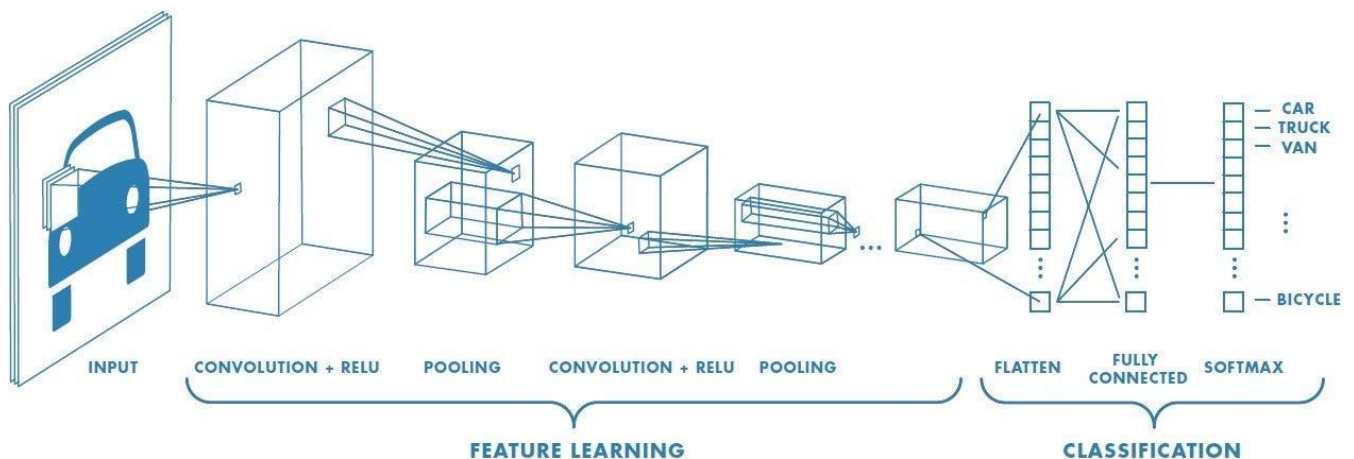


Рисунок 2.2 – Згорткова нейронна мережа

Використання CNN в медичній області дозволяє досягти високої точності і швидкості обробки медичних зображень.

Згорткові нейронні мережі використовуються для автоматичного виявлення та аналізу різних патологічних змін на зображеннях легень. Вони можуть визначати наявність аномалій, таких як вогнища пневмонії, туберкульозу, опухолей або інших захворювань. [9]

Основними перевагами використання згорткових нейронних мереж є їх здатність до автоматичного вивчення корисних ознак на зображеннях, безпосередньо з розпізнаних даних. Це означає, що вони можуть ефективно працювати з реальними зображеннями легень і навчатися визначати характерні риси, що свідчать про наявність патології.

Багато досліджень було проведено для розвитку та оптимізації згортових нейронних мереж для класифікації медичних зображень. Різні архітектури, такі як VGG, ResNet, Inception та інші, були успішно використані для розв'язання задач медичної класифікації.

2.2.2. Рекурентна нейронна мережа

Поряд зі згортковими нейронними мережами, рекурентні нейронні мережі (Recurrent Neural Networks, RNN) використовуються для аналізу послідовних даних, таких як часові серії зображень (рис 2.3). Вони можуть виявляти та моделювати залежності між сусідніми зображеннями, що дозволяє враховувати динаміку змін у легенях з плином часу. RNN також ефективно використовуються для виявлення розсіяних патологічних змін та моніторингу прогресу хвороби.

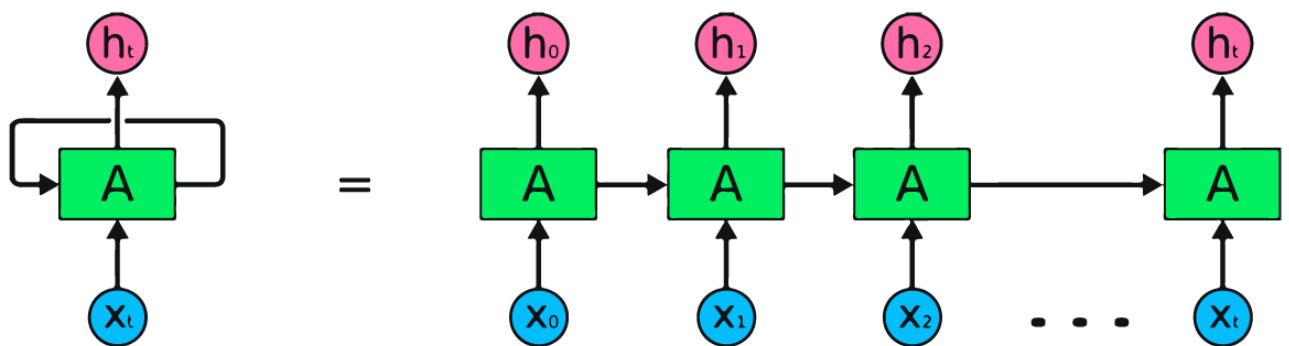


Рисунок 2.3 – Рекурентна нейронна мережа

Окрім цього, розробка глибоких нейронних мереж (Deep Neural Networks, DNN) та генеративних моделей, таких як автоенкодерів, показала високу ефективність у виявленні та класифікації рентген зображень легень. Вони дозволяють враховувати складні структури та залежності у зображеннях, що покращує точність та надійність класифікації. [10]

Проте, варто зазначити, що нейронні мережі вимагають великого обсягу анотованих даних для тренування та налагодження параметрів. Також важливо враховувати обмеження обробки зображень на мобільних пристроях, таких як обмежена потужність обчислення та обсяг пам'яті. Тому вибір та оптимізація

відповідної архітектури нейронної мережі для мобільного додатку є важливим аспектом розробки.

Отже, з використанням нейронних мереж для класифікації зображень легень пов'язані як перспективи, так і виклики. Враховуючи досягнення у цій галузі та потенціал мобільних додатків у медицині, розробка мобільного додатку для класифікації рентгенівських знімків легень з використанням нейронних мереж може стати важливим кроком у поліпшенні діагностики та контролю туберкульозу.

2.3. Традиційні алгоритми машинного навчання

Однією з альтернативних методик є традиційні алгоритми машинного навчання, такі як метод опорних векторів (Support Vector Machines, SVM), рішітки (Random Forests), наївний Баєсівський класифікатор та інші. Ці методи засновані на визначенні індивідуальних ознак зображень та побудові моделі для класифікації. Вони використовують статистичні алгоритми для розрізнення між здоровими і хворими зразками. [11]

Однак, нейронні мережі виявляються більш потужними і гнучкими у розв'язанні завдань класифікації зображень. Вони можуть автоматично вивчати репрезентативні ознаки зображень шляхом згорток та пулінгу, а також здатні адаптуватися до різноманітних даних та вирішувати складні завдання.

Переваги використання нейронних мереж включають:

1. Здатність до автоматичного вивчення репрезентативних ознак без необхідності ручного визначення параметрів алгоритму.
2. Гнучкість та масштабованість моделей, що дозволяє працювати з різними розмірами та типами зображень.
3. Здатність до вирішення складних завдань класифікації, включаючи виявлення малих аномалій або патологічних змін на зображеннях.

Нейронні мережі також можуть досягати високої точності в класифікації зображень легень за умови належного навчання та оптимізації моделі. Їхній

потенціал виявляється в розпізнаванні складних структур та патологічних ознак, що можуть бути важкодоступні для традиційних алгоритмів машинного навчання.

Отже, нейронні мережі виявляються перевагою у порівнянні з альтернативними методами класифікації рентгенівських зображень легень, завдяки їхній здатності до автоматичного вивчення репрезентативних ознак, гнучкості та здатності до розв'язання складних завдань.

2.4. Переваги використання мобільних пристроїв в медицині

Використання мобільних пристроїв у медичній сфері має декілька переваг, які роблять їх привабливими для розробки додатків з класифікації зображень легень.

По-перше, мобільні пристрої, такі як смартфони і планшети, є широко поширеними та доступними для багатьох людей. Це підтверджують світові продажі (рис. 2.4).

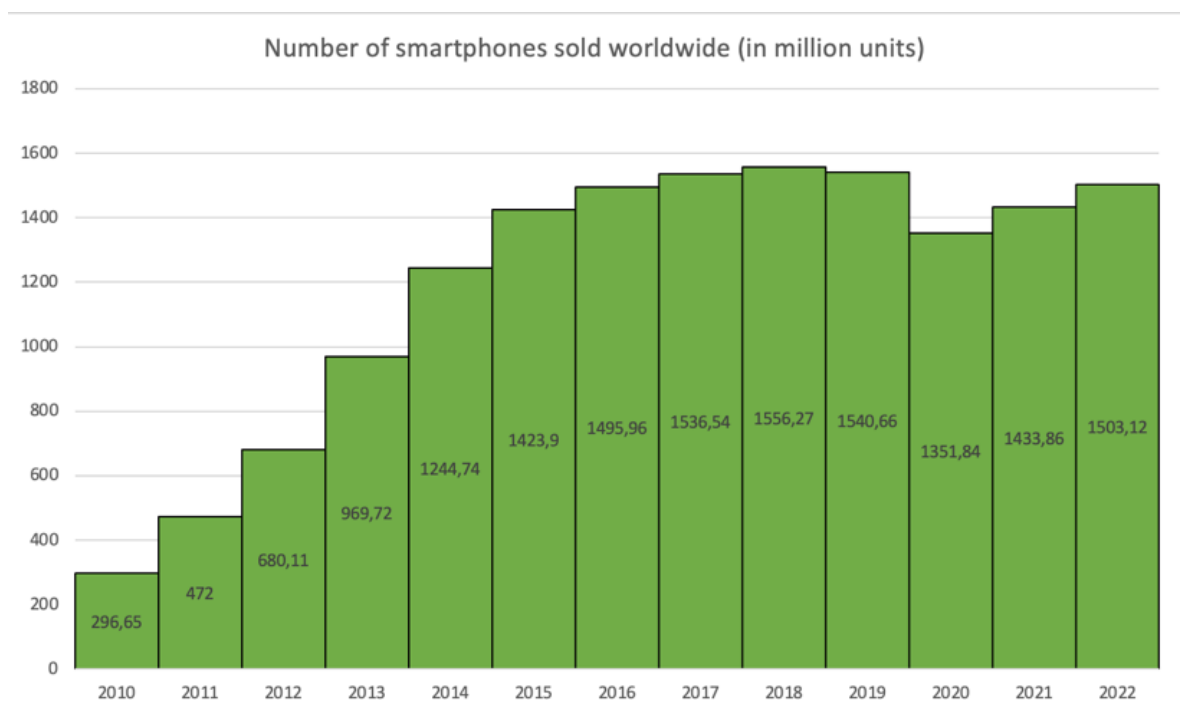


Рисунок 2.4 - Світові продажі смартфонів

Вони вже використовуються в повсякденному житті, що робить їх зручними для впровадження мобільних додатків з медичною функціональністю. Мобільний додаток для класифікації рентгенівських зображень легень може забезпечити швидкий та зручний доступ до результатів діагностики навіть за межами медичного закладу. В той же час, продажі комп'ютерів з року в рік стрімко падають (рис. 2.5), що каже про те, що їх час спливає, мобільні девайси захоплюють світ.

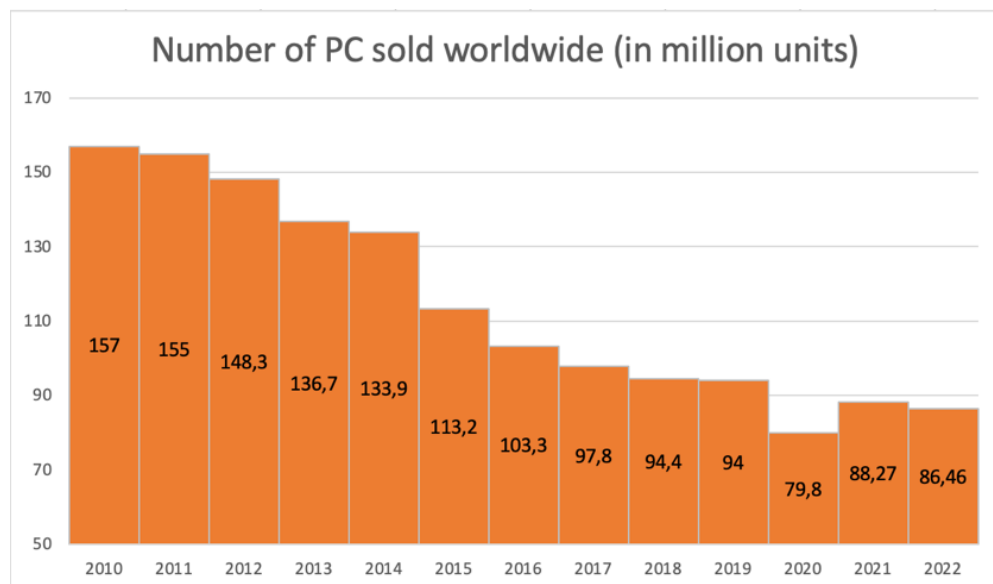


Рисунок 2.5 - Продажі комп'ютерів

По-друге, мобільні пристрої мають потужність обчислення та графічні можливості, які зростають з кожним поколінням. Це дозволяє виконувати складні алгоритми обробки зображень та нейронні мережі прямо на самому пристрої, забезпечуючи швидку та ефективну роботу мобільного додатку. Враховуючи специфіку класифікації рентген зображень легень, де точність і швидкість є важливими факторами, використання мобільного пристрою може бути оптимальним варіантом. [12]

Незважаючи на переваги, варто розглянути і альтернативні підходи до розробки додатків для класифікації зображень легень. Одним з таких підходів є використання хмарних обчислень, де обробка та аналіз зображень здійснюється на віддалених серверах. Це може забезпечити більшу потужність обчислення та

доступ до великого обсягу даних, але може виникати затримка у передачі та обробці даних. Крім того, залежність від Інтернет-з'єднання може створювати обмеження щодо доступності та надійності роботи додатку.

Ураховуючи переваги мобільних пристроїв та специфіку задачі класифікації зображень легень, використання мобільного додатку з нейронними мережами на самому пристрої може бути ефективним та зручним рішенням. Враховуючи технологічний прогрес та широке поширення мобільних пристроїв, такий підхід може сприяти поліпшенню діагностики та контролю за туберкульозом.

2.5. Актуальні тренди в розробці Android застосунків

За останні роки мобільні технології зазнали значних змін та вдосконалень, що призвело до з'явлення нових можливостей та трендів у розробці андроїд додатків.

Прогресивний розвиток платформи Android привів до зростання функціональності додатків. Введення нових версій операційної системи Android сприяло вдосконаленню інтерфейсу користувача, забезпеченню безпеки, оптимізації продуктивності та підтримці передових технологій.

Для зручності розробки та налагодження андроїд додатків розробники використовують спеціальне інтегроване середовище розробки (IDE) - Android Studio. Це потужний інструмент, який надає розробникам багатофункціональне середовище для розробки, тестування та розгортання додатків.

Сучасні технології, що використовуються для розробки андроїд додатків, включають розробку на мовах програмування Java та Kotlin. Java є традиційним і широко використовуваним мовою, тоді як Kotlin набуває більшої популярності завдяки своїй простоті та функціональним можливостям (рис 2.6). [13]



Рисунок 2.6 – Порівняння Kotlin та Java

Додатковою складовою сучасних технологій андроїд розробки є використання API (Application Programming Interface) та SDK (Software Development Kit). Ці інструменти надають доступ до різноманітних функцій пристроїв, таких як камера, геолокація, датчики, мультимедіа, що дозволяє створювати розширені та інтерактивні додатки.

