Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Факультет інформатики
Кафедра мультимедійних систем

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ПІД ОПЕРАЦІЙНУ СИСТЕМУ IOS

Текстова частина до курсової роботи
за спеціальністю 122 «Комп'ютерні науки»

Керівник курсової роботи
ст.в.___Борозенний С.О.___
(прізвище та ініціали)
_____
(підпис)
"____" _____ 2021 р.
Виконав студент Миронович О.Ю.
(прізвище та ініціали)
_____
(підпис)
"____" _____ 2021 р.

Київ-2021

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Завідувач кафедри мультимедійних систем,

канд. фіз-мат. наук, доц. – Жежерун О.П.

_____ *(підпис)*

“____” _____ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу
студенту Мироновичу Олександру Юрійовичу
Факультету інформатики 3 р.н. бакалаврської програми
ТЕМА: Розробка мобільного застосунку під операційну систему iOS

Зміст ТЧ до курсової роботи:
Календарний план
Анотація
Вступ
1 Дослідження та аналіз
2 Технології
3 Розробка мобільного застосунку
Висновки
Список використаної літератури та посилань
Додатки (за необхідністю)

Дата видачі “____” _____ 2020 р. Керівник _____

(підпис)

Завдання отримав _____
(підпис)

Тема: Розробка мобільного додатку для операційної системи iOS

Календарний план виконання роботи:

/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	жовтень	
2.	Огляд потрібних технологій.	листопад	
3.	Вивчення технологій.	листопад - грудень	
4.	Розробка мобільного застосунку.	січень - березень	
5.	Написання текстової частини.	квітень	
6.	Виправлення текстової частини.	13.05.2021	
7.	Створення презентації для доповіді	14.05.2021	
8.	Захист курсової роботи.	Згідно з розкладом ЕК	

Студент _____

Керівник _____

“_____”_____

Зміст

Анотація

Метою цієї роботи є створення мобільного застосунку, для людей, які не володіють особливими знаннями у сфері медицини та здоров’я, але хотіли б мати змогу, маючи рентген легень, самостійно дізнатись чи наявне у їх дітей захворювання такою хворобою як пневмонією і у висновку зменшити кількість смертей у дітей спричинених запалюванням легень. Описується створення такого додатку під операційну систему iOS, інструменти, що були використані в процесі та їх актуальність.

Вступ

Виходячи з поточного положення у світі стосовно захворювань пневмонією та тенденцією максимально задовольнити свої потреби з мінімальним відвідуванням місць зі скупченням людей, за мету даної роботи було поставлено розробити мобільний застосунок, за допомогою якого, людина, маючи рентген легень, зможе визначити чи наявна у її дитини віком до 5 років пневмонія.

Використання такого додатку зможе мінімізувати кількість походів по кабінетам у лікарні, де особливо підвищеній рівень ризику підхопити нове захворювання, але найголовніше це те, що також буде надана можливість оцінити рентгенівський знімок у місцях, далеких від кваліфікованого лікаря і тим самим скоротити час і мати приблизний діагноз як можна раніше, що може посприяти якомога раніше розпочати лікування.

Робота складається з трьох розділів.

Перший розділ був присвячений дослідженню.

Другий розділ був присвячений опису операційної системи та бібліотек.

Третій розділ присвячений реалізації додатку.

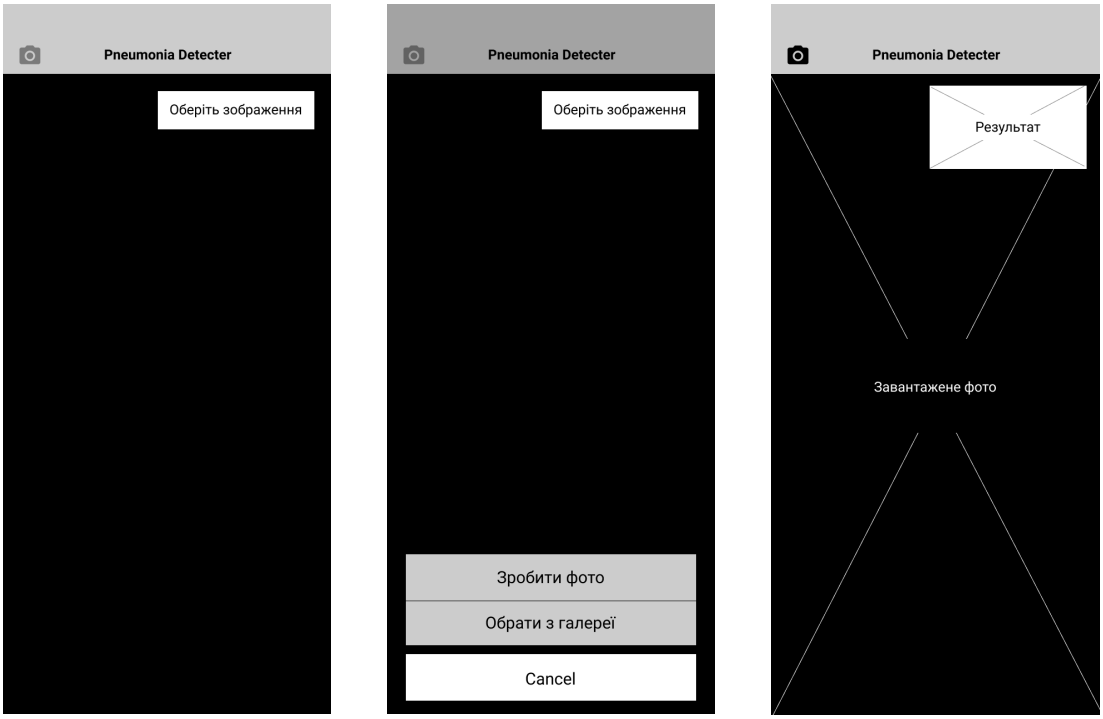
Розділ 1. Дослідження та аналіз предметної області

