

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій**

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інтегровані інформаційні системи»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Діагностична система для передбачення темпу зростання пухлин
на основі медичних зображень»**

Виконав:

студент IV курсу, групи ІА-94

Семеняк Євген Андрійович _____

Керівник:

Асистент

Вітюк Альона Євгеніївна _____

Рецензент:

зав. каф. СП ІПСА, проф., д.т.н.

Мухін Вадим Євгенович _____

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Семеняку Євгену Андрійовичу

1. Тема проєкту «Діагностична система для передбачення темпу зростання пухлин на основі медичних зображень», керівник проєкту Вітюк Альона Євгеніївна, асистент, затверджені наказом по університету від «31» травня 2023 р. №2101-с.
2. Термін подання студентом проєкту: 12 червня 2023 року
3. Вихідні дані до проєкту: мова програмування Python, фреймворки Numpy, Pandas, Sklearn, PyTorch та Streamlit, принципи, підходи, методи та технології прогнозування та розробки вебзастосунку
4. Зміст пояснювальної записки: Вступ, Аналіз предметної області, аналіз існуючих рішень, Проєктування системи та вибір технологій, Розробка додатку, Тестування системи.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): діаграма конволюційної нейромережі, діаграма послідовностей, діаграма класів.
6. Дата видачі завдання: 1 березня 2023 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Аналіз предметної області	17.04.2023	
2	Підготовка інструментарію	20.04.2023	
3	Завершення планування та пошук датасетів	29.04.2023	
4	Розробка нейронної мережі	05.05.2023	
5	Розробка серверної частини	15.05.2023	
6	Фіналізація розробки	23.05.2023	
7	Розробка діаграм	27.05.2023	
8	Оформлення документації	05.06.2023	
9	Подання готового проєкту	12.06.2023	

Студент

Євген СЕМЕНЯК

Керівник

Альона ВІТЮК

АНОТАЦІЯ

Семеняк Євген Андрійович Cancer diagnostic system вебзастосунок . КІІ імені Ігоря Сікорського, Київ, 2023.

Пояснювальна записка дипломного проекту складається з 4 розділів, містить 63 сторінок тексту, 16 рисунків, 1 таблицю, 1 додаток, 13 інформаційних посилань та 3 кресленики.

Ключові слова: cancer detection, вебзастосунок, Convolutional neural network, Python, Computer Vision, онкологічні утворення , розпізнавання зображень.

Об'єктом дослідження є діагностика злоякісних новоутворень.

Предметом дослідження є вебзастосунок для аналізу онкоутворень.

Метою дослідження є розробка вебзастосунку для розпізнавання злоякісних новоутворень.

У дипломному проекті реалізована можливість завантаження зображень вебзастосунку; отримання бінарних (позитивних/негативних) результатів.

Отримані результати можуть бути корисними як для користувачів, які бажають отримати результати стосовно своїх аналізів, так і для професійних лікарів які хочуть підтвердити або спростувати свої висновки отримані за допомогою спостереження неозброєним оком.

SUMMARY

Semenyak Yevhen Andriyovych Cancer diagnostic system web application.
KPI named after Ihor Sikorskyi, Kyiv, 2023.

The explanatory note of the diploma project consists of 4 chapters, contains 63 pages of text, 16 figures, 1 table, 1 appendix, 13 information links and 3 drawings.

Keywords: cancer detection, web application, Convolutional neural network, Python, Computer Vision, oncological formations, image recognition.

The object of the study is the diagnosis of malignant neoplasms.

The subject of research is a web application for the analysis of tumors.

The purpose of the research is to develop a web application for the recognition of malignant neoplasms.

In the diploma project, the possibility of downloading images of web applications has been implemented; receiving binary (positive/negative) results.