Крім того, важливим трендом в розробці андроїд додатків є інтеграція з хмарними сервісами. Хмарні технології дозволяють зберігати дані та ресурси на віддалених серверах, що сприяє збільшенню масштабованості, доступності та зручності роботи з додатками.

Усі ці технологічні та інноваційні рішення сприяють постійному розвитку та удосконаленню андроїд додатків. Розробники продовжують досліджувати нові можливості та пристосовуватися до змінних потреб користувачів, забезпечуючи високоякісні, функціональні та інноваційні мобільні додатки. [14]

2.6. Сучасні архітектури мобільних застосунків

Розвиток мобільних технологій та зростання популярності платформи Android спонукають розробників до використання оптимальних підходів у процесі створення додатків.

Сучасні архітектури надають можливість розробникам створювати додатки зі стабільною архітектурою, розширюваністю та простотою тестування. Архітектурні підходи, такі як MVC, MVP, MVVM та Clean Architecture, визначають чіткий розподіл відповідальностей та структуру проекту, що полегшує розробку та підтримку коду. [15]

Загалом, розуміння актуальних архітектур та технологій розробки андроїд додатків є важливим аспектом для успішної реалізації проектів у сфері мобільних технологій. Ці підходи та інструменти дозволяють розробникам створювати додатки, які відповідають вимогам ринку та надають зручний та ефективний користувацький досвід. [16]

Ці архітектури надають структурований підхід до розробки додатків, сприяють підтримці розширюваності, тестування та підтримуваності коду.

2.6.1. Архітектура Model-View-Controller

Одна з найпопулярніших архітектур, яка використовується в андроїд додатках, - це Model-View-Controller (MVC) (рис 2.7). У цій архітектурі модель відповідає за управління даними та бізнес-логікою, представлення (view) - за відображенням даних користувачеві, а контролер - за обробку вхідних подій та координацію роботи моделі та представлення. [17]

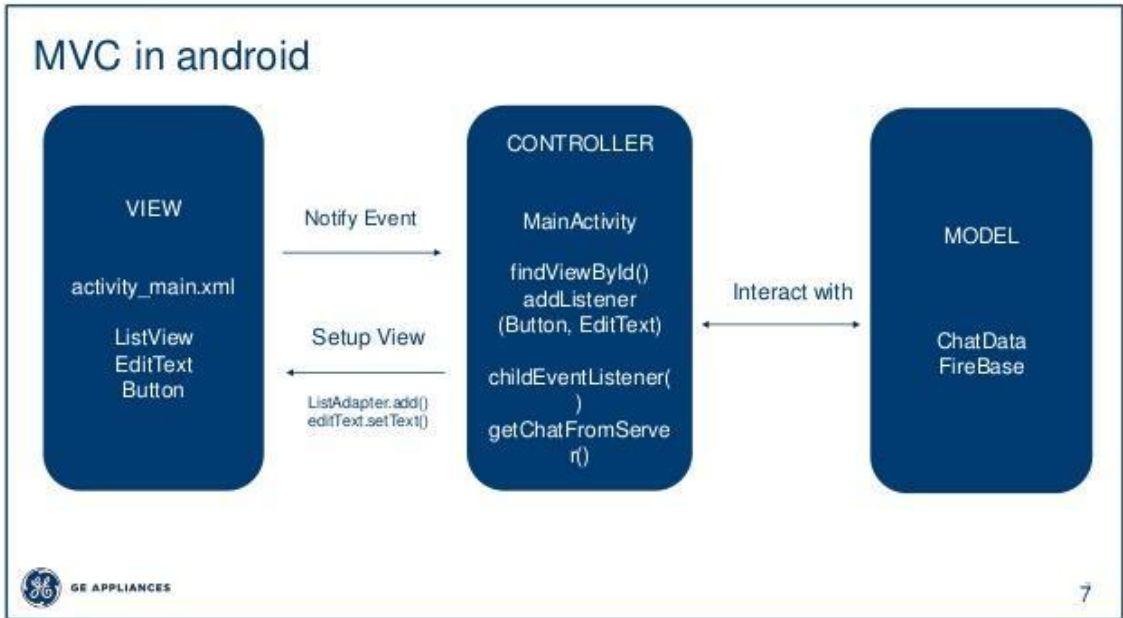


Рисунок 2.7 - Архітектура Model-View-Controller

2.6.2. Архітектура Model-View-Presenter

Іншою популярною архітектурою є Model-View-Presenter (MVP) (рис. 2.8). У MVP модель виконує ті ж функції, що і в MVC, але презентер відповідає за управління взаємодією між моделлю та представленням, а також за логіку взаємодії з користувачем. Представлення відповідає за відображення даних та обробку подій користувача. [18]

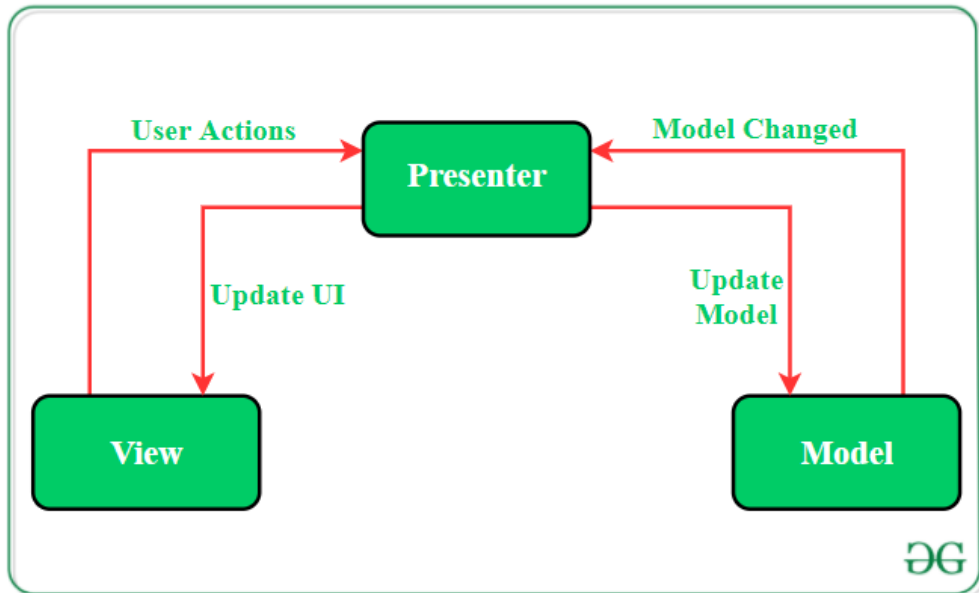


Рисунок 2.8 - Архітектура Model-View-Presenter

2.6.3. Архітектура Model-View-ViewModel

Останнім часом набуває популярності архітектура Model-View-ViewModel (MVVM) (рис 2.9). У цій архітектурі модель відповідає за управління даними, представлення відображає ці дані та обробляє події користувача, а ViewModel виконує роль посередника між моделлю та представленням, забезпечуючи зв'язок та обмін даними між ними. MVVM є популярним шаблоном розробки для платформи Android, оскільки він забезпечує хорошу організацію коду, полегшує розширення додатків та покращує тестованість. Використання MVVM може підвищити ефективність розробки та підтримку довгострокової розширюваності вашого Android-додатку. [19]

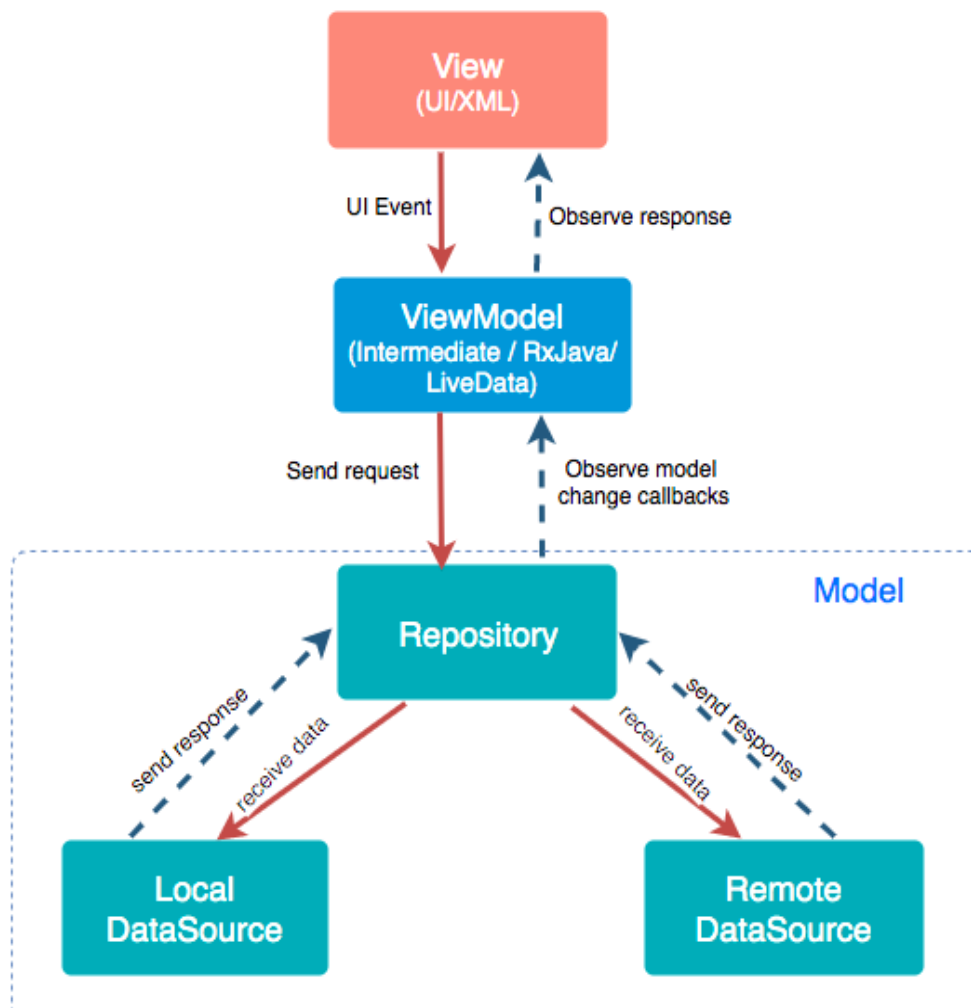


Рисунок 2.9 - Архітектура Model-View-ViewModel

2.7. Clean Architecture, Dependency Injection та Reactive Programming

Також варто зазначити архітектуру Clean Architecture, яка розроблена з орієнтацією на підтримуваність, тестування та незалежність від фреймворків. Вона використовує принципи "зовнішній світ не повинен залежати від внутрішнього" та "абстракція повинна залежати від деталей".

Окрім цих архітектур, в андроїд додатках також використовуються різноманітні підходи, як-от Dependency Injection (DI), Reactive Programming (RxJava, LiveData), а також використання різних бібліотек та фреймворків, які спрощують розробку та забезпечують високу продуктивність. [20]

Актуальні архітектури андроїд додатків розвиваються швидкими темпами, адаптуючись до нових вимог ринку та технологічних зрушень. Розробники постійно вдосконалюють та впроваджують нові підходи, щоб створювати ефективні та масштабовані додатки для платформи Android.

Висновки до розділу 2

У цьому розділі були розглянуті та описані різні важливі теми, які становлять основу для подальшої розробки мобільного додатку для класифікації рентгенівських зображень. Було розглянуто питання туберкульозу, нейронних мереж, включаючи згорткові та рекурентні нейронні мережі, альтернативні методи класифікації, використання мобільних пристроїв у медицині, розвиток операційної системи Android та сучасні архітектури мобільних застосунків, зокрема архітектури Model-View-Controller, Model-View-Presenter і Model-View-ViewModel. Крім того, були висвітлені інші важливі патерни в розробці Android додатків. Ця інформація буде використана для подальшої розробки додатку та реалізації його функціональності.

РОЗДІЛ 3

ПРАКТИЧНА ЧАСТИНА

Для розробки нашого мобільного додатку з класифікацією знімків рентгену здорових легень і хворих на туберкульоз, нам було необхідно вибрати відповідну нейронну мережу. При виборі ми врахували кілька факторів, таких як точність, швидкість, архітектура та можливість оптимізації для мобільних пристроїв.

Після аналізу різних нейронних мереж, включаючи відомі архітектури, такі як VGG, ResNet і Inception, ми ретельно вивчили їх потужність та ефективність у вирішенні задач класифікації зображень легень. На жаль, після оцінки їхніх результатів і порівняння з нашими вимогами, ми прийшли до висновку, що жодна з існуючих предобучених моделей не відповідає нашим потребам щодо точності та продуктивності.

Тому, з метою досягнення кращих результатів, ми вирішили розробити власну нейронну мережу засновану на архітектурі Convolutional Neural Network (CNN). Використання CNN є основоположним для класифікації зображень, оскільки вона вміє ефективно розпізнавати образи та впоратися зі складними структурами даних та включає ряд інноваційних підходів для досягнення максимальної точності та ефективності.

Це забезпечує нам відмінну базову модель для нашої задачі класифікації зображень легень .

Основною метою нашого дослідження було створення мобільного додатку, який забезпечує швидку та надійну класифікацію знімків, що дозволяє вчасно виявляти патологічні зміни та сприяє ефективній діагностиці туберкульозу. Для досягнення цієї мети ми розробили та натренували власну нейронну мережу з використанням набору даних, який містить зображення здорових легень та хворих на туберкульоз.

3.1. Тренування нейронної мережі

Для тренування нашої нейронної мережі ми використали публічний датасет з Kaggle, який містив рентгенівські зображення здорових легень і хворих на туберкульоз (рис. 3.1).



Рисунок 3.1 – Приклад зображення з датасету

Датасет мав роздільну здатність 512x512 пікселів, але ми зменшили його до 250x250 пікселів та перетворили у формат Grayscale для більшої ефективності обробки даних (рис 3.2).



Рисунок 3.2 – Приклад обробленого зображення

Перед тренуванням мережі, ми розділили датасет на тренувальну, валідаційну та тестову вибірки. Тренувальна вибірка складалась з 80% зображень, валідаційна - з 10%, а тестова - з 10%. Ми використали методи аугментації даних, такі як зміщення, обертання та збільшення/зменшення масштабу зображень, щоб розширити обсяг тренувального набору та запобігти перенавчанню.

Нейронну мережу було треновано протягом 3 епох з пакетом розміром 64 зображення. Після кожної епохи ми оцінювали точність на валідаційній вибірці, щоб контролювати процес тренування та уникнути перенавчання.

3.2. Результати тренування нейронної мережі

Після завершення тренування ми оцінили ефективність нейронної мережі на тестовій вибірці. Отримана точність моделі становить 96% (рис. 3.3), що свідчить про її високу здатність до класифікації рентгенівських знімків здорових легень і хворих на туберкульоз.

```
Epoch 1/3
99/99 [=====] - 556s 6s/step - loss: 0.3530 - accuracy: 0.8642 - val_loss: 0.1637 - val_accuracy: 0.9285
Epoch 2/3
99/99 [=====] - 536s 5s/step - loss: 0.1365 - accuracy: 0.9470 - val_loss: 0.1007 - val_accuracy: 0.9628
Epoch 3/3
99/99 [=====] - 518s 5s/step - loss: 0.1365 - accuracy: 0.9489 - val_loss: 0.1188 - val_accuracy: 0.9600
```

Рисунок 3.3 - Результати тренування нейронної мережі

Ці результати є обнадійливими і свідчать про потенційну ефективність нашого мобільного додатку у діагностиці туберкульозу. Висока точність класифікації (рис 3.4) та низький рівень втрати (рис. 3.5) може допомогти медичним працівникам у швидкій та надійній ідентифікації патологічних змін на рентген зображеннях.