1.1. Пневмонія

Пневмонія – це захворювання тканин легень. Таке запалення зазвичай викликане інфекцією у разі чого вражаються альвеоли. Пневмонія вбиває більше дітей аніж будь-яка інша інфекційна хвороба. “У 2017 році, в Україні, процент смертей у дітей віком до 5 років був на рівні 8 відсотків” [1]. У Румунії це число дорівнює 23 відсоткам, у Китаї – 13.

1.2. Розбір потреб користувачів

У наш час користувач дуже вимогливий до будь-яких застосунків. Щоб мобільним додатком користувались він має бути максимально простим та зрозумілим користувачеві, а отже, потрібно позбавити його всіх можливих складнощів. Для людини, яка може не мати навіть базових речей у сфері медицини та здоров’я, потрібно зробити простий користувацький інтерфейс, за допомогою якого, вона зможе легко вирішити поставлену задачу, в цьому випадку – це, маючи рентген легень, визначити чи наявне захворювання пневмонією.



Рисунки 1.2.1, 1.2.2, 1.2.3 – прототип користувацького інтерфейсу

1.3. Класифікація зображення

Для того, щоб додаток зміг видати користувачеві відповідь, потрібно, за зробленим знімком, класифікувати зображення. Сама ж класифікація зображення – це модель машинного навчання, що ідентифікує зображення. Подаючи картинку на вхід, модель нам повинна видати відповідь з назвою категорії, яких у нашому випадку дві: «пневмонія» або «пневмонія відсутня».

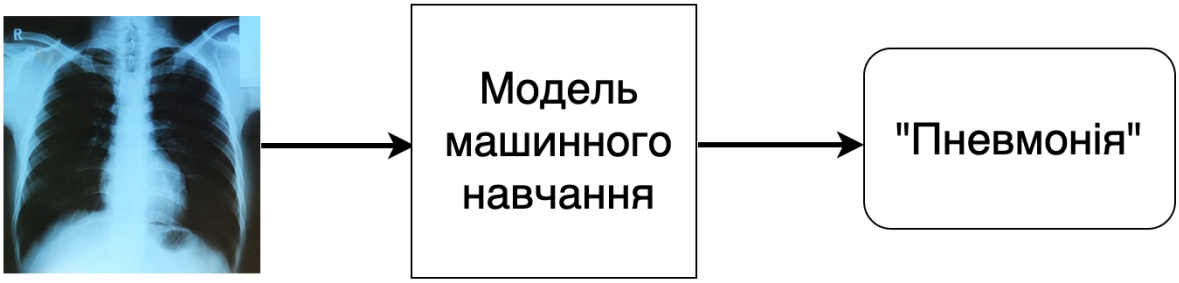


Рисунок 1.1 – Діаграма процесу класифікації

1.4. Набір даних

Для того щоб натренувати модель, яка буде класифікувати зображення, потрібно мати дані, на основі яких буде відбуватись навчання. Потрібно мати набір зображень як з пневмонією так і зі здоровими легенями і розділити на 3 категорії:

1) Тренування
2) Тестування
3) Оцінювання

Тренування – це найбільша, за кількістю зображень, категорія, використовуючи її відбувається саме навчання.

Тестування – це категорія, за допомогою якої ми тестуємо отриману модель, на даних, які раніше модель не бачила.

Оцінювання – категорія, за допомогою якої ми визначаємо оцінку коректності роботи отриманої моделі.

Розділ 2. Операційна система та технології

2.1. Операційна система iOS.

Операційна система iOS, також відома як iPhone OS, була випущена у червні 2007 році американською компанією Apple, що на той час була під керівництвом Стів Джобса. Також варто зазначити, що ця операційна система є Unix-подібною.

Компанія Apple дозволяє використовувати їх операційну систему лише в своїх мобільних пристроях, таких як: iPhone, iPad, iPod, Apple TV. Саме iOS поклала початок популярності такої технології як Multi-Touch.