The obtained results can be useful both for users who want to receive results regarding their tests, and for professional doctors who want to confirm or refute their conclusions obtained by observing with the naked eye.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IA94.170БАК.004 ПЗ	Пояснювальна записка	60		
6	A3	IA94.170БАК.004 Д1	Діагностична система для передбачення темпу	1		
7			зростання пухлин на основі медичних зображень			
8			Діаграма конволюційної мережі			
9	A3	IA94.170БАК.004 Д2	Діагностична система для передбачення темпу	1		
10			зростання пухлин на основі медичних зображень			
11			Діаграма класів			
12	A3	IA94.170БАК.004 Д3	Діагностична система для передбачення темпу	1		
13			зростання пухлин на основі медичних зображень			
14			Діаграма послідовностей			
15	A3	IA94.170БАК.004 Д4	Діагностична система для передбачення темпу	1		
16			зростання пухлин на основі медичних зображень			
17			Діаграма use-case			
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						

IA94.170БАК.004 ТП

Зм.	Аркуш	№ докум.	Підпис	Дата				
Розроб.		Семеняк Є.А.			Діагностична система для передбачення темпу зростання пухлин на основі медичних зображень.	Літ.	Аркуш	Аркушів
Керівн.		Вітюк А.Є.				Т	1	1
Затв.					Пояснювальна записка	КПІ ім. Ігоря Сікорського Група ІА-94		

**Пояснювальна записка
до дипломного проєкту
на тему: «Діагностична система для передбачення
темпу зростання пухлин на основі медичних зображень»**

Київ – 2023 року

ЗМІСТ

<u>ВСТУП</u>	4
<u>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ</u>	6
<u>1.1 Змістовний опис і аналіз предметної області</u>	6
<u>1.2 Аналіз існуючих технологій та успішних ІТ-проектів</u>	9
<u>1.3 Постановка задачі</u>	11
<u>Висновки до розділу</u>	12
<u>2 ПРОЄКТУВАННЯ СИСТЕМИ ТА ВИБІР ТЕХНОЛОГІЙ</u>	13
<u>2.1 Огляд технологій та інструментів для створення CNN</u>	13
<u>2.2 Архітектура програмного забезпечення</u>	14
<u>2.3 Конструювання програмного забезпечення</u>	16
<u>2.4 Аналіз безпеки даних</u>	17
<u>Висновки до розділу</u>	18
<u>3 РОЗРОБКА ДОДАТКУ</u>	20
<u>3.1 Опис програмного забезпечення</u>	20
<u>3.2 Навчання та побудови моделі</u>	20
<u>3.3 Front-end частина</u>	36
<u>3.4 Опис контрольного прикладу</u>	40
<u>Висновки до розділу</u>	48
<u>4 ТЕСТУВАННЯ СИСТЕМИ</u>	49
<u>4.1 Повноти</u>	49
<u>4.2 Precision</u>	50
<u>4.3 F1-score</u>	51
<u>Висновки до розділу</u>	52
<u>ВИСНОВКИ</u>	54
<u>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</u>	60
<u>ДОДАТКИ</u>	61

					IA94.170BAK.004 ПЗ		
		№ докум.	Підпис				
Розробив	Семеняк Є.А.			Діагностична система для передбачення темпу зростання пухлин на основі медичних зображень. Пояснювальна записка	Літ.	Арк.	Аркушів
Перевірила	Вітюк А.Є				2		60
Затв.					КПІ ім. Ігоря Сікорського Група ІА-94		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AI	– Artificial Intelligence – Штучний інтелект.
API	– Application programming interface, прикладний програмний Інтерфейс
CNN	– Convolutional neural network – Згорткова нейромережа
SDK	– Software development kit
IT	– Інформаційні технології
ER	– Entity-Relation diagram
OC	– Операційна система.
БД	– База даних.

					ІА94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

ВСТУП

Одним з ключових напрямків сучасної медицини є використання штучного інтелекту (ШІ) для підвищення точності, швидкості та ефективності діагностики і лікування. В контексті постійного розвитку технологій обробки зображень та аналізу даних, розробка систем, що використовують машинне навчання та ШІ для аналізу медичних зображень, стає все більш актуальною.