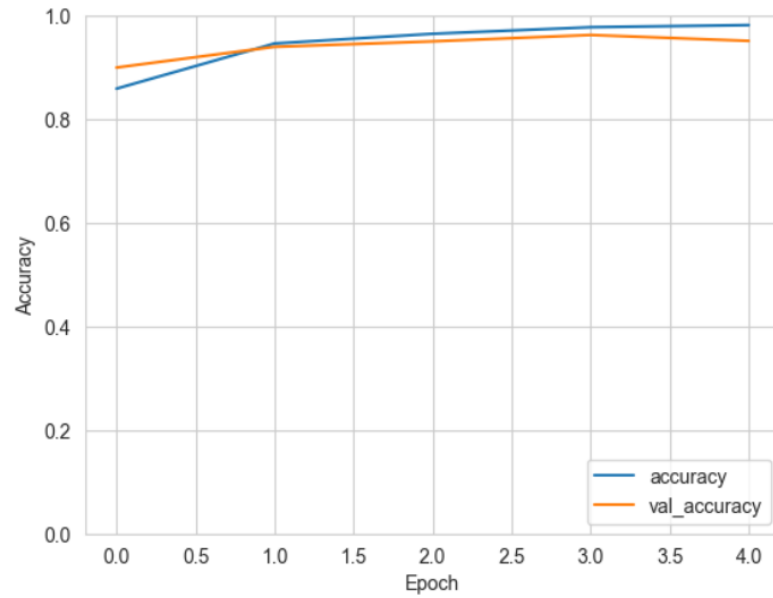


Рисунок 3.4 - Графік функції якості

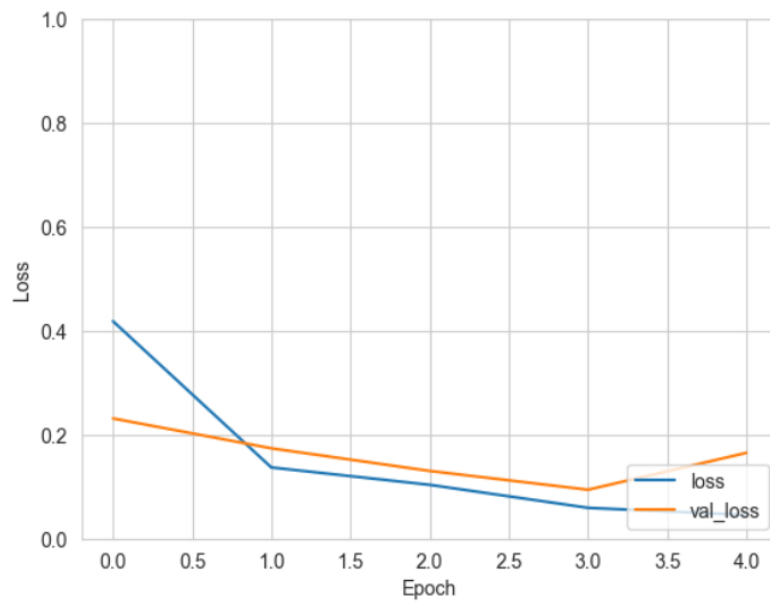


Рисунок 3.5 - Графік функції витрати

3.3. Аналіз фреймворків для інтеграції нейромережі у android додаток

Вибір підходящого фреймворку є важливим етапом розробки, оскільки він визначає продуктивність, функціональність та зручність роботи з моделлю на мобільних пристроях. [21]

Під час аналізу різних варіантів фреймворків для андроїд додатків, ми врахували кілька ключових критеріїв:

- Підтримка нейронних мереж: Ми шукали фреймворки, які забезпечують підтримку роботи з нейронними мережами і здатні ефективно працювати з великими моделями.
- Швидкість та продуктивність: Важливо, щоб фреймворк був оптимізований для мобільних пристроїв та забезпечував швидку обробку даних.
- Легкість використання: Ми звернули увагу на інтерфейс та документацію фреймворків, оцінили, наскільки легко навчитися та використовувати їх у розробці.

3.3.1. PyTorch

PyTorch є популярним фреймворком глибинного навчання, розробленим на мові програмування Python, але також має підтримку для мобільної платформи Android.

Завдяки можливості розгортання PyTorch моделей на Android, розробники можуть створювати мобільні додатки, які використовують нейронні мережі для класифікації зображень легень, включаючи розрізнення між здоровими та хворими на туберкульоз. [22]

Ось кілька важливих переваг PyTorch:

1. Простота використання та легкість навчання: PyTorch має інтуїтивний та досить простий для розуміння інтерфейс. Його синтаксис дуже схожий на звичайні математичні вирази, що полегшує розробку та експериментування з моделями глибинного навчання. Багато розробників вважають, що PyTorch є більш легким у вивченні порівняно з іншими фреймворками глибинного навчання.
2. Гнучкість та експериментування: PyTorch надає розробникам велику свободу в експериментуванні з різними архітектурами та підходами до глибинного навчання. Це дозволяє розробникам швидко прототипувати

та змінювати моделі, відлагоджувати їх та оптимізувати. Підтримка динамічного обчислення у PyTorch також робить його гнучким для виконання складних операцій та адаптації моделей під конкретні вимоги проекту.

3. Активна спільнота та ресурси: PyTorch має велику та активну спільноту розробників, яка постійно актуалізує документацію, надає підтримку та допомогу. Існує безліч навчальних матеріалів, статей, блогів, відеоуроків та прикладів коду, що полегшує вивчення та розвиток навичок у роботі з фреймворком. Завдяки активній спільноті, розробникам легше знайти відповіді на свої запитання та отримати рекомендації щодо оптимальних практик.

Незважаючи на переваги, PyTorch також має кілька недоліків при розробці мобільних додатків на Android:

1. Великий розмір моделей: Одним з недоліків PyTorch є те, що моделі глибинного навчання можуть мати великий розмір. Це може створювати проблеми з пам'яттю на мобільних пристроях, особливо з обмеженим обсягом доступної пам'яті. При розгортанні моделей PyTorch на Android важливо забезпечити оптимальну компресію та оптимізацію моделей для забезпечення ефективної роботи на мобільних пристроях.
2. Обмежені ресурси мобільних пристроїв: Мобільні пристрої мають обмежені обчислювальні ресурси та обмежену тривалість роботи від акумулятора. Використання глибоких моделей вимагає значних обчислювальних ресурсів та енергії. Для забезпечення оптимальної продуктивності на мобільних пристроях, може бути необхідно провести додаткову оптимізацію, наприклад, використовувати квантизацію або моделі з низькою розширеністю.
3. Інтеграція з Android: При інтеграції PyTorch моделей з мобільним додатком на Android можуть виникати деякі виклики. Наприклад, необхідно забезпечити сумісність версій PyTorch з Android, обробку

даних в реальному часі, інтеграцію з іншими компонентами додатка та оптимізацію продуктивності для мобільної платформи.

Загалом, PyTorch є потужним фреймворком глибинного навчання, який може бути використаний для розробки мобільного додатку на Android з класифікацією зображень легень.

Проте, перед розгортанням моделі PyTorch на мобільних пристроях, важливо ретельно планувати його оптимізацію, щоб забезпечити ефективну роботу, мінімізувати використання ресурсів та забезпечити гладку інтеграцію з Android-платформою.

3.3.2. Tensor Flow Lite

У сучасному світі зростає популярність мобільних пристроїв, які є невід'ємною частиною нашого повсякденного життя. Разом з цим зростає і попит на мобільні додатки, здатні виконувати складні завдання, включаючи машинне навчання. Щоб задовольнити цей попит, необхідно мати ефективні та оптимізовані рішення для розгортання моделей машинного навчання на мобільних пристроях.

TensorFlow Lite - це легка версія популярного фреймворку TensorFlow, спеціально розроблена для роботи на мобільних та вбудованих пристроях. Вона забезпечує оптимізовану виконавчу середу та інструменти для конвертації та розгортання нейронних мереж на мобільних пристроях.

Він забезпечує широкі можливості розгортання оптимізованих моделей, що працюють швидко та ефективно навіть на пристроях з обмеженими обчислювальними ресурсами. [23]

TensorFlow Lite має кілька переваг, які роблять його привабливим вибором для реалізації моделей машинного навчання в андроїд додатках:

- **Ефективність:** Фреймворк забезпечує оптимізовану виконавчу середу, що дозволяє швидко та ефективно виконувати моделі машинного навчання на мобільних пристроях з обмеженими обчислювальними ресурсами.

- Низькі вимоги до ресурсів: TensorFlow Lite використовує мінімальну кількість пам'яті та обчислювальних ресурсів, що дозволяє ефективно використовувати його на пристроях з обмеженими характеристиками.
- Підтримка апаратного прискорення: TensorFlow Lite може використовувати апаратне прискорення на мобільних пристроях, таке як графічні процесори, для прискорення обчислень. Це дозволяє отримати ще більшу продуктивність та швидкість виконання моделей.
- Широкий спектр підтримуваних моделей: TensorFlow Lite підтримує різні типи моделей, включаючи глибокі нейронні мережі, лінійні моделі та інші. Це дає можливість використовувати різноманітні моделі залежно від потреб проекту.

Загальне поєднання цих переваг робить TensorFlow Lite привабливим інструментом для реалізації моделей машинного навчання в андроїд додатках. Його легка інтеграція, ефективність та оптимізована виконавча середовище роблять його ідеальним вибором для нашого проекту.

Результат порівняння PyTorch та TensorFlow Lite представлено в табл. 3.1

Таблиця 3.1

Переваги та недоліки TensorFlow Lite та PyTorch

	TensorFlow Lite	PyTorch
Переваги	Підтримка розгортання моделей Висока продуктивність Широкий спектр функціональності	Легкість використання та навчання Гнучкість та модульність Спільнота та екосистема
Недоліки	Складніше вивчення Обмежена гнучкість	Вищі вимоги до ресурсів Підтримка розгортання

Після ретельного аналізу різних варіантів, ми прийняли рішення використовувати TensorFlow Lite в якості фреймворку для імплементування нашої нейронної мережі в андроїд додаток.

3.4. Конвертація моделі нейромережі для використання у мобільному додатку

Конвертація моделі з TensorFlow до TensorFlow Lite (рис 3.6) є важливим етапом у розробці мобільних додатків для машинного навчання. TensorFlow Lite - це легковага бібліотека, оптимізована для роботи на мобільних пристроях та вбудованих системах. Ця бібліотека дозволяє виконувати навчені моделі швидко та ефективно без значного впливу на ресурси пристрою.

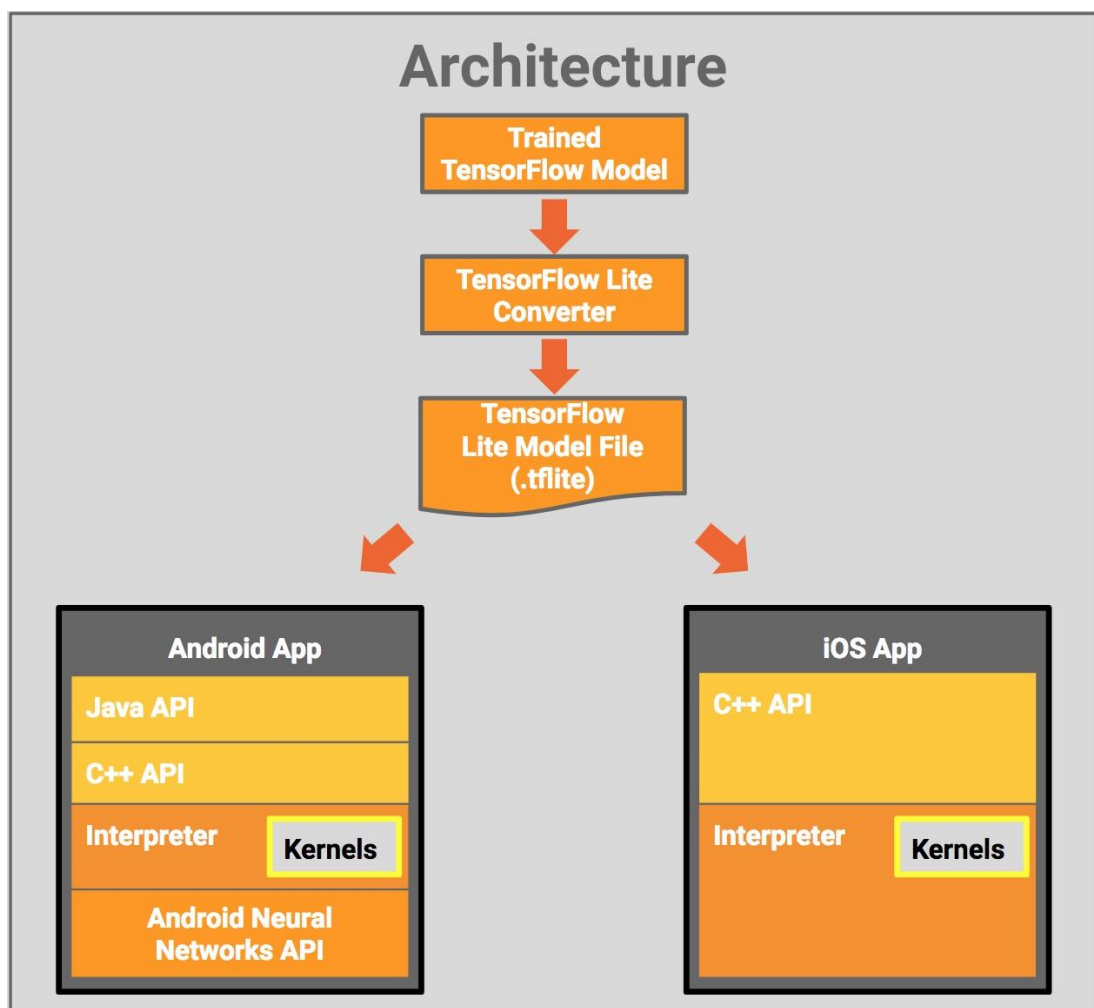


Рисунок 3.6 - Схема конвертації TF моделі у TF Lite

Цей процес був здійснений за допомогою TensorFlow Lite Converter API, який дозволяє зберегти архітектуру, ваги та параметри моделі в форматі TensorFlow Lite. Конвертована модель була оптимізована для забезпечення

швидкості та ефективності на мобільних пристроях, використовуючи методи, такі як квантизація ваг, обрізка моделі та інші. [24]

Перед розгортанням моделі на мобільному пристрої, ми провели перевірку її працездатності. Для цього використали TensorFlow Lite Interpreter API, який дозволяє завантажити та виконати модель на зразках даних. Ми переконалися, що модель працює належним чином та досягає точності в 96%.

3.5. Архітектура застосунку MVVM

Додаток має архітектуру MVVM (Model-View-ViewModel), яка дозволяє розподілити функціональність додатку на окремі компоненти та забезпечити масштабованість, тестованість та підтримку обміну даними між ними.