Основною з переваг цієї операційної системи вважають її простоту, оскільки людям важливо щоб інтерфейс був зрозумілий і не виникало труднощів з базовими операціями. Користувачі також цінують безпечність та плавність роботи.

Якщо ж дивитись зі сторони розробників, то тут також є декілька переваг порівнюючи з розробкою додатків під інші операційні системи такі як Android та Windows Phone. По-перше, найбільшою перевагою можна вважати невелику кількість мобільних пристроїв, для яких пишеться програмне забезпечення. Це дає змогу легко протестувати фінальний продукт на всіх пристроях на яких можливе використання застосунку і одразу помітити якщо на якомусь екрані щось пішло не так. По-друге, не можна не сказати про стабільність iOS. Оскільки Apple розробляє свою операційну систему під свої пристрої, вони мають повний контроль над вихідним продуктом і знають, що коїться всередині, не маючи інших залежностей. По-третє, це повна забезпеченість якісними інструментами розробки, які є повністю безкоштовними і постійно оновлюються під потреби розробників. Також, американська компанія, описала рекомендації стосовно розробки користувацького інтерфейсу та виклала їх у відкритий доступ, щоб кожен розробник міг зробити свій застосунок простим.

2.2. Мова програмування

Говорячи про мову програмування для написання нативного застосунку під iOS, можна сказати, що тут вибір не такий великий, бо існує дві мови на яких може вестись нативна розробка.

Першою мовою якою писали застосунки під iPhone OS можна згадати Objective-C. Вона була створена на основі двох мов C та SmallTalk. Її можна назвати надійною, адже була протестована не одним поколінням розробників, бо була створена на кінці 80-х років [2]. З переваг можна перелічити наступні:

1) Надійна
2) На ній можна писати для ранніх версій iOS
3) Багата документація
4) Сумісна з C++

Говорячи про недоліки, можна згадати:

1) Важка читабельність
2) Складність вивчення
3) Складний синтаксис
4) Нижчий рівень безпеки
5) З кожним роком кількість розробників, які пишуть на Objective-C, стає все меншою.

Другою, і наразі найбільш популярною для розробки під iOS, є мова під назвою Swift. Її сміло можна назвати дружньою до вивчення, адже, на відміну від Objective-C, має простий та зрозумілий синтаксис. Також Swift не має залежностей від C і був розроблений для того щоб бути швидким. Якщо перераховувати головні його переваги, то не можна не згадати:

1) Швидкість
2) Безпечність
3) Читабельність

4) Знаходиться у відкритому доступі

Говорячи про недоліки, можна перелічити:

1) Повільна компіляція
2) Не підтримує ранні версії iOS

На сьогоднішній день, більшість з розробників, обираючи між цими двома мовами, обирають саме Swift, адже він виглядає більш перспективним та має набагато більші можливості.

2.3. Середовище розробки

Компанія Apple у червні 2008 року випустила середовище розробки під назвою Xcode. До того часу було неможливим створити додаток під iPhone OS стороннім розробникам і вони були вимушені створювати лише веб застосунки. Після появи програмного забезпечення, що дало змогу створювати мобільні додатки під пристрої Apple, почало з’являтись все більше якісних та цікавих застосунків. Важливо зазначити, що користуватись iOS SDK можливо лише на комп’ютерах що використовують операційну систему macOS, тобто користувач Windows не зможе розробляти додатки під iOS. Варто зазначити, що середовище розробки розроблене компанією Apple є безкоштовним.