Світові тенденції в галузі медицини вказують на все більшу інтеграцію ШІ в клінічну практику. Використання ШІ дозволяє значно покращити якість діагностики, включаючи виявлення патологій на медичних зображеннях. Така тенденція підкреслює важливість подальшого дослідження та розробки в цій області.

Об'єктом нашого дослідження є веб-застосунок, що використовує моделі глибокого навчання для аналізу медичних зображень, зокрема, знімків МРТ. В даний час вже існують деякі системи, які використовують ШІ для обробки та аналізу медичних зображень, проте має місце невідповідність між потенціалом цих технологій та їх практичною реалізацією.

Деякі провідні наукові установи та організації вже розробляють та використовують ШІ для аналізу медичних зображень. Наприклад, Google Health розробляє алгоритми ШІ для виявлення раку грудей на мамограмах, що демонструє потенціал таких технологій у підвищенні точності діагностики. Тим не менше, їхня система все ще потребує висококваліфікованого медичного персоналу для подальшого аналізу та перевірки результатів, що підкреслює невід'ємну роль людини у процесі.

IBM Watson Health — ще один приклад компанії, яка використовує ШІ для аналізу медичних зображень. Вони створили систему, що може допомогти в ідентифікації та інтерпретації медичних зображень. Однак, система Watson Health зіткнулась з викликами у точності прогнозування, що ставить під сумнів її надійність у реальних клінічних сценаріях.

					ІА94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

Aidos, стартап з Ізраїлю, використовує ШІ для надання допомоги радіологам у виявленні невідкладних випадків, таких як кровотеча в мозку, на КТ-знімках. Ця система показала високу точність, але її використання обмежено лише окремими випадками, і вона не може замінити повне оглядове сканування, яке виконує радіолог.

Застосування розроблених моделей ШІ у веб-додатках або мобільних застосунках може сприяти їх широкому застосуванню та доступності для медичних фахівців та пацієнтів. Такі рішення можуть допомогти медичним працівникам отримувати більш точні результати діагностики швидше, що, в свою чергу, може покращити загальну якість медичної допомоги.

Можливі сфери застосування таких систем включають, але не обмежуються, раннє виявлення патологій, оптимізацію процесів діагностики, моніторинг стану пацієнтів під час лікування та навіть планування хірургічних втручань.

При цьому, необхідно підкреслити, що використання ШІ в медицині потребує особливої уваги до питань етики, конфіденційності даних та їх безпеки. Оскільки медичні дані є чутливою інформацією, необхідно забезпечити їх належне зберігання, обробку та передачу.

Отже, розробка веб-застосунку, що використовує ШІ для аналізу медичних зображень, є актуальним та перспективним напрямком, який відповідає світовим тенденціям і може зробити значний внесок у поліпшення якості медичних послуг.

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Штучний інтелект (ШІ) вже використовується в багатьох галузях, включаючи медицину, де він пропонує нові та унікальні можливості для діагностики та лікування різноманітних захворювань. Машинне навчання, підгалузь ШІ, що зосереджена на розробці систем, здатних вчитися та вдосконалюватися без прямого програмування, зокрема активно використовується в сучасній медичній практиці .

Конволюційні нейронні мережі відкривають широкі можливості для розв'язання складних задач, які включають аналіз зображень, розпізнавання образів, автоматичне оцінювання, сегментацію зображень, і багато іншого. Вони стають незамінними там, де традиційні методи обробки даних не можуть забезпечити необхідну гнучкість, швидкість обробки, точність або інтуїтивно зрозумілі результати.

Конволюційні нейронні мережі дозволяють комп'ютерам "бачити" і розуміти зображення на дуже деталізованому рівні, впроваджуючи ієрархічний підхід до розпізнавання шаблонів та концепцій.