Архітектура MVVM складається з трьох основних компонентів: моделі (Model), представлення (View) та моделі представлення (ViewModel) (рис. 3.7). Кожен з цих компонентів виконує свої функції та взаємодіє з іншими компонентами для забезпечення працездатності додатку. [25]

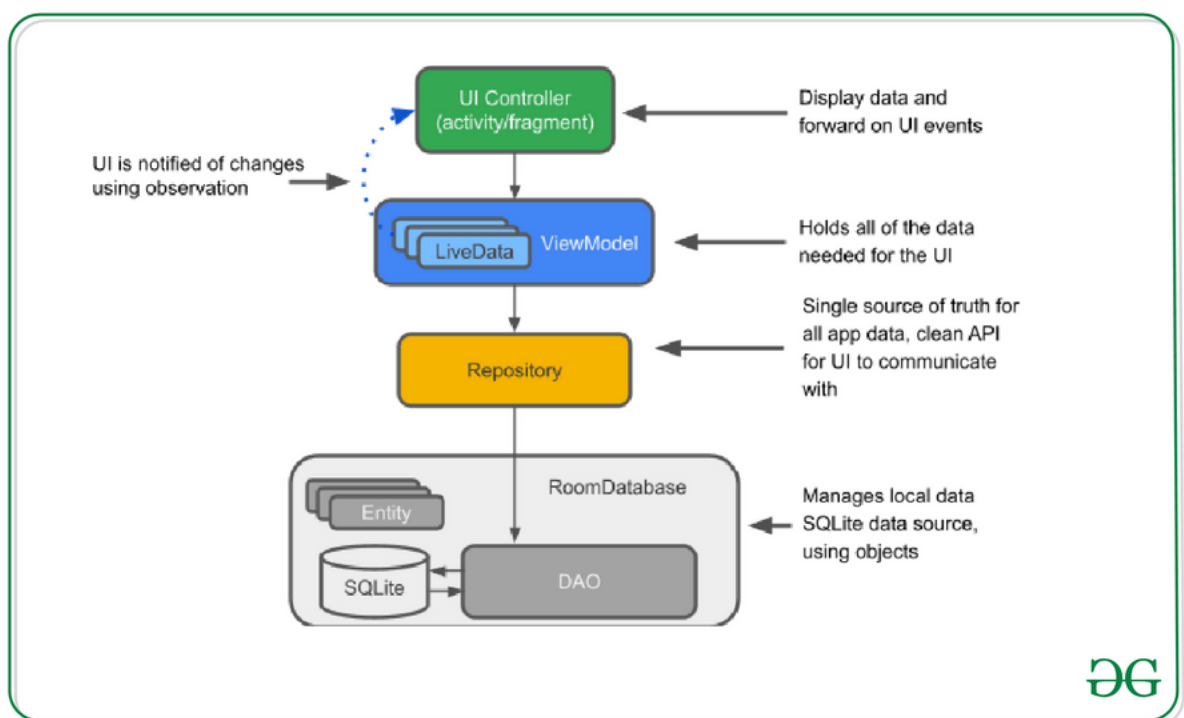


Рисунок 3.7 - Схема архітектури додатку

3.5.1. Model (Data Layer)

Модель (Data Layer) у MVVM використовується для управління доступом до даних та обробки бізнес-логіки. Цей шар відповідає за отримання та збереження даних, а також за взаємодію з різними джерелами даних, такими як бази даних, веб-сервіси або репозиторії.

Основна мета моделі в MVVM - це забезпечити розділення логіки даних від інтерфейсної логіки та представлення даних. Це дозволяє створювати модульні, масштабовані та легкі для тестування додатки.

Основні завдання моделі у MVVM:

1. Отримання даних: Модель відповідає за отримання даних з різних джерел, таких як бази даних, API або файлової системи. Вона може виконувати запити до зовнішніх ресурсів, обробляти отримані дані та передавати їх у ViewModel для подальшої обробки.
2. Збереження даних: Модель також відповідає за збереження даних до відповідних джерел. Це може включати оновлення бази даних, відправку даних до веб-сервера або збереження даних в локальному сховищі.
3. Бізнес-логіка: Модель може містити бізнес-логіку, яка відповідає за обробку даних та виконання розрахунків або перетворень даних. Вона може забезпечувати логіку валідації, фільтрації або сортування даних перед їхнім представленням.
4. Взаємодія з репозиторіями: Модель може взаємодіяти з репозиторіями, які служать проміжним шаром між моделлю та джерелами даних. Репозиторії виконують завдання кешування даних, керування запитамі та забезпечують єдиний доступ до даних з різних джерел.
5. Взаємодія з нейронною мережею: необхідна логіку для обробки зображень та їх класифікації. Методи для підготовки зображень перед класифікацією, такі як зменшення розміру, перетворення у формат, зручний для обробки моделлю, та інші операції попередньої обробки. Зміна розміру зображень

до необхідного розміру, конвертацію в відтинки сірого та інші операції, що поліпшують якість зображення або полегшують обробку.

Загалом, модель у MVVM виконує ключову роль у керуванні даними та бізнес-логікою. Вона забезпечує абстракцію доступу до даних та виконання операцій над ними, що дозволяє забезпечити чистоту та модульність коду.

3.5.2. View (Presentation Layer)

Представлення (View) використовується для відображення інтерфейсу користувача та обробки користувацьких дій. Цей шар відповідає за відображення даних, збереження стану елементів інтерфейсу та реагування на події від користувача.

Основна мета Представлення у MVVM - це створення користувацького інтерфейсу, який відображає дані з ViewModel та дозволяє користувачу взаємодіяти з додатком.

Представлення відповідає за відображення даних у відповідних компонентах інтерфейсу, таких як тексти, зображення, списки тощо, а також за обробку користувацьких подій, таких як натискання кнопок, введення тексту тощо.

Основні завдання Представлення у MVVM:

1. Відображення даних: Представлення отримує дані з ViewModel та відображає їх у відповідних компонентах інтерфейсу, таких як тексти, зображення, списки, карти тощо. Воно відповідає за відображення актуальних даних та їхнє оновлення при зміні в ViewModel.
2. Збереження стану інтерфейсу: Представлення також відповідає за збереження стану елементів інтерфейсу, наприклад, видимості, активності, вмісту полів введення тощо. Це дозволяє зберігати стан елементів інтерфейсу при зміні орієнтації екрану або при відновленні додатка зі сплячого режиму.
3. Обробка користувацьких подій: Представлення слухає користувацькі події, такі як натискання кнопок, введення тексту, переміщення по списку тощо. Воно передає ці події до ViewModel для подальшої обробки та

здійснення необхідних дій, наприклад, оновлення даних або виконання певної операції.

Загалом, Представлення у MVVM відповідає за взаємодію з користувачем та відображення даних з ViewModel. Воно виконує роль посередника між ViewModel та інтерфейсом користувача, допомагаючи створити зручний та інтуїтивно зрозумілий користувацький досвід.

3.5.3. ViewModel (Domain Layer)

Модель представлення (ViewModel) у архітектурному патерні MVVM (Model-View-ViewModel) використовується для управління даними та бізнес-логікою за допомогою зберігання та обробки даних, не залежно від конкретного інтерфейсу користувача.

Основна мета ViewModel у MVVM - це представлення даних та логіки, які використовуються в Представленні (View) і не пов'язані з конкретним інтерфейсом. Вона забезпечує розділення між логікою додатку та його візуальною частиною, що сприяє полегшенню тестування, підтримці інтерфейсів та повторному використанню.

Основні завдання ViewModel у MVVM:

1. Збереження та управління даними: ViewModel забезпечує збереження та управління даними, необхідними для відображення на екрані. Вона може отримувати дані з репозиторіїв, мережевих джерел або інших джерел даних, а також зберігати та оновлювати їх для відображення в Представленні.
2. Підготовка даних для відображення: ViewModel може виконувати обробку та форматування даних, щоб підготувати їх для відображення на екрані. Наприклад, вона може конвертувати числові значення у рядки, формувати дати та часи, вираховувати показники з даних тощо.
3. Керування бізнес-логікою: ViewModel відповідає за управління бізнес-логікою, пов'язаною з конкретними діями або операціями, які виконує користувач або система. Вона може містити методи для обробки подій,

валідації даних, виконання запитів до бази даних, мережі або інших джерел, обчислення результатів тощо.

4. Забезпечення зв'язку з Представленням: ViewModel надає зв'язок між Представленням і рештою системи. Вона може забезпечувати зв'язок з іншими компонентами, такими як репозиторії, сервіси або інші ViewModel, для отримання та передачі даних або сповіщень.
5. Збереження стану: ViewModel може зберігати стан додатку або екрану, щоб забезпечити правильне відновлення після повороту екрану або інших системних подій. Вона може зберігати дані у вигляді LiveData або інших рішень для забезпечення стабільності та працездатності додатку.

Загалом, ViewModel у MVVM відповідає за управління даними та бізнес-логікою, забезпечуючи розділення між візуальною частиною додатку та логікою за допомогою спеціального проміжного шару. Вона допомагає створити масштабовані, прості у тестуванні та підтримці додатки з чіткою відповідальністю для кожного компонента архітектури.

3.5.4. Repository

Окрім цих трьох компонентів, в архітектурі MVVM також використовуються різні допоміжні компоненти, такі як репозиторії, які відповідають за доступ до даних, інтерактори, які виконують бізнес-логіку додатку, та інші.

Репозиторій є допоміжним компонентом архітектури MVVM, який відповідає за комунікацію з джерелами даних та управління ними.

Основна функція репозиторію полягає в абстрагуванні деталей доступу до даних від решти додатку. Він надає один спосіб взаємодії з джерелами даних незалежно від того, чи це локальна база даних, веб-сервіс, файлова система або будь-яке інше джерело.

В нашому додатку, репозиторій відповідає за завантаження зображень зі зовнішнього джерела даних, такого як локальне сховище чи камера девайсу. Він також може включати логіку для перетворення та підготовки даних перед їх подальшою обробкою, зберігання даних.

Використання репозиторію має кілька переваг. По-перше, він дозволяє відокремити джерело даних від решти додатку, що полегшує зміни в цьому джерелі або введення нових джерел без впливу на решту коду. По-друге, він спрощує управління даними і дозволяє зосередитися на бізнес-логіці додатку. Крім того, він сприяє тестуванню, оскільки репозиторій може бути легко підмінений фейковими або тестовими реалізаціями для виконання автономних тестів.

У нашому додатку, репозиторій допомагає нам управляти завантаженням зображень та забезпечує їх доступність для обробки нейронною мережею. Репозиторій використовує певний інтерфейс, який описує операції доступу до даних, і реалізує цей інтерфейс з урахуванням конкретних потреб нашого додатку.

Загалом, використання репозиторію допомагає забезпечити чистоту коду, полегшує розширення функціональності і підтримку додатку, а також полегшує тестування.

3.5.5. Клас ViewModel

ViewModel є одним з ключових компонентів архітектури, відповідаючим за управління даними та бізнес-логікою додатку.

ViewModel використовується для зберігання та управління даними, які потрібні для відображення та інтеракції з користувачем. Він забезпечує розділення між репозиторієм, що надає дані, та UI-компонентами, які відображають ці дані. ViewModel зберігає дані, необхідні для екрану чи фрагмента, і надає їх у відповідному форматі для UI-компонентів. [26]

Одна з переваг використання ViewModel полягає в збереженні стану даних під час змін конфігурації, наприклад, повороту екрану. ViewModel зберігає дані незалежно від життєвого циклу активності або фрагмента, тому після зміни конфігурації дані залишаються доступними і можуть бути використані для відновлення стану UI (рис 3.8).

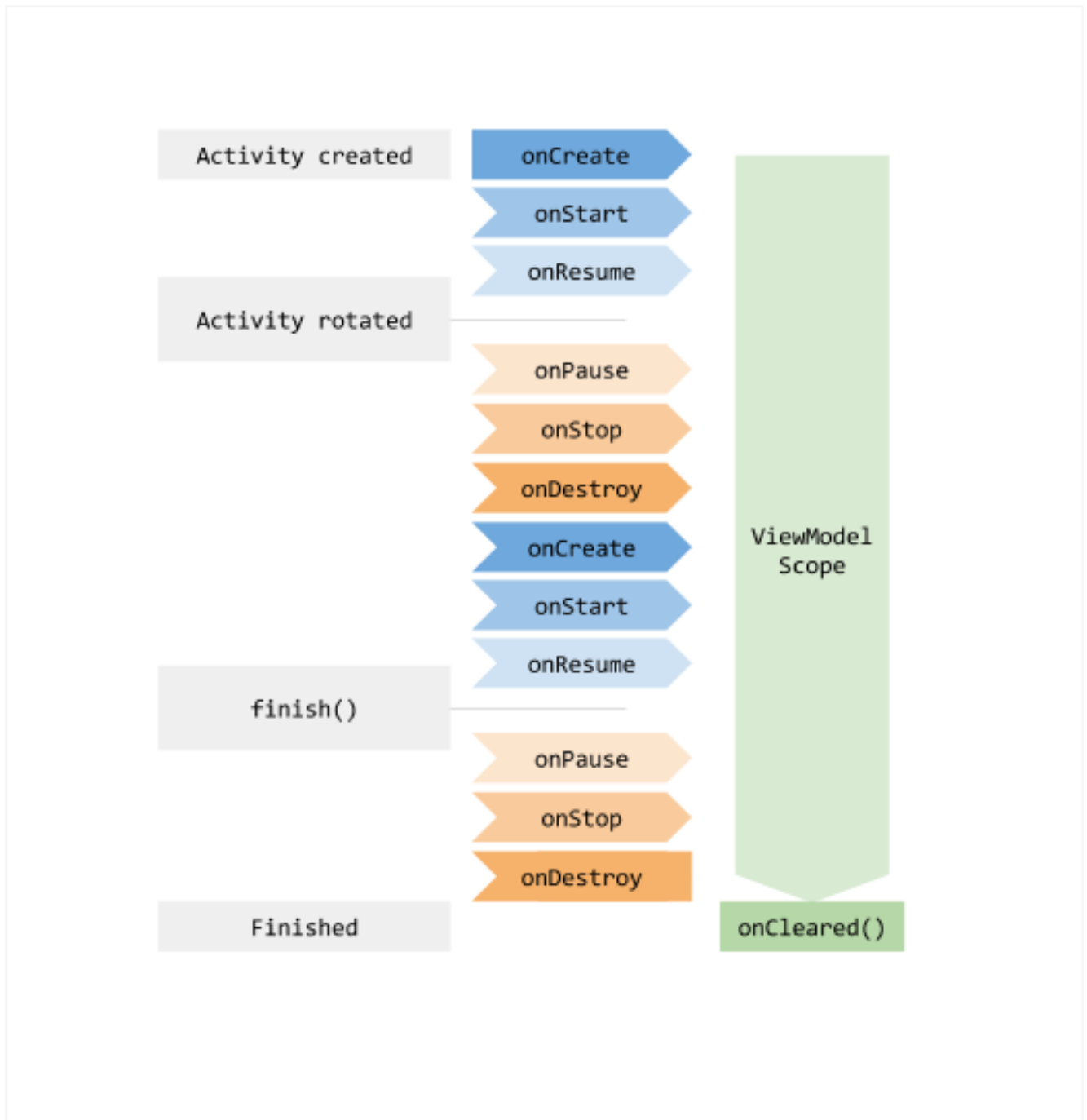


Рисунок 3.8 - Порівняння життєвого циклу Activity та ViewModel

Крім того, використання ViewModel спрощує тестування, оскільки дані та бізнес-логіка зосереджені в одному місці і можуть бути легко тестовані окремо від UI-компонентів. ViewModel дозволяє розібратися з потенційними проблемами, такими як пам'ять, утримання посилань та збереження стану додатку.

У нашому додатку, ViewModel відповідає за управління даними, отриманими від репозиторію, і підготовкою цих даних для відображення відповідними UI-компонентами.

Він також надає методи для виконання операцій класифікації зображень та обробки результатів. Це дозволяє забезпечити окремість відповідальностей між різними компонентами додатку та спрощує розробку та тестування.

Використання ViewModel в архітектурі MVVM дозволяє забезпечити чітку розділення між UI та бізнес-логікою, полегшує тестування та підтримку додатку, а також сприяє збереженню стану даних під час змін конфігурації.