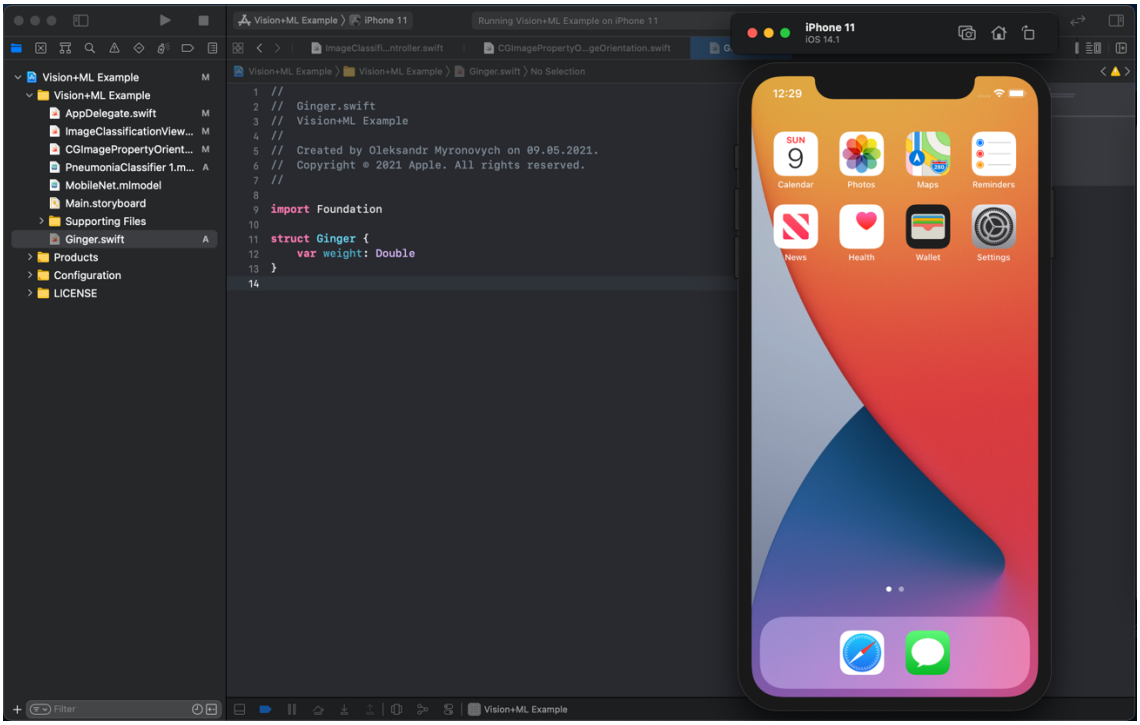


Рисунок 2.1 – середовище розробки Xcode

Серед опцій також потрібно виділити середовище розробки розроблене компанією JetBrains під назвою AppCode. У неї значне краще автодоповнення та рефакторинг, але вона не безкоштовна і навіть обравши AppCode як основне середовище потрібно буде використовувати Xcode для таких функцій як робота зі інтерфейсом в графічному редакторі під назвою Interface Builder та архівації проекту задля його подальшого розповсюдження.

Більшість розробників користуються саме Xcode. Через його переваги було вирішено використати саме його в цілях цієї роботи.

2.4. Огляд інструментів для машинного навчання

Інтерес до застосування машинного навчання на мобільних пристроях з часом все більше зростає. На перший погляд може здатись, що заради поєднання машинного навчання та мобільного застосунку потрібно спершу розібратись в багатьох аспектах:

1) Нейронні мережі

2) Декілька мов програмування
3) Глибоке знання математики
4) Поєднання користувацького інтерфейсу з результатами моделі

Можна згадати те, що зазвичай машинне навчання відбувається на сервері використовуючи такі технології як TensorFlow, PyTorch бо такі моделі можуть важити сотні мегабайт, робити величезну кількість обчислювань і потрібно опанувати зовсім інший світ, порівняно з мобільною розробкою.

У користувачів не завжди є інтернет з'єднання і тому дуже важливим є можливість провести обчислення на власному смартфоні. Робити висновок на мобільному пристрої, без підключення до серверу має наступні переваги:

- Швидка відповідь: не потрібно відправляти запити на сервер, що скорочує витрачений час та робить користувацький досвід більш приємним.
- Дані залишаються на пристрої користувача і не відправляються на сервери, що у висновку робить досвід користування конфіденційним.
- Для розробників, в свою чергу, не потрібно платити за використання серверів та переживати за конфіденційне збереження даних користувачів.

На сьогоднішній день з'являється все більша кількість інструментів, що допомагають спростити процес поєднання машинного навчання та мобільних пристроїв.

"Під систему iOS існує досить багато інструментів і продуктивність моделей набагато краща завдяки чіпу від Apple" [3].

Що ж стосується навчання на мобільному пристрої, то все ж таки ще існують різні обмеження пов'язані з технічними можливостями, що існують сьогодні. Навчання моделі вимагає наступних речей:

- Велика потужність обчислювальних машин
- Багато енергозатрат

2.4.1. Бібліотеки для машинного навчання від Apple

Американська компанія пропонує декілька опцій для роботи з машинним навчанням:

- Vision – для роботи з зображеннями
- Natural language – для роботи з текстом
- SoundAnalysis – для роботи з аудіо
- Speech – для переведення розмови в текст

2.4.2. Core ML

“Core ML дозволяє інтегрувати моделі машинного навчання в мобільний застосунок. Core ML надає уніфіковане представлення для всіх моделей. Застосунок використовує Core ML API і дані користувача щоб зробити прогноз на пристрої.” [4].