1.1 Змістовний опис і аналіз предметної області

Конволюційні нейронні мережі (Convolutional Neural Networks, CNN) - це тип глибоких нейронних мереж, що були спеціально розроблені для обробки даних з такою ґратовою структурою, як зображення. CNN використовують математичну операцію, відому як конволюція, яка систематично застосовується до вхідних даних. Це дозволяє мережі автоматично та адаптивно вчити просторові ієрархії ознак.

Фільтр/Ядро (Filter/Kernel): Це невелика матриця, що використовується при конволюції. Різні фільтри можуть виявляти різні типи ознак на зображеннях, такі як краї, лінії, кольори та інші.

					ІА94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

Конволюція (Convolution): Основна операція в конволюційних нейронних мережах. Це математична операція, що включає елементне множення двох матриць. В контексті обробки зображень, одна матриця представляє зображення, а інша - фільтр або ядро. Результатом цієї операції є нове зображення, де кожне значення відображає інформацію про локальний контекст вихідного зображення.

Карта ознак (Feature Map): Це зображення, що виходить після застосування конволюції до вхідного зображення за допомогою фільтра. Кожен піксель в карті ознак відображає інформацію про відповідну область вихідного зображення.

Функція активації (Activation Function): Функції активації використовуються для додавання не лінійності в нейронну мережу. Вони дозволяють моделі вчитися і адаптуватися до складних шаблонів. Найчастіше використовується функція Rectified Linear Unit (ReLU), яка виводить вхідне значення, якщо воно позитивне, або нуль, якщо воно негативне.

Pooling (або Subsampling): Операція, що використовується для зменшення просторових розмірів карт ознак. Це робиться шляхом вибору найбільшої (max pooling) або середньої (average pooling) величини з невеликої підмножини пікселів.

Fully Connected Layer (Повністю з'єднаний шар): Цей шар використовується в кінці мережі, після декількох шарів конволюції та пулінгу. Він з'єднує кожен нейрон в одному шарі з кожним нейроном в наступному шарі.

Padding: Додавання нулів до вхідного зображення по краях. Це дозволяє контролювати просторові розміри вихідного зображення після конволюції.

Stride: Крок, з яким фільтр пересувається по вхідному зображенні під час операції конволюції. Stride рівний 1 означає, що фільтр зміщується на 1 піксель кожного разу, коли він пересувається. Більший stride приводить до зменшення просторових розмірів вихідної карти ознак.

Використання знань предметної області в програмному забезпеченні для аналізу медичних зображень на сучасному етапі розвитку ІТ-технологій зосереджено на використанні алгоритмів машинного навчання та штучного інтелекту. На даний момент, найбільш поширеними є методи глибокого навчання, зокрема конволюційні нейронні мережі, для обробки медичних зображень.

					ІА94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		7

Однак, незважаючи на значний прогрес у цій області, існують декілька важливих викликів та обмежень, які потребують подальшого дослідження та розвитку.

Перше — це великі вимоги до обчислювальних ресурсів для тренування та роботи моделей глибокого навчання. Вони вимагають великих обсягів даних та високої обчислювальної потужності, що може бути проблемою для медичних установ, які не мають доступу до великих обчислювальних центрів.

Друге — це проблема переносимості та узагальнення моделей. Моделі, які були натреновані на одних даних, можуть показувати погані результати на інших даних через варіативність медичних зображень та патологій.

Третє — це виклик етики та конфіденційності даних. Медичні дані є чутливою інформацією, і використання їх у машинному навчанні вимагає належного зберігання, обробки та захисту.

Четверте — необхідно зазначити важливість інтерпретації моделей машинного навчання. Для більшої довіри медичних спеціалістів до використання таких технологій, моделі машинного навчання мають бути не просто ефективними, але й прозорими у своїх рішеннях.

П'яте — існує виклик щодо розробки інтуїтивно зрозумілих та простих в користуванні інтерфейсів для веб-застосунків, які використовують машинне навчання для аналізу медичних зображень. Оскільки кінцевими користувачами таких систем зазвичай є медичні фахівці, а не спеціалісти з ІТ, дизайн інтерфейсу повинен бути простим, зрозумілим і зручним для використання.