Застосування архітектури MVVM в нашому додатку дозволяє розділити функціональність на окремі компоненти, що полегшує розробку, тестування та підтримку.

Крім того, вона забезпечує більшу гнучкість та масштабованість, що дозволяє легко вносити зміни та розширювати функціональність додатку у майбутньому.

Таким чином, архітектура MVVM виявилась ефективним вибором для реалізації нашого Android додатку для класифікації рентгенівських знімків. Вона допомогла нам забезпечити розділення відповідальностей, покращити швидкодію та забезпечити гнучкість розробки.

3.6. Зберігання даних. SQLite та Room

Room є потужною бібліотекою для роботи з базами даних в Android, яка надає зручний та ефективний спосіб зберігання та отримання даних.

Ця бібліотека є частиною Android Jetpack - набору компонентів, розроблених для полегшення процесу розробки Android додатків.

Однією з ключових переваг використання Room є його високий рівень абстракції, який дозволяє розробникам працювати з базою даних на рівні об'єктів, а не SQL-запитів. Room використовує підхід ORM (Object-Relational

Mapping), який забезпечує зв'язок між об'єктами даних та таблицями бази даних. Це дає змогу зберігати, оновлювати, видаляти та отримувати дані з бази даних з використанням звичайних об'єктно-орієнтованих операцій, що спрощує роботу з даними в Android додатках. [27]

Основними компонентами Room є:

1. Entity: Це клас або об'єкт, який представляє таблицю в базі даних. Кожен об'єкт Entity відповідає рядку в таблиці і містить поля, що представляють колонки таблиці.
2. DAO (Data Access Object): Це інтерфейс або абстрактний клас, який містить методи для доступу до даних в базі даних. За допомогою анотацій, розробники можуть визначати SQL-запити для виконання різних операцій з даними, таких як вставка, оновлення, видалення та вибірка.
3. Database: Це абстрактний клас, який представляє базу даних SQLite в додатку. Він включає методи для отримання об'єкту DAO та налаштування бази даних, таких як версія, індекси, початкові дані тощо.

Однією з ключових переваг Room є його можливість виконувати міграцію бази даних. Якщо структура бази даних змінюється, Room дозволяє автоматично або вручну оновити базу даних без втрати існуючих даних.

Крім того, Room надає додаткові можливості, такі як підтримка відношень між таблицями, асинхронні запити, спостереження за змінами даних та кешування результатів запитів.

У результаті використання Room розробники можуть зручно та ефективно працювати з базами даних в своїх Android додатках. Вона спрощує розробку, підтримку та розширення функціональності, пов'язаної з базами даних, і допомагає забезпечити ефективну та стабільну роботу додатків зберігання даних.

У нашому додатку, ми використовуємо Room для збереження історії сканувань зображень. Ми визначаємо сутність (Entity) для збереження даних про кожне сканування, такі як його айді, дата сканування, результат класифікації, назва скану, місце збереження картинки (табл. 3.2, рис. 3.9). Далі, ми створюємо

DAO, який визначає методи для роботи з цими даними, такі як вставка нового сканування, отримання списку всіх сканувань, отримання конкретної інформації про сканування, видалення тощо.

Таблиця 3.2

Дані сканування

Field	Type	Description
id	String	The primary key for the image data entity
imagePath	String	The path to the image file
name	String	The name of the image
classificationAccuracy	Float	The accuracy of the image classification
createDate	Long	The date the image data entity was created

	id	imagePath	name	classificationAccuracy	createDate
35	0413e00/-0301-47c	/data/user/0/com.kykar4a.lungimageclassifier/files/0413e00/-0301-47c3-044b-aa2f	Легені 8	0.311/823302/45819	16840/8322800
36	a8b064e1-68ba-40	/data/user/0/com.kykar4a.lungimageclassifier/files/a8b064e1-68ba-4045-901a-b3e	Легені 7	0.32335954904556274	1684078356450
37	03dea39e-8b47-4b	/data/user/0/com.kykar4a.lungimageclassifier/files/03dea39e-8b47-4bbe-94d9-cd1	Легені 6	0.32335954904556274	1684078378272
38	afc339df-3bb8-45e	/data/user/0/com.kykar4a.lungimageclassifier/files/afc339df-3bb8-45e9-9bd6-b47f	Легені 5	0.006142006255686283	1684079369099
39	4445777b-7c86-4a	/data/user/0/com.kykar4a.lungimageclassifier/files/4445777b-7c86-4a51-8a98-d9e	Легені 4	4.4374624849297106E-4	1684080310965
40	5a19bd35-e694-40	/data/user/0/com.kykar4a.lungimageclassifier/files/5a19bd35-e694-4068-a3c3-96f	Легені 3	0.0069556692615151405	1684080345017
41	99e504b4-84a0-4a	/data/user/0/com.kykar4a.lungimageclassifier/files/99e504b4-84a0-4aa8-a5ed-f3e	Легені 2	0.0069556692615151405	1684080414379
42	8cdf5743-5989-49	/data/user/0/com.kykar4a.lungimageclassifier/files/8cdf5743-5989-4922-add5-e18	Легені 1	3.829077943606762E-9	1684094136231

Рисунок 3.9 - Приклад зберігання даних знімків легень

Room дозволяє нам ефективно зберігати та отримувати історію сканувань зображень в нашому додатку. Це дозволяє користувачам переглядати свої попередні результати сканування та управляти ними.

3.7. Material Design You

Material Design You є новою еволюцією Material Design, яка надає можливості автоматичного адаптування інтерфейсу до вподобань користувача та контексту пристрою. Вона забезпечує персоналізацію інтерфейсу та створення унікального вигляду для кожного користувача.

Один з ключових аспектів Material Design You - це використання динамічних кольорів. Замість того, щоб просто вибирати статичні кольорові схеми, Material Design You дозволяє створювати динамічні палітри кольорів, які можуть змінюватися в залежності від вибору користувача або контексту пристрою. Це дозволяє створювати інтерфейси, які відповідають індивідуальним уподобанням користувача та забезпечують єдиний вигляд на різних пристроях (рис 3.10).

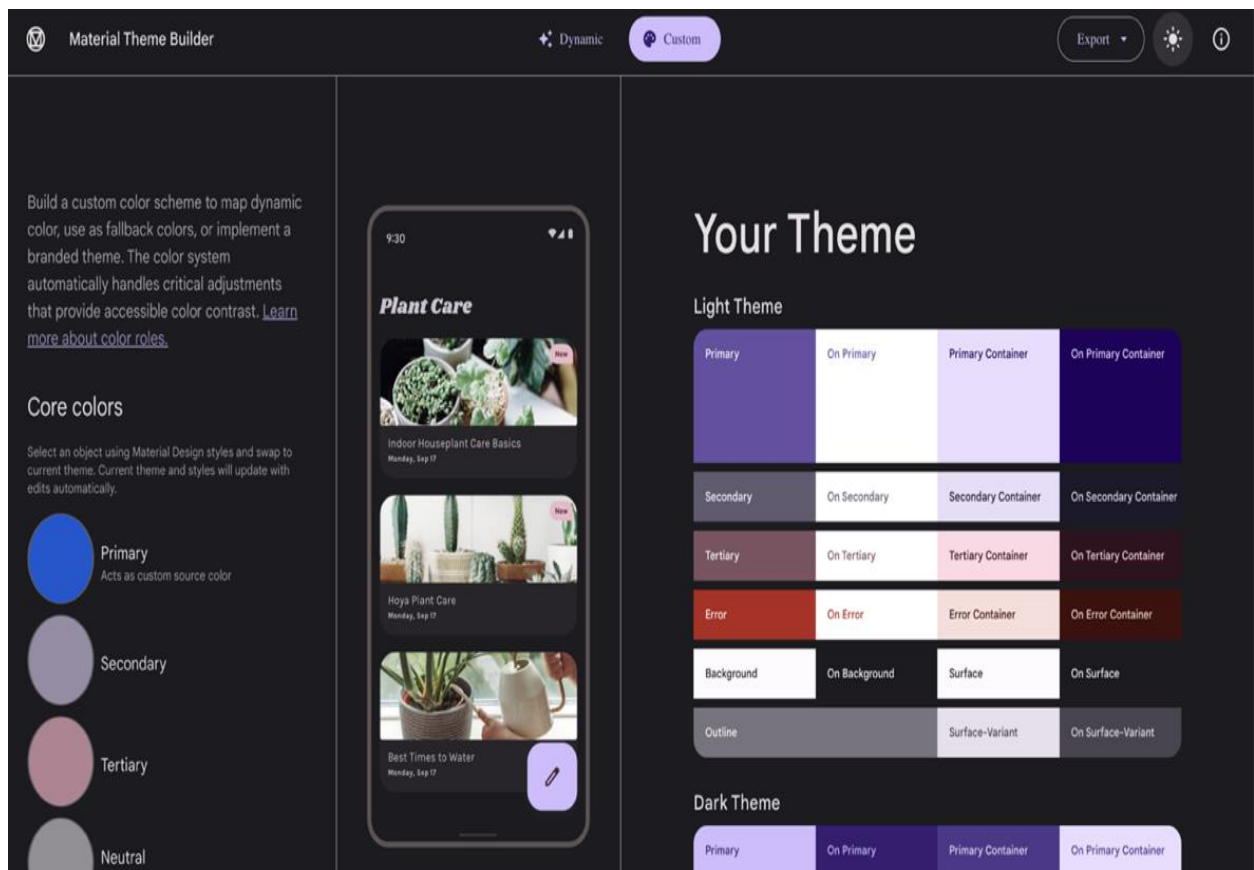


Рисунок 3.10 - Приклад використання Material Design You

3.8. Динамічні кольори

У нашому андроїд-застосунку ми використовуємо Material Design You і динамічні кольори для створення персоналізованого та змінного інтерфейсу. За допомогою цих функцій ми забезпечуємо користувачам можливість налаштування кольорової палітри за їхніми вподобаннями, створюємо візуальну консистентність інтерфейсу на різних пристроях та надаємо можливість вираження унікальності кожного користувача через дизайн (рис. 3.11).

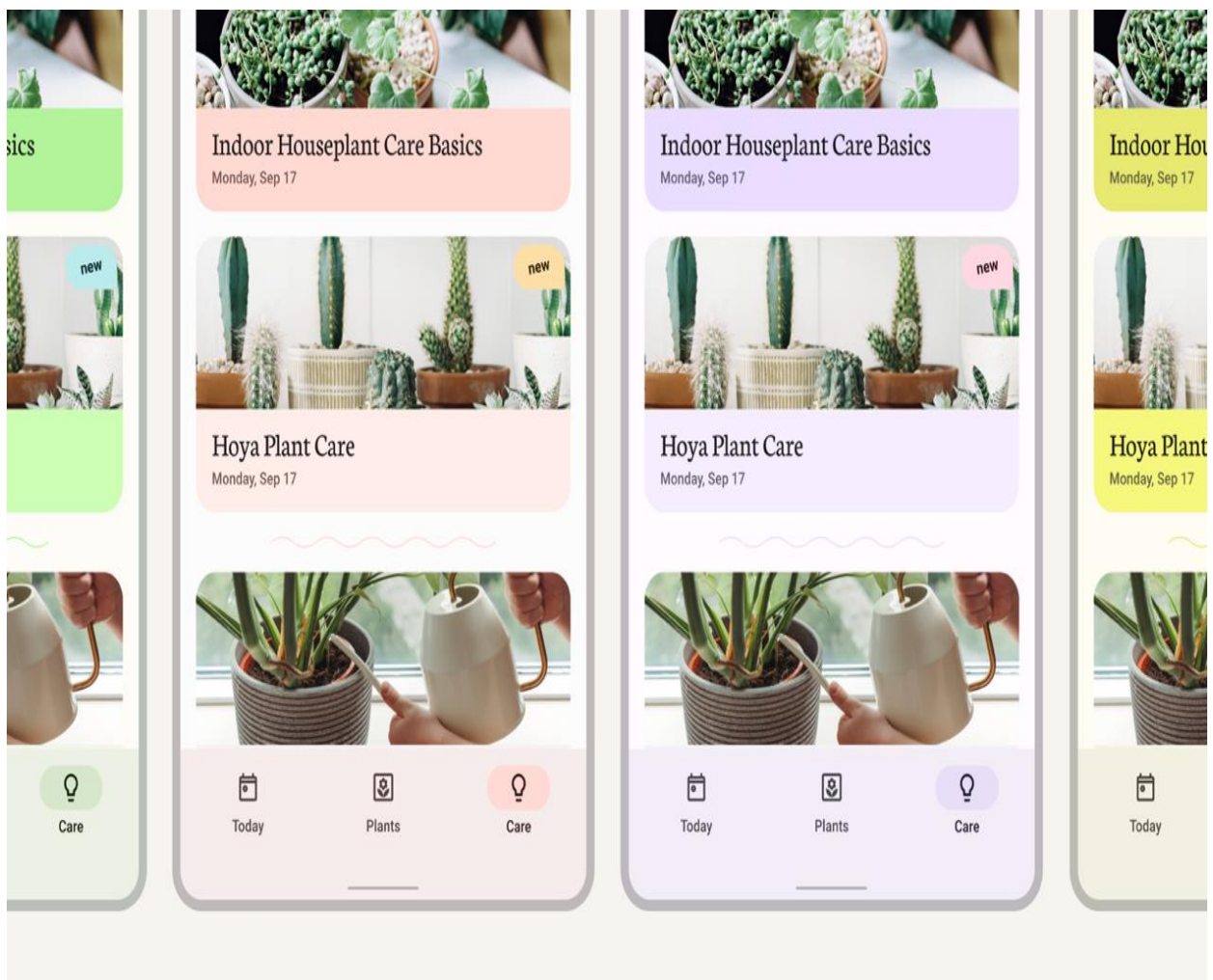


Рисунок 3.11 - Адаптація кольорів під шпалери користувача

Використання Material Design You та динамічних кольорів у нашому андроїд-застосунку дозволяє нам створювати модний, привабливий та

персоналізований інтерфейс для користувачів. Застосування компонентів Material Design та використання динамічних кольорів покращує взаємодію з застосунком, забезпечує єдиний вигляд на різних пристроях та створює незабутній враження для користувачів.

3.9. Демонстрація роботи програмного застосунку

У даному розділі буде продемонстровано роботу розробленого мобільного додатку для класифікації рентгенівських знімків здорових легень і хворих на туберкульоз. Нижче наведено опис основних функцій та фрагментів додатку, які будуть ілюструватися за допомогою скріншотів.

При першому вході, коли користувач ще не додав жодного зображення, або видалив усі, буде відображено привітальне вікно та підказка, що для початку потрібно натиснути на кнопку «Add image» (рис 3.12).

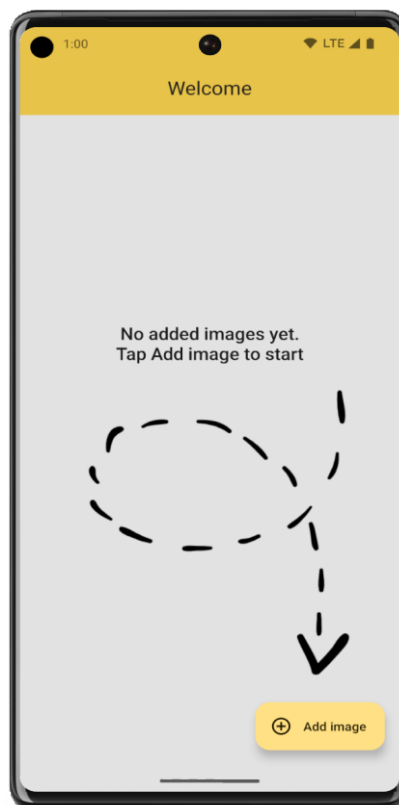


Рисунок 3.12 - Початковий екран програми

Після того, як користувач натиснув на кнопку Додати зображення, відкривається екран додавання. Користувачу надається 2 варіанти дії – додати зображення з галереї, або сфотографувати (рис 3.13).