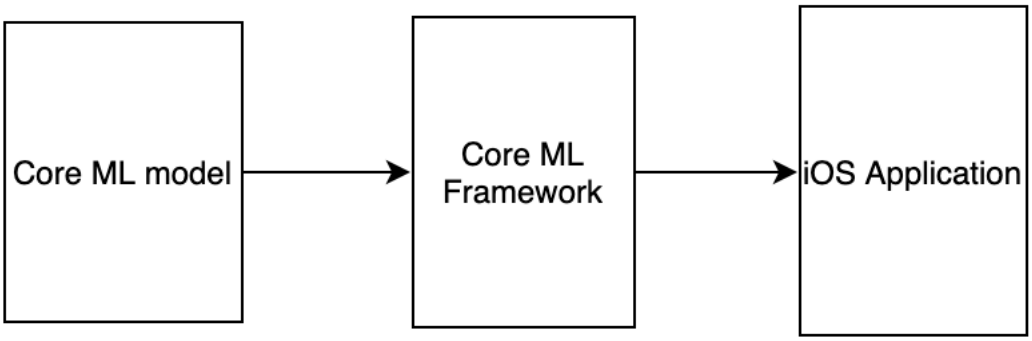


Рисунок 2.1 – діаграма роботи Core ML

2.4.3. Create ML

Create ML – це фреймворк, що дозволяє створити та натренувати модель не маючи ніяких особливих знань в сфері машинного навчання та нейронних мереж. Цей інструмент чудово підходить для такого завдання як

класифікація зображення і після створення моделі її можна буде застосувати в мобільному застосунку за допомогою Core ML.

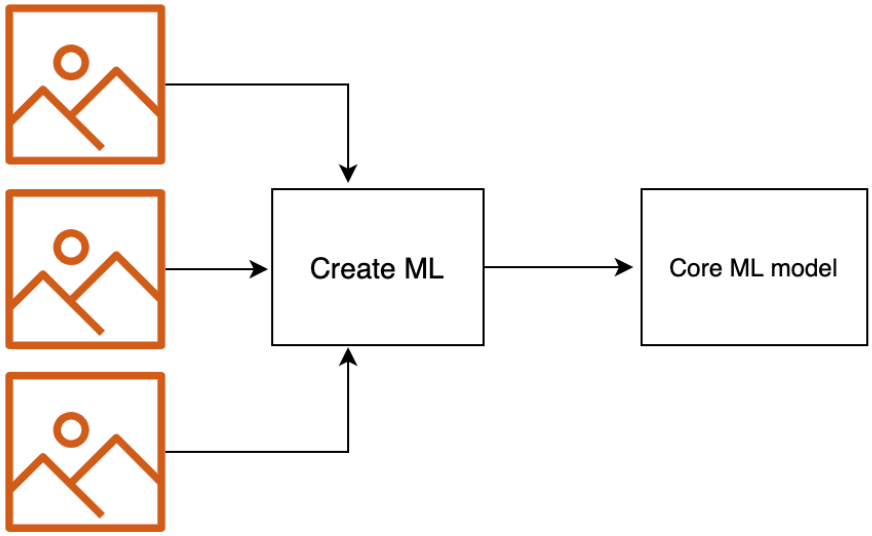


Рисунок 2.4.2.1 – діаграма застосування Create ML

2.4.4. Turi Create

"Turi Create – це бібліотека написана на мові Python, що використовується для створення Core ML моделей. Володіє простим та легким до використання API. Є кросплатформенним, працюючи під MacOS та Linux." [5].

2.4.5. Google ML Kit

Ця бібліотека – це аналог до бібліотеки Vision. За допомогою цього фреймворку виконується робота з зображеннями, але вона доступна і для iOS і для Android.

2.4.6. Вибір інструменту для створення додатку

Проаналізувавши доступні технології на ринку, можна обрати комбінацію двох технологій Core ML та Create ML. Маючи ці два могутні

інструменти можна без великих зусиль спочатку створити модель а потім інтегрувати цю модель в мобільний застосунок. Оскільки ці технології вбудовано в iOS SDK не потрібно буде додатково витрачати час та зусилля у під’єднані інших інструментів. Також важливо зазначити, що не потрібно буде вивчати іншу мову програмування та заглиблюватись у сферу машинного навчання.

Розділ 3. Розробка мобільного застосунку

3.1. Розробка моделі

Використовуючи Core ML, потрібно мати набір даних, за допомогою яких буде відбуватись навчання. В роботі було використано дата сет зображень, що були зібрані в Центрі жінки та дитини, що знаходиться в Гуанчжоу, у пацієнтів віком від 1 до 5 років [6]. Набір налічує 5856 зображення і поділений на 2 категорії:

1. Pneumonia
2. Normal

Відкривши Create ML, що вбудований в Xcode, з'явиться меню з набором опцій. У даній роботі, як згадувалось раніше, потрібно обрати Image Classification.