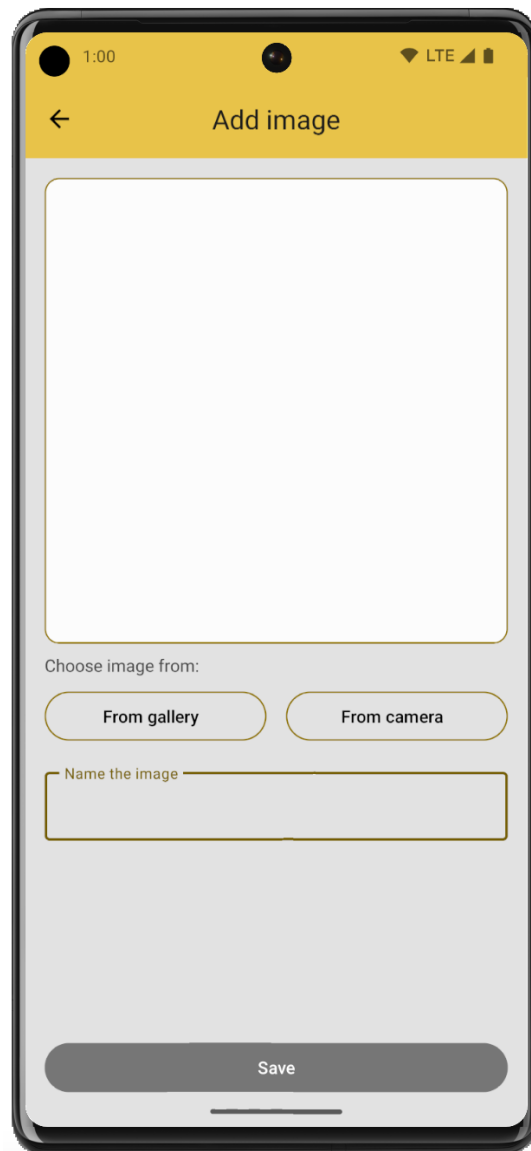


Рисунок 3.13 - Вікно додавання зображення

Якщо користувач обрав варіант додавання зображення з камери девайса – відкривається фото прев'ю (рис 3.14). Тут користувач може обрати ракурс фотографії і в залежності від марки та моделі телефону камеру, на яку буде зроблено знімок. Користувач може ввімкнути або вимкнути спалах та таймер.

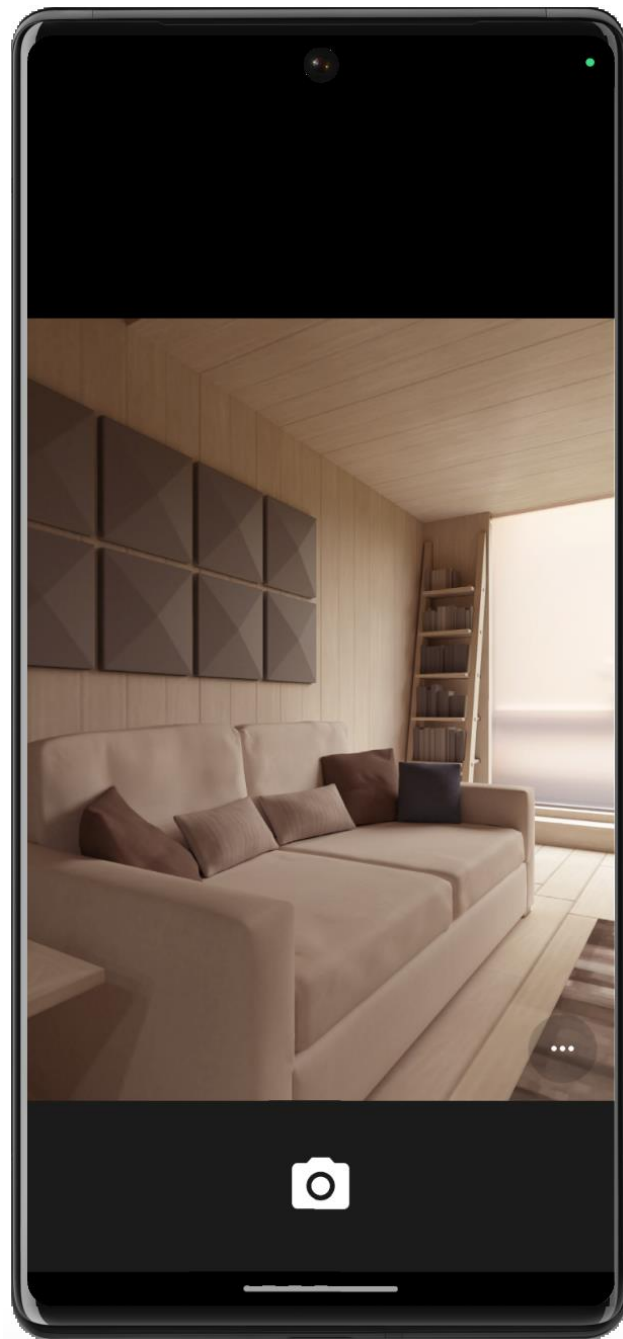


Рисунок 3.14 - Екран додавання зображення з камери

Після додавання фото з камери, застосунок повертає користувача на попередній екран, де вже відображається зроблене фото у верхній частині екрану (рис. 3.15). Якщо користувач вирішив переробити фото, то він має можливість повторно натиснути на відповідну кнопку та зробити нове.

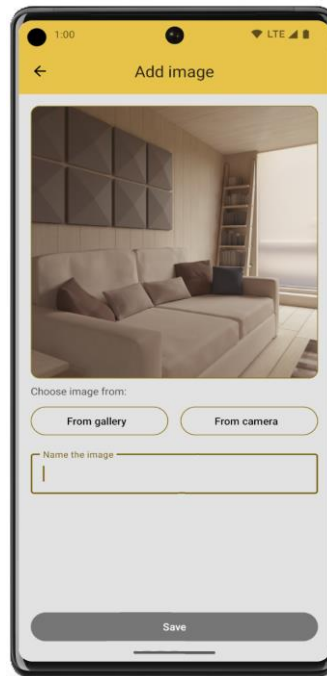


Рисунок 3.15 - Результат додавання фото з камери

При натисканні кнопки «from gallery» відкривається модальне вікно, в якому користувач може обрати одне з фото (рис. 3.16).

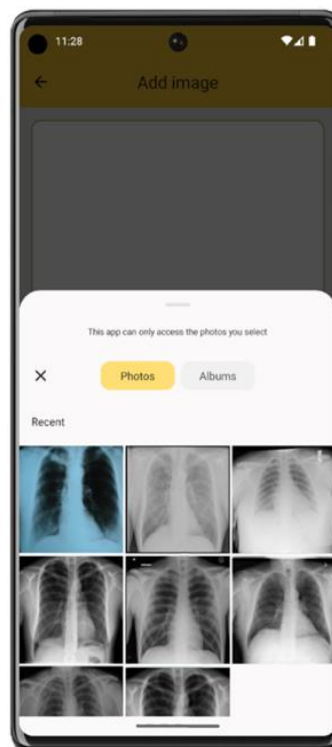


Рисунок 3.16 - Додавання зображення з галереї

За необхідності це вікно можна зробити більшим, або відкрити перегляд фотографій по папкам (рис. 3.17).

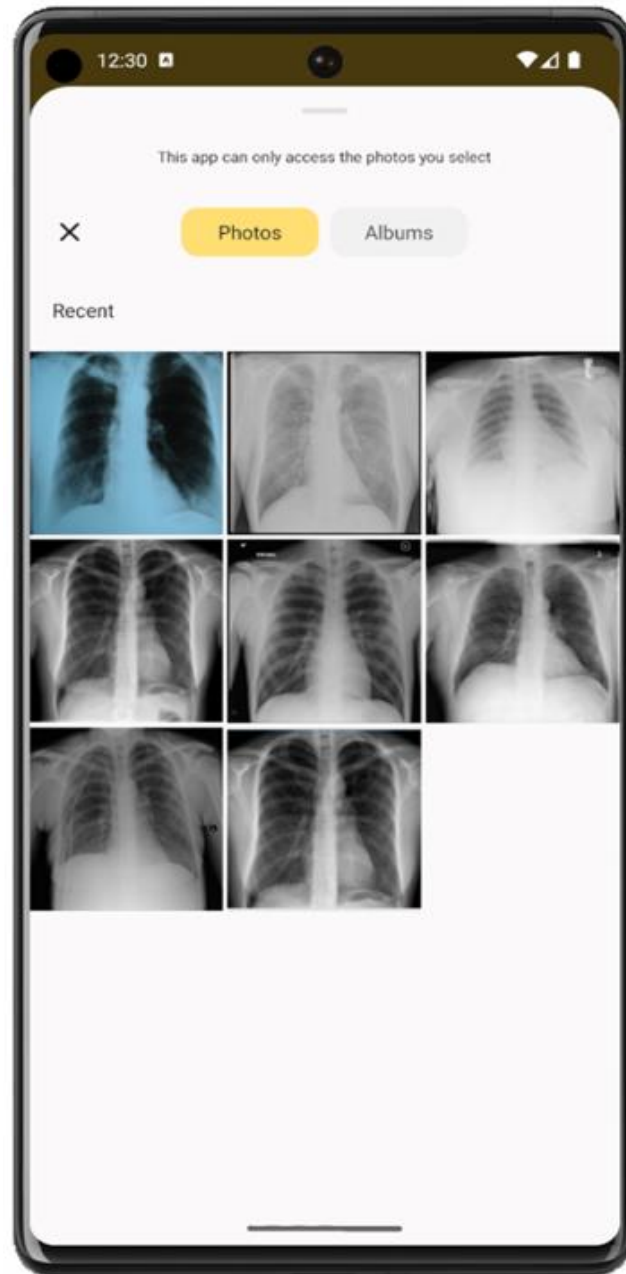


Рисунок 3.17- Розширений вигляд

Після додавання фото з галереї, воно завантажується у відповідне прев'ю. За потреби, натиснувши одну з двох кнопок from gallery або from camera обраний знімок можна замінити. Якщо ж користувач передумав додавати фото, в правому верхньому куті є стрілка, натиснувши на яку можна повернутися на головний екран.



Рисунок 3.18 - Вигляд екрану після додавання знімку з галереї

Однак додати лише зображення недостатньо. Кнопка «Save» буде неактивна поки користувач не введе назву зображення для подальшої ідентифікації (рис. 3.19).

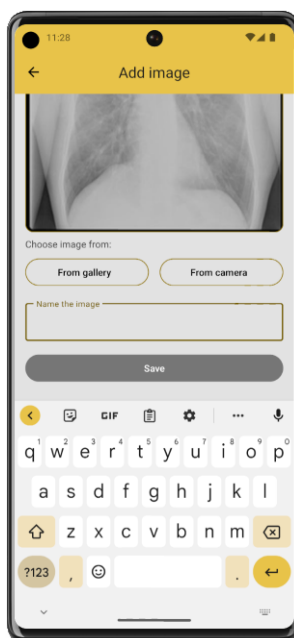


Рисунок 3.19 - Екран після натискання на поле вводу назви

Після того, як користувач додав зображення та ввів його назву, кнопка «Save» стає активною (рис. 3.20).

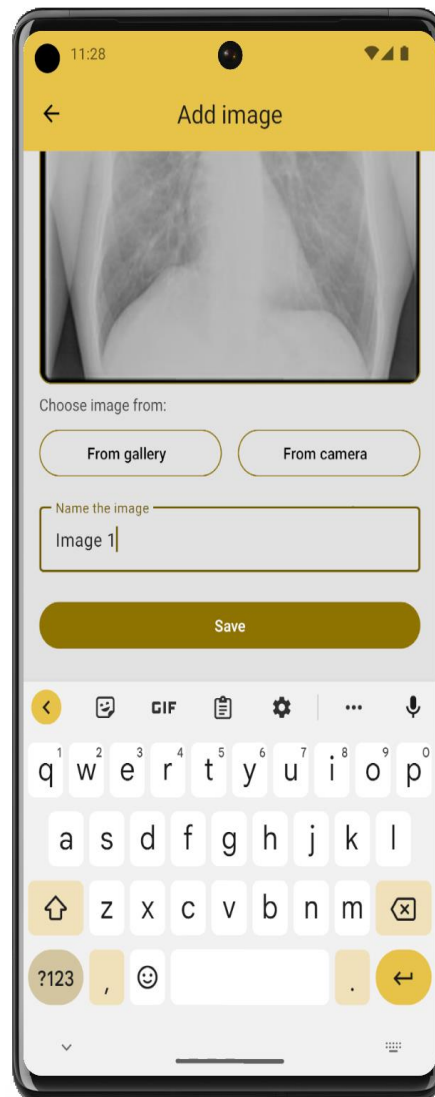


Рисунок 3.20 - Активна кнопка "Save"

Після того, як користувач натиснув кнопку «Save», інтерфейс програми блокується, з'являється лоадер, який сигналізує про те, що потрібно трішки почекати (рис. 3.21). В цей час відбувається аналіз зображення, його адаптація під нейронну мережу, збереження файлу зображення до приватної файлової системи застосунку та збереження інформації про зображення до бази даних Room на базі SQLite.

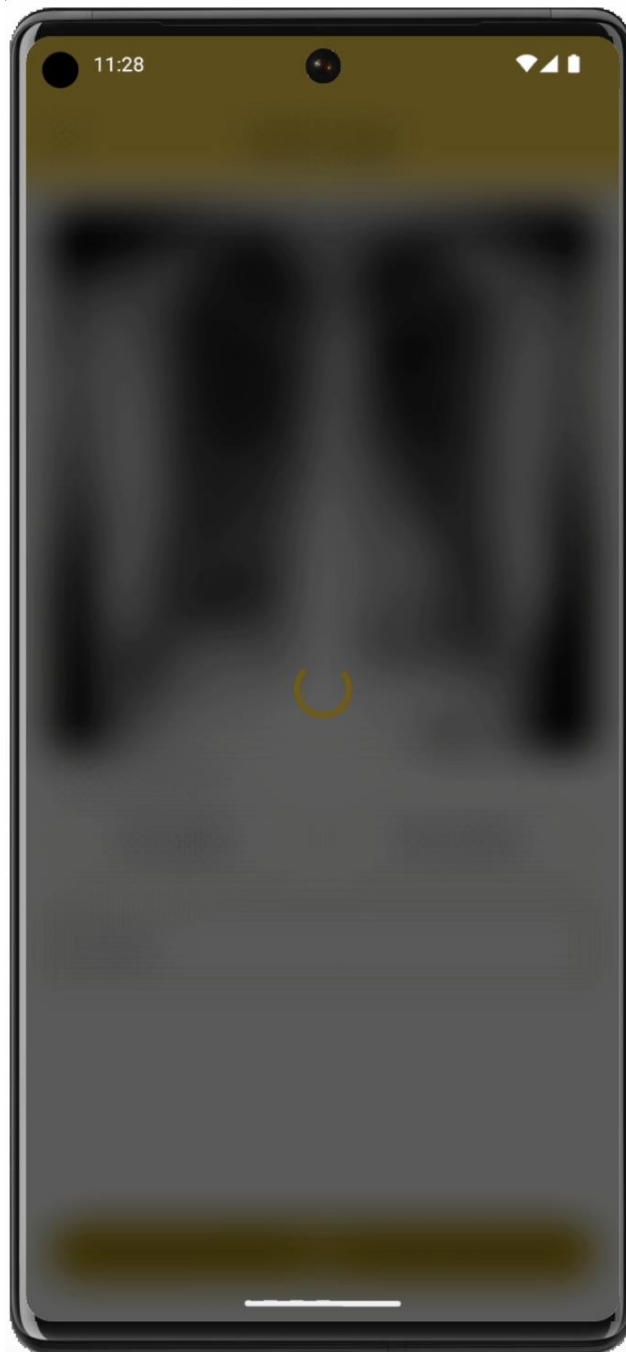


Рисунок 3.21 - Вигляд додатку під час аналізу зображення

Після того як обробка зображення закінчилась, користувач потрапляє на екран детальної інформації, де може ознайомитися з результатом аналізу зображення. В полі *illness probability* з'являється вірогідність того, що пацієнт, чий рентген знімок було завантажено, хворий на туберкульоз. Поле може приймати різні кольори в залежності від того, який відсоток написано. Червоний колір – висока вірогідність, жовтий – середня, зелений – низька (рис 3.22).

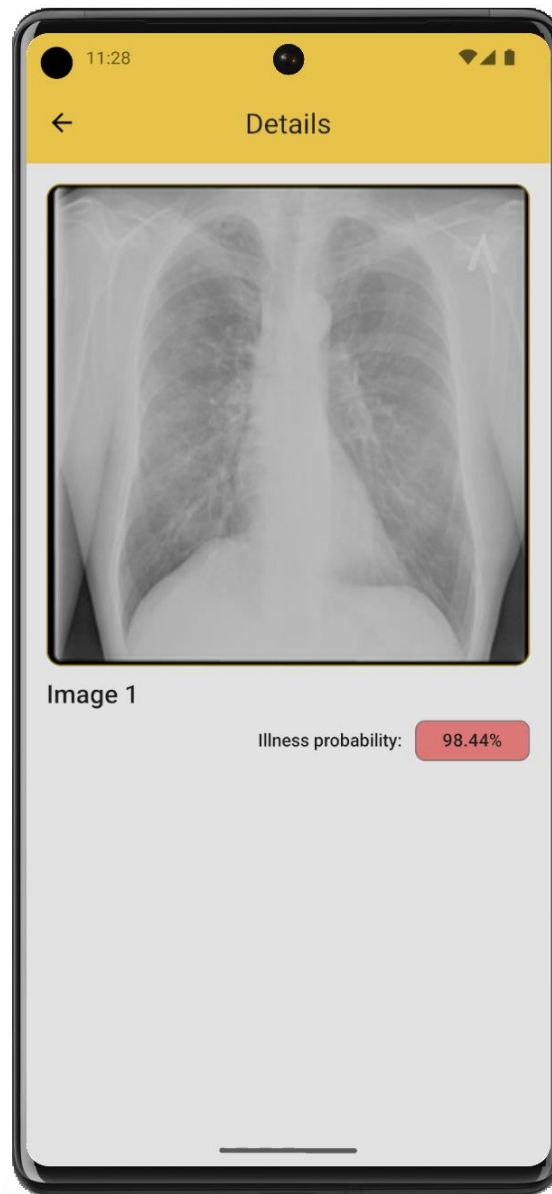


Рисунок 3.22 - Екран деталей зображення

Щоб потрапити на головний екран, потрібно натиснути стрілку в верхній лівій частині екрану, або використати кнопку або жест назад нативної навігації.

Після повернення на головний екран користувач бачить попередньо додані зображення та основну інформацію про них (рис. 3.23 – рис 3.24). По натисканню на зображення, буде відкритий відповідний екран детальної інформації. Також можна знову натиснути кнопку «Add image» для того, щоб додати інший знімок.

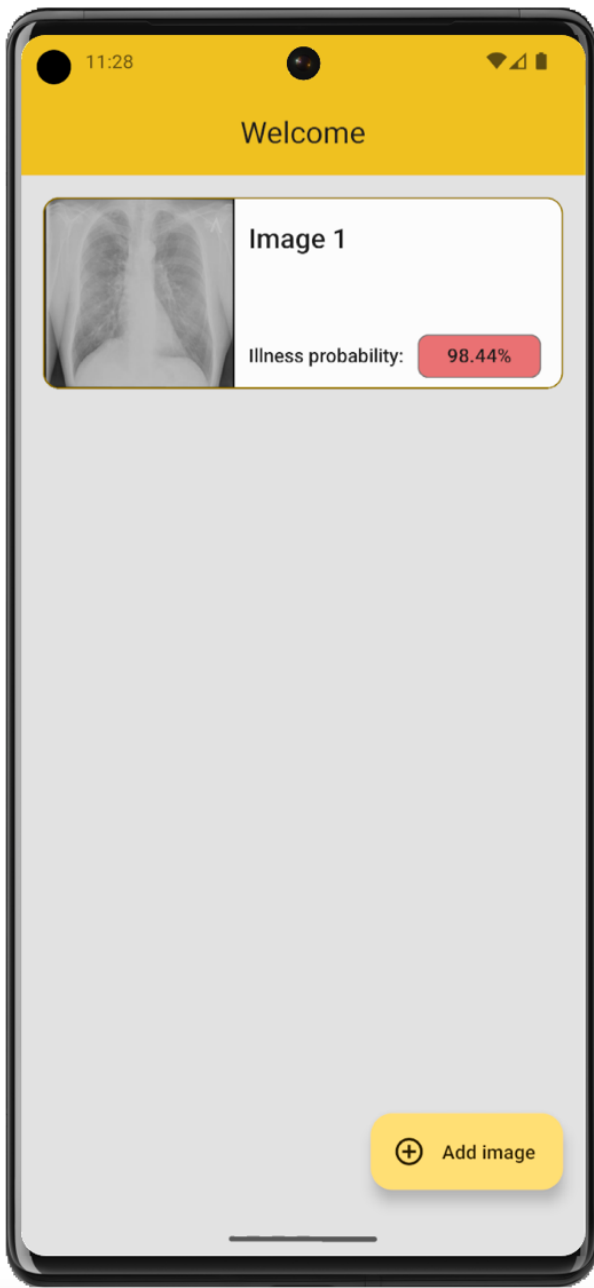


Рисунок 3.23 - Головний екран

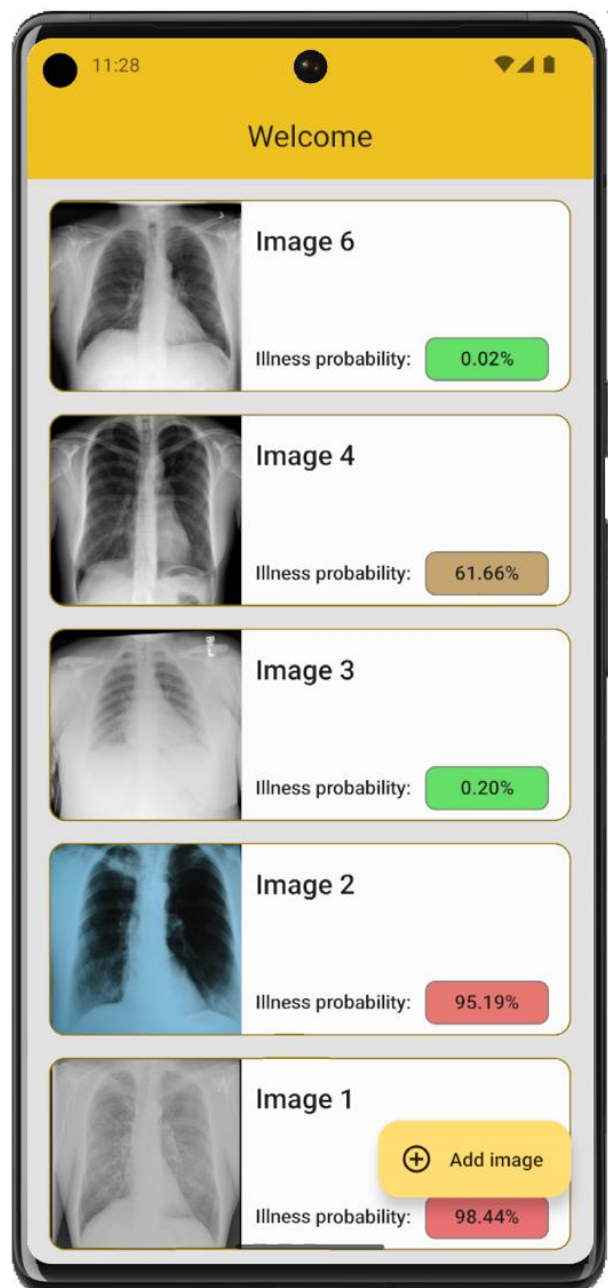


Рисунок 3.24 - Головний екран після додавання знімків

Якщо користувач зрозумів, що помилився, або інформація більше не актуальна, то він може видалити зображення зі списку. Для цього потрібно

провести справа наліво по відповідному елементу та змахнути його вліво (рис 3.25 – рис. 3.26).

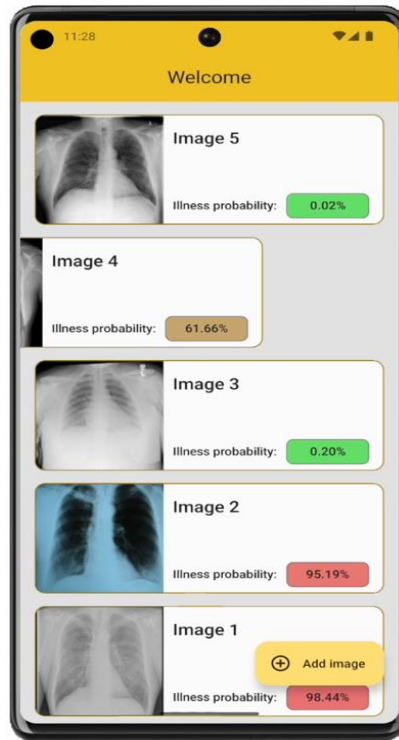


Рисунок 3.25 - Процес видалення зображення

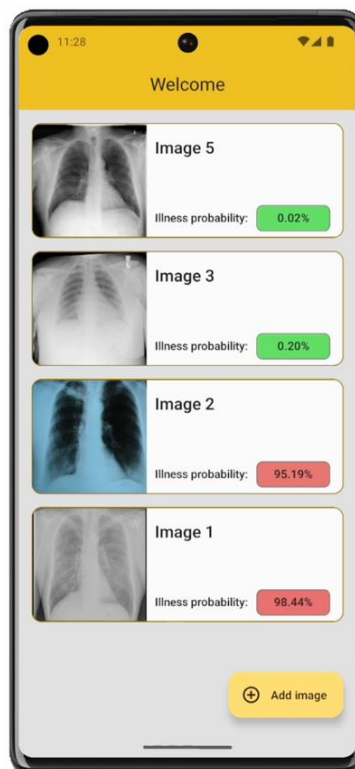


Рисунок 3.26 – Результат видалення зображення

Як бачимо, на головному екрані також є кольорова візуалізація вірогідності захворювання – від червоного при великій вірогідності і до зеленого при мінімальній.

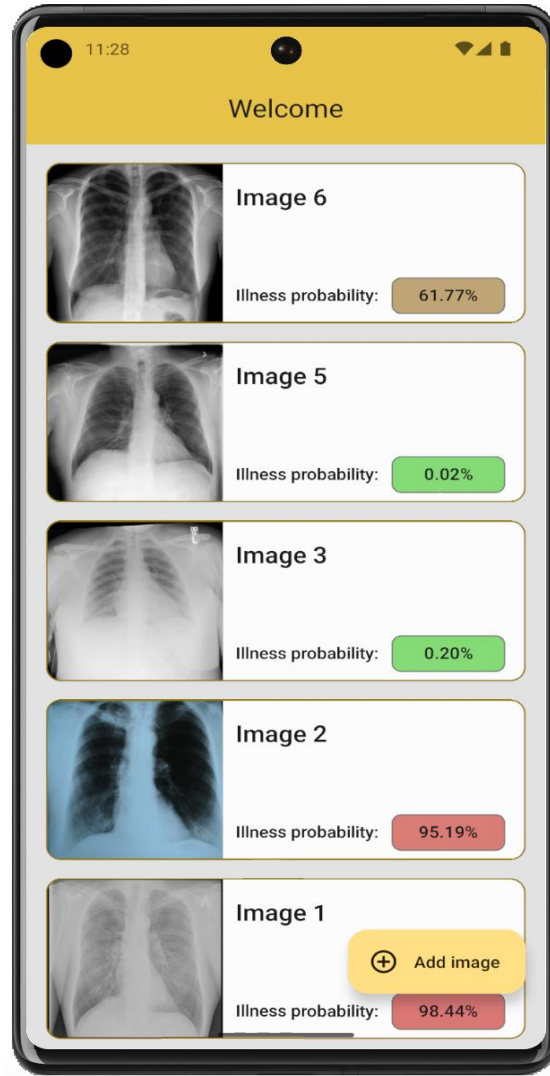


Рисунок 3.27 - Вигляд зображень з різною вірогідністю захворювання

Якщо на девайсі використовується темна тема, то фон нашого додатку стає темним, що сприяє комфортному використанню в умовах недостатнього освітлення. Текст, навпаки, змінюється на світлий, щоб забезпечити читабельність і контрастність. Крім того, кольорові елементи, такі як кнопки або індикатори, також адаптуються до темної теми, змінюючи свої відтінки для забезпечення кращої видимості та зручного користування (рис. 3.28).

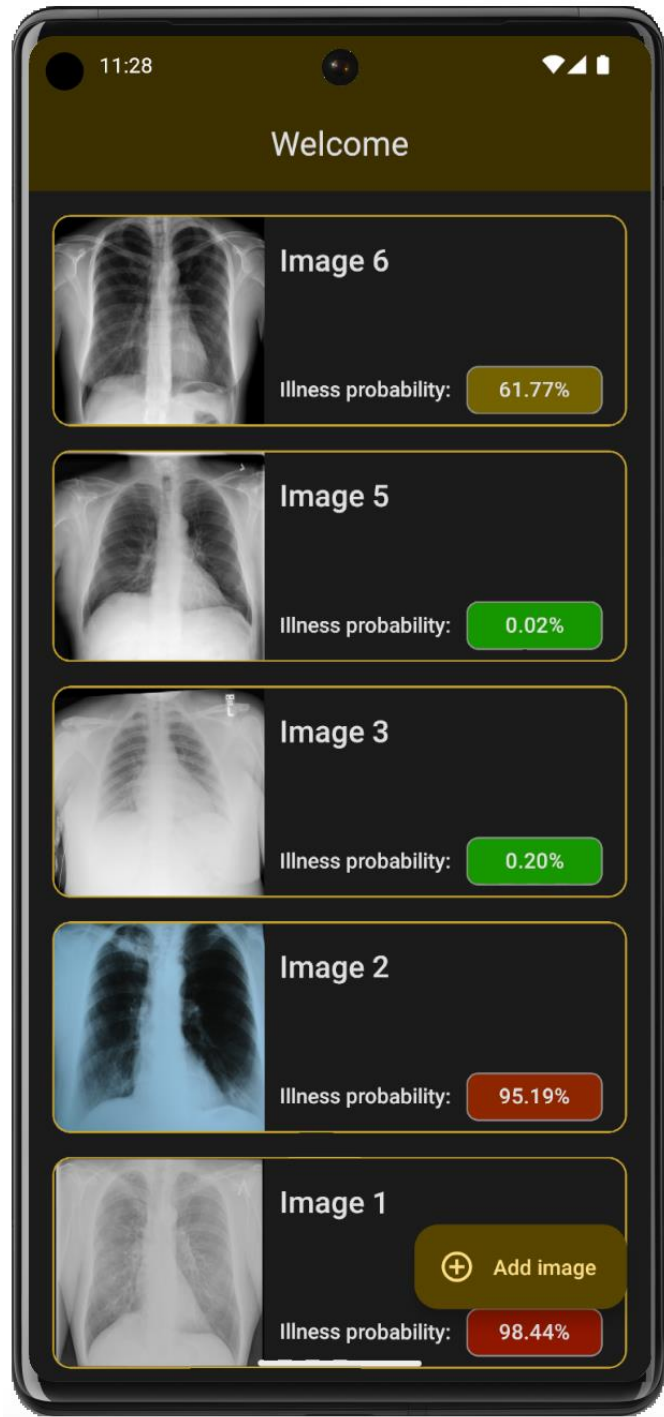


Рисунок 3.28 - Темна тема застосунку

Основний колір інтерфейсу застосунку напряму залежить від заставки на робочому столі девайсу (рис 3.29 – рис 3.30). Dynamic colors із концепції дизайну Material You від Google робить зовнішній вигляд програми унікальним та сучасним.