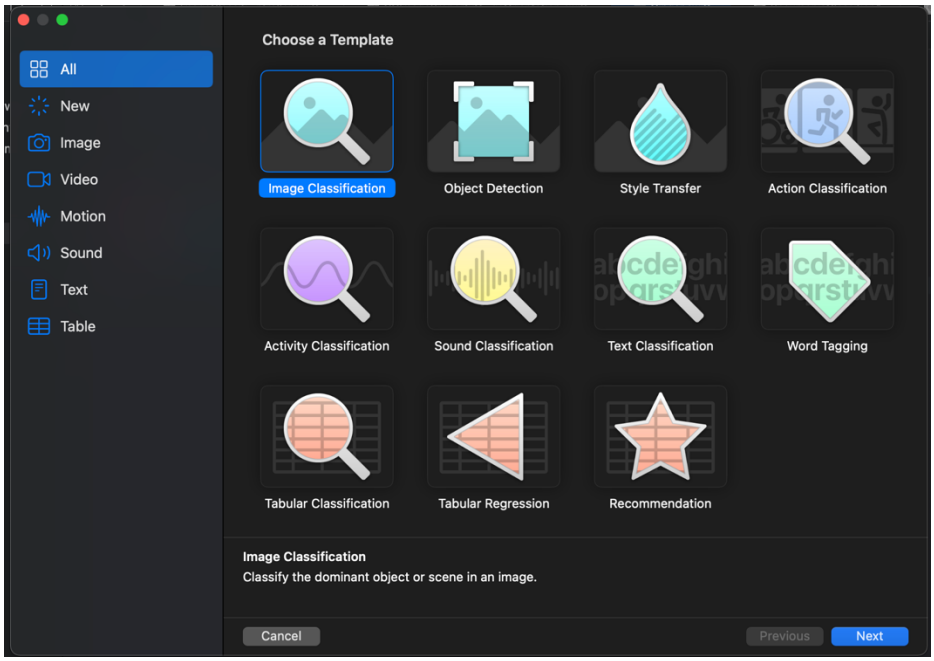


Рисунок 3.1.1 – опції Create ML

Після того як модель була навчена, проводиться оцінка, і якщо результат нас задовольняє то модель можна інтегрувати за допомогою Core ML.

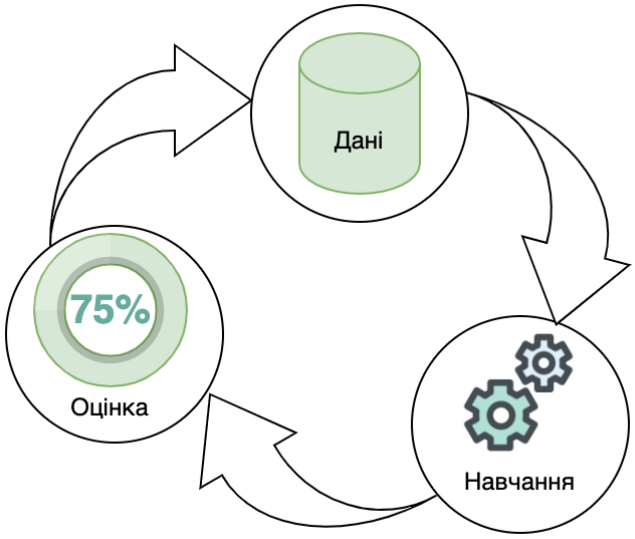


Рисунок 3.1.2 – Процес створення моделі

У висновку було отримано модель з наступними результатами:

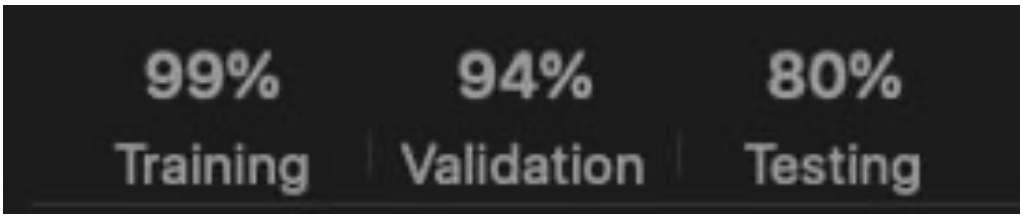


Рисунок. 3.1.3 - Результати навчання

3.2. Покращення моделі

Оскільки навчання проводилось на не збалансованому наборі даних, а саме кількість зображень з пневмонією досягала 3875, в той час як кількість зображень зі здоровими легенями дорівнювала 1341. Було вирішено скоротити кількість зображень з пневмонією до 1341, видаливши випадковим чином зайві зображення.

Таким чином було досягнено покращення моделі з наступними результатами:

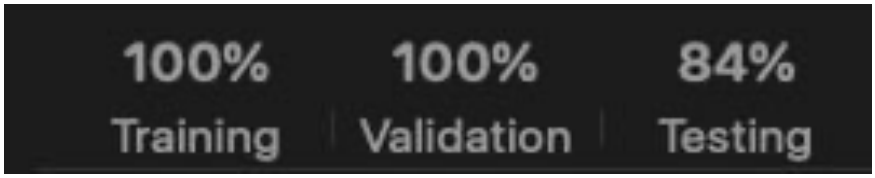


Рисунок. 3.2.1 - Результати після збалансованого набору даних

3.3. Спосіб зчитування зображення

Щоб дати користувачу відповідь на його запитання: «Чи наявне захворювання пневмонією на рентгеновому знімку?» - потрібно якимось чином зчитати це зображення. І тут у нас є дві опції:

- Обрати вже існуюче зображення з галереї
- Зробити знімок за допомогою камери пристрої

Ці функції реалізовані завдяки класу UIImagePickerController, що є контролером, який керує системним інтерфейсом для отримання фотографій, запису відео та отримання існуючих медіа даних з галереї користувача [7].

```
func presentPhotoPicker(sourceType: UIImagePickerControllerSourceType) {
    let imagePicker = UIImagePickerController()
    imagePicker.delegate = self
    imagePicker.sourceType = sourceType
    present(imagePicker, animated: true)
}

func imagePickerController(_ picker: UIImagePickerController,
    didFinishPickingMediaWithInfo info: [String: Any]) {
    picker.dismiss(animated: true)

    let image = info[UIImagePickerControllerOriginalImage] as! UIImage
    imageView.image = image
    classify(for: image)
}
```

Рисунок 3.2.1 – Частина коду, що відповідає за функціонал вибору фотографії

Функція presentPhotoPicker отримує параметр типу UIImagePickerControllerSourceType, що є перелічуваним типом даних та в залежності від бажаного варіанту користувача відкриває екран з вибором або створенням фотографії. Далі функція imagePickerController викликається коли було обрано/зроблено фотографію та викликає далі функціонал класифікації передаючи туди отриманий медіа об’єкт.

3.4. Контейнер для моделі машинного навчання

Core ML модель містить інформацію тренування з набору даних і потрібно використати клас VNCoreMLModel, що є контейнером для Core ML моделі та частиною бібліотеки Vision, щоб створити запит на обробку зображення [8].

3.5. Запит на аналіз зображення

Бібліотека Vision дає змогу створити запит на обробку зображення і отримати результат та провести дії, що були передані в замиканні до ініціалізації.

```
let model = try VNCoreMLModel(for: PneumoniaClassifier_1().model)

let request = VNCoreMLRequest(model: model, completionHandler: { [weak
    self] response, error in
    self?.startClassification(for: response, error: error)
})

return request
```

Рисунок 3.5.1 – Створення запиту на обробку зображення

3.6. Фінальний результат

В кінцевій версії роботи було отримано застосунок з простим та зрозумілим інтерфейсом. На тестових даних точність моделі, після процесу покращення, досягла 84 %.

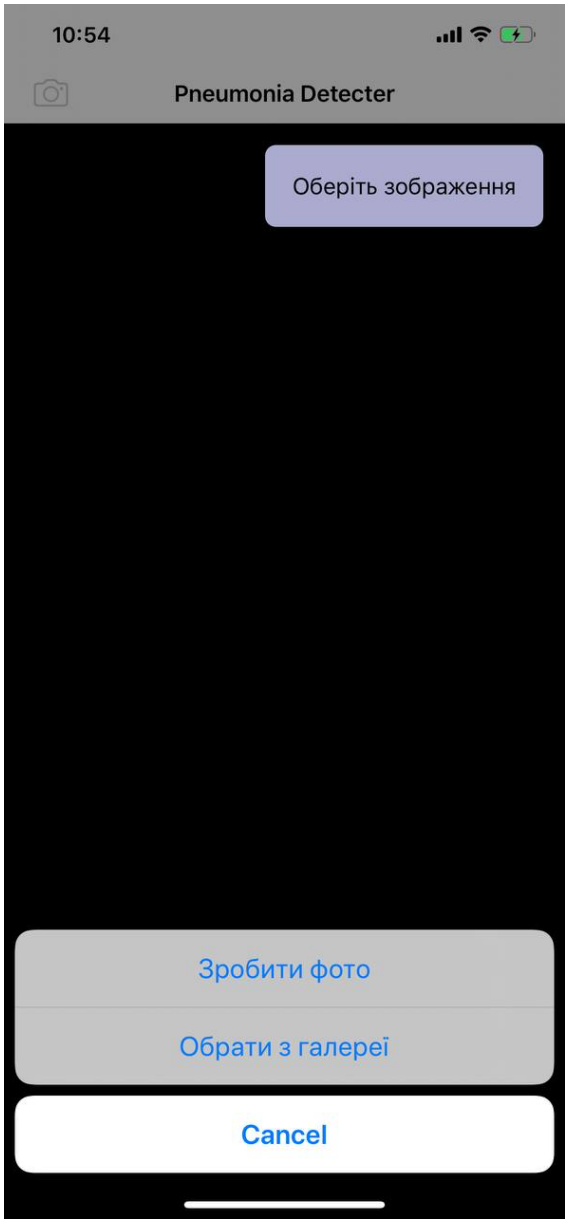


Рисунок 3.6.1 – інтерфейс вибору ресурсу зображення

Користувач має змогу обрати зображення з галереї або ж зробити фото рентгенівського знімку за допомогою камери (див. рис. 3.6.1).