Рисунок 3.29 - Кольорова гама застосунку при синіх шпалерах

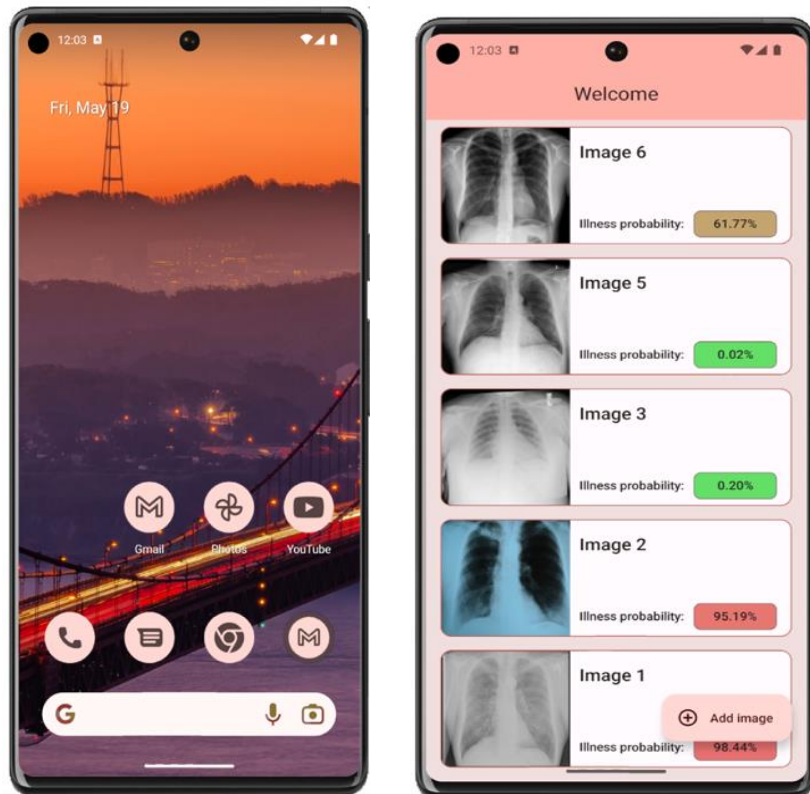


Рисунок 3.30 - Кольорова гама застосунку при червоних шпалерах

3.10. Розрахунок економічного ефекту

У цьому розділі було проведено функціонально-вартісний аналіз програмного продукту, що розроблюється для класифікації рентгенівських знімків легень на здорові та хворі на туберкульоз. Для цього використовується нейронна мережа. Програмне забезпечення буде працювати на android девайсі.

Відобразимо визначені основні функції на морфологічній карті (рис. 3.31):

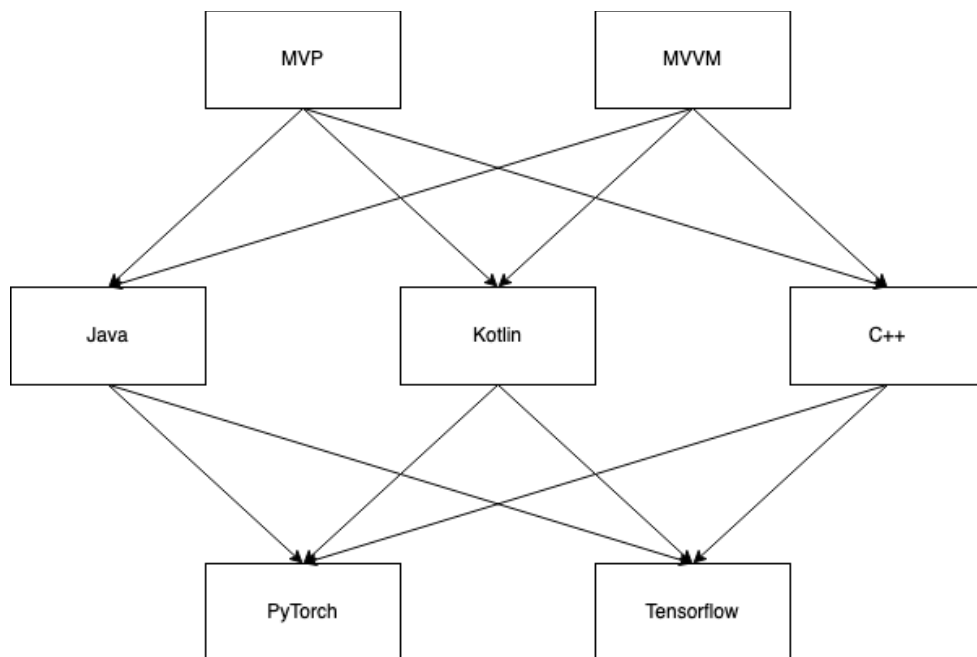


Рисунок 3.31 – Морфологічна карта системи

В результаті дослідження в галузі економіки та організації виробництва для розробки мого андроїд додатку для класифікації КТ знімків легень були проведені техніко-економічні обґрунтування, порівняння та вибір оптимального варіанту реалізації. Аналізуючи основні функції та ефективність двох варіантів реалізації, було встановлено, що обидва варіанти мають свої переваги і вартість розробки. Але другий варіант виявлений більш ефективний з техніко-економічної точки зору у виконанні функцій програмного продукту, який забезпечує менший об'єм коду, при цьому не надто збільшуючи вартість проекту, порівняно з першим варіантом, а саме 342 687.98 грн замість

341 759.04 грн. Цей варіант передбачає використання мови програмування Kotlin, платформи навчання нейронної мережі для класифікації знімків TensorFlow та архітектури програмного застосунку MVVM.

Висновки до розділу 3

В практичній частині дипломної роботи було реалізовано мобільний додаток для класифікації рентгенівських зображень здорових легень і хворих на туберкульоз. Додаток був розроблений з використанням архітектурного патерну MVVM (Model-View-ViewModel) та найновіших технологій розробки Android.

Окрім того, було успішно реалізовано базу даних з використанням SQLite та Room для зберігання історії сканувань зображень. Інтеграція технології глибокого навчання дозволила класифікувати зображення та визначати ознаки хвороби на туберкульоз з високою точністю.

Крім того, була використана Material You концепція для персоналізації інтерфейсу та динамічних змін кольорової палітри відповідно до налаштувань користувача. Це дозволило забезпечити індивідуальний досвід користувача та підвищити зручність використання додатку.

Також, було проведено функціонально-вартісний аналіз, де в результаті аналізу основних функцій та ефективності двох основних варіантів реалізації було встановлено, що обидва варіанти мають свої переваги і вартість розробки. Але другий варіант виявлений більш ефективний з техніко-економічної точки зору у виконанні функцій програмного продукту, який забезпечує менший об'єм коду, при цьому не надто збільшуючи вартість проекту, порівняно з першим варіантом

У результаті практичної частини був успішно реалізований мобільний додаток, який може бути використаний для класифікації рентгенівських зображень та визначення ознак захворювання на туберкульоз. Цей додаток має потенціал для використання в медичних установах та сприяє покращенню діагностики та контролю за туберкульозом.

ЗАГАЛЬНІ ВИСНОВКИ

В результаті виконання роботи було проведено обширне дослідження та розробка мобільного додатку з метою класифікації зображень комп'ютерної томографії здорових легень і хворих на туберкульоз. Аналітичний огляд літературних джерел дозволив ознайомитись з актуальними дослідженнями та рекомендаціями, зокрема з інформації від Всесвітньої організації охорони здоров'я щодо управління латентною формою туберкульозу та лікування.

Також було детально проаналізовано роль нейронних мереж у класифікації зображень, зокрема згорткових та рекурентних нейронних мереж. Використання згорткових нейронних мереж дозволяє ефективно виявляти особливості та ознаки захворювання на туберкульоз, тоді як рекурентні нейронні мережі можуть бути корисними при аналізі динаміки хвороби.

Також були розглянуті альтернативні методи класифікації зображень, які можуть бути застосовані у випадках, коли нейронні мережі не є найкращим варіантом. Досліджено використання комп'ютерної томографії та інших методів, таких як аналіз морфологічних ознак та текстурних характеристик зображень.

Практична частина роботи включала розробку мобільного додатку, який був успішно реалізований. Використовуючи фреймворк TensorFlow та згорткову нейронну мережу, додаток здатний точно класифікувати зображення комп'ютерної томографії і визначати ознаки захворювання на туберкульоз.

Отримані результати вказують на потенціал мобільних технологій у медицині та показують, що розроблений мобільний додаток може бути цінним інструментом для поліпшення діагностики та контролю за туберкульозом. Це відкриває перспективи для подальших досліджень та розробок у цій галузі, зокрема для вдосконалення алгоритмів класифікації та впровадження додаткових функціональних можливостей, що сприятимуть покращенню якості діагностики та лікування хворих на туберкульоз.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. World Health Organization. (2021). Latent tuberculosis infection: updated and consolidated guidelines for programmatic management. 2018. License: CC BY-NC-SA 3.0 IGO.
2. Treatment of latent tuberculosis infection: An updated network meta-analysis [Електронний ресурс] // Annals of Internal Medicine. – 2020. – Режим доступу до ресурсу: <https://doi.org/10.7326/M17-0609>.
3. Sreelakshmi, P. R., & Jayasankar, R. (2020). A review on computer-aided diagnosis of tuberculosis using chest radiographs. Journal of Medical Imaging and Health Informatics.
4. Jin, C., Chen, X., Wei, L., Liu, X., Zhang, Z., & Li, Y. (2020). Artificial intelligence accurately predicts tuberculosis in chest radiographs of HIV-negative and HIV-positive patients.
5. TensorFlow. TensorFlow Lite: Deploy machine learning models on mobile and IoT devices. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tensorflow.org/lite?hl=en>.
6. TensorFlow Lite Android Image Classification Example [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/tensorflow/examples/tree/master/lite/examples/image_classification/android.
7. Тодоріко, Л. Д., Островський, М. М., Семянів, І. О., & Шевченко, О. С. (2020). Особливості перебігу туберкульозу в умовах пандемії COVID-19. Туберкульоз Легеневі хвороби ВІЛ-інфекція, 43(4), 52-63.
8. Нерус, В. М. (2022). Нейронна мережа для виявлення відхилень грудної клітини на рентгенівських знімках (Master's thesis, КПІ ім. Ігоря Сікорського).
9. Шемет, С. В. (2022). Застосування згорткової нейронної мережі для обробки та аналізу МРТ-зображень.

10. Chapaliuk, B. V., & Zaychenko, Y. P. (2019). Використання рекурентних нейронних мереж для автоматичної діагностики раку легенів. *System research and information technologies*, (3), 33-40.
11. Кучер, В. В. (2022). Методи машинного навчання аналізу рентгенівських зображень для діагностики COVID-19 (Master's thesis, КПІ ім. Ігоря Сікорського).
12. Бондаренко, О. С. (2016). Мобільні додатки для медицини. *Міжнародний науковий журнал*, (6 (2)), 66-69.
13. Ardito, L., Coppola, R., Malnati, G., & Torchiano, M. (2020). Effectiveness of Kotlin vs. Java in android app development tasks. *Information and Software Technology*, 127, 106374.
14. Devagiri, J. S., Paheding, S., Niyaz, Q., Yang, X., & Smith, S. (2022). Augmented Reality and Artificial Intelligence in industry: Trends, tools, and future challenges. *Expert Systems with Applications*, 118002.
15. Sanchez, D., Rojas, A. E., & Florez, H. (2022). Towards a Clean Architecture for Android Apps using Model Transformations. *IAENG International Journal of Computer Science*, 49(1), 270-278.
16. Данілов, Д. В. (2022). Дослідження основних шаблонів Model View Controller та Model View ViewModel для проектування Android додатків. In *Інноватика в освіті, науці та бізнесі: виклики та можливості*. Київський національний університет технологій та дизайну.
17. Jaya, I. K. (2021). Penerapan Model View Controller (MVC) Pada Perancangan Aplikasi Pencarian Pada Sistem Informasi Perpustakaan Berbasis Android Menggunakan Algoritma Boyer Moore (Studi Kasus: Perpustakaan SMA Methodist 1 Medan). *Jurnal Ilmiah Teknik Informatika METHOTIKA*, 1(2), 52-59.
18. Muttaqin, M. A., & Susanto, A. (2020). Aplikasi Stok Audit di PT. Media Sarana Data Berbasis Android Dengan Arsitektur Model View Presenter. *Jurnal Maklumatika*, 112-123.

19. Arif, M. S., Musthafa, A., & Muriyatmoko, D. (2019). Implementation of Model-View-ViewModel (MVVM) Architecture Pattern in the Sistem Informasi Akademik UNIDA Gontor Mobile Application (Doctoral dissertation, Universitas Darussalam Gontor).
20. Kartarina, K., & Apriliansyah, A. (2022). Analisis Dependency Injection dan Model-View-Presenter Pada Aplikasi Berbasis Android. *Progresif: Jurnal Ilmiah Komputer*, 18(1), 23-32.
21. Радзієвська, О. (2020). Особливості використання машинного навчання та доповненої реальності на пристроях на базі Android.
22. Lee, J. J., Aime, M. C., Rajwa, B., & Bae, E. (2022). Machine Learning-Based Classification of Mushrooms Using a Smartphone Application. *Applied Sciences*, 12(22), 11685.
23. Singh, A., & Bhadani, R. (2020). *Mobile Deep Learning with TensorFlow Lite, ML Kit and Flutter: Build scalable real-world projects to implement end-to-end neural networks on Android and iOS*. Packt Publishing Ltd.
24. Марич, Т. І. (2019). Реалізація нейронних мереж в мобільних застосунках (Master's thesis, КПІ ім. Ігоря Сікорського).
25. Козін, І. П. (2021). ВИКОРИСТАННЯ ДИЗАЙН-ПАТЕРНІВ MVVM ТА MVP ПРИ РОЗРОБЦІ МОБІЛЬНИХ ЗАСТОСУНКІВ МОВОЮ KOTLIN. *Наукові записки молодих учених*, (8).
26. Saputro, Y. S. (2020). IMPLEMENTASI ANDROID ARCHITECTURE COMPONENTS DENGAN PATTERN MVVM (MODEL-VIEW-VIEW MODEL) PADA APLIKASI PELAPORAN KEMISKINAN DI KABUPATEN BANTUL (Doctoral dissertation, STMIK AKAKOM Yogyakarta).
27. Білаш, Д. А. ОСОБЛИВОСТІ ЗАСТОСУВАННЯ БАЗ ДАНИХ ROOM В ANDROID-ЗАСТОСУНКАХ. 27-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у ХХІ столітті». Зб. матеріалів форуму. Т. 5.–Харків: ХНУРЕ. 2023.–178 с., 77.