Після обраного користувачем зображення проводиться класифікація та результат демонструється користувачеві на екрані разом з обраною фотографією (див. рис. 3.6.1).

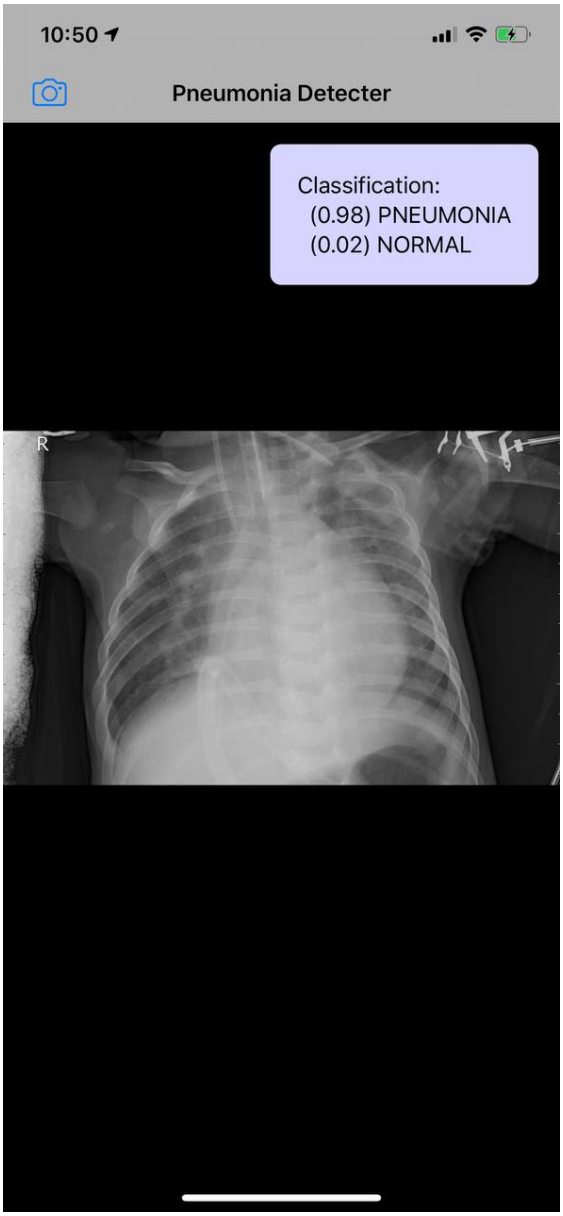


Рисунок 3.6.2 – Результат класифікації на екрані користувача.

Висновок

У цій роботі було розглянуто машинне навчання на мобільному пристрої з операційною системою iOS. Результатом роботи є мобільний застосунок, що класифікує зображення рентгену легень у дітей до 5 років і дає відповідь чи наявне захворювання пневмонією. У рамках цієї роботи вдалось досягти якості класифікації, на тестових даних, у 80 відсотків, але після оптимізації набору даних якість покращилась до 84 відсотків.

Було використано наступні технології:

- Swift
- Core ML
- Create ML
- Vision
- Xcode

У висновку також можна сказати, що уже існує досить багато інструментів для машинного навчання на мобільному пристрої і на сьогоднішній день, щоб натренувати модель, не обов’язково використовувати потужні сервери, а це можна зробити і на своєму домашньому комп’ютері.

Список використаних джерел

1. Pneumonia April 2021 [Електронний ресурс] – Режим доступу до ресурсу: https://data.unicef.org/topic/child-health/pneumonia/.
2. Lastovetska A. Swift vs Objective-C. Which iOS Language To Choose [Електронний ресурс] / Anastasiia Lastovetska. – 2021. – Режим доступу до ресурсу: https://mlsdev.com/blog/swift-vs-objective-c.
3. Kodra A. Machine learning on mobile: What can you actually do with it? [Електронний ресурс] / Austin Kodra. – 2019. – Режим доступу до ресурсу: https://heartbeat.fritz.ai/machine-learning-on-mobile-what-can-you-actually-do-with-it-8437fa782165.
4. Integrate machine learning models into your app. [Електронний ресурс] – Режим доступу до ресурсу: https://developer.apple.com/documentation/coreml.
5. Drawing Classification and One-Shot Object Detection in Turi Create [Електронний ресурс] – Режим доступу до ресурсу: https://developer.apple.com/videos/play/wwdc2019/420/.
6. Daniel K. Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification [Електронний ресурс] / K. Daniel, Z. Kang, G. Michael. – 2018. – Режим доступу до ресурсу: https://data.mendeley.com/datasets/rscbjbr9sj/2.
7. UIImagePickerController [Електронний ресурс] – Режим доступу до ресурсу: https://developer.apple.com/documentation/uikit/uiimagepickercontroller.
8. VNCoreMLModel [Електронний ресурс] – Режим доступу до ресурсу: https://developer.apple.com/documentation/vision/vncoremlmodel/.