В рамках даного проекту буде розроблено не лише алгоритм для аналізу медичних зображень, а й створити простий та інтуїтивний веб-інтерфейс для його використання. Також планується забезпечити прозорість та інтерпретацію результатів нашої моделі, щоб забезпечити довіру та зрозуміння з боку медичних фахівців.

1.2 Аналіз існуючих технологій та успішних ІТ-проектів

Проаналізуємо відоме на сьогодні алгоритмічне забезпечення у даній області та технічні рішення, що допоможуть у реалізації проекту на тему «діагностична система для виявлення онкологічних пухлин на основі медичних зображень».

Існує ряд існуючих технологій та успішних ІТ-проектів, які можна використовувати як вихідний пункт або навіть основу для нашого проекту.

DeerLab: DeerLab використовує технологію атрибутної дилатації конволюції, щоб дозволити мережі "дивитися" на більш широкі області зображення. Це робить її дуже ефективною для вирізання об'єктів на зображеннях. DeerLab була розроблена в Google та написана на TensorFlow, глибокому фреймворку навчання від Google, який забезпечує високу продуктивність та масштабованість.

U-Net: U-Net використовує концепцію енкодера та декодера у своїй архітектурі. Енкодер відповідає за "звуження" вхідного зображення, зберігаючи при цьому важливу інформацію про об'єкти на ньому. Декодер здійснює "розширення" цієї стисненої інформації, відтворюючи оригінальне зображення, але з певними об'єктами вирізаними. U-Net може бути реалізований в багатьох популярних фреймворках машинного навчання, включаючи TensorFlow і PyTorch.

Fast.ai: Fast.ai - це високорівнева бібліотека, яка побудована на PyTorch. Вона надає прості високорівневі інтерфейси для багатьох складних операцій машинного навчання, що дозволяє розробникам створювати та тренувати складні моделі з невеликою кількістю коду. Більше того, fast.ai активно підтримує "transfer learning", метод, який дозволяє використовувати попередньо навчені моделі як вихідний пункт для нових завдань.

Google's AutoML: Google's AutoML - це набір сервісів машинного навчання, що автоматизують процес створення моделей машинного навчання. Зокрема, AutoML Vision може бути використаний для автоматичного створення моделей розпізнавання зображень, включаючи моделі для сегментації зображень. AutoML

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		9

використовує технологію переносу навчання, а також автоматичне навчання, щоб вибрати найкращі моделі та гіперпараметри для заданого набору даних.

Усі ці технології мають свої переваги та недоліки, і вибір між ними залежатиме від специфіки конкретного проєкту.

Аналіз алгоритмів для розпізнавання патологій на медичних зображеннях:

Конволюційні нейронні мережі (CNN): Це тип штучних нейронних мереж, що використовується для обробки зображень. Вони здатні вчити і витягувати важливі особливості зображення через процес конволюції та пулінгу. Це дозволяє мережі витягувати важливу інформацію з зображення, не втрачаючи деталей через зменшення розміру.

Переваги: CNN володіють глибоким рівнем уявлення про об'єкти на зображеннях, здатні адаптуватися до різних перетворень і здатні автоматично навчатися ієрархічних особливостей, що забезпечують відмінні результати в багатьох задачах розпізнавання образів.

Недоліки: Велика потреба в обчислювальних ресурсах, велика кількість параметрів для налаштування, можуть виникати проблеми з перенавчанням.

Сегментаційні алгоритми (такі як U-Net): Ці алгоритми використовуються для визначення областей або сегментів зображення, які містять певний об'єкт або особливість. Вони особливо корисні для медичних зображень, де важливо точно визначити область, що містить патологію.

Переваги: Дуже ефективні для точної локалізації об'єктів на зображеннях. U-Net особливо ефективний для медичних зображень, оскільки він був спеціально розроблений для цієї цілі.

Недоліки: Потребують великої кількості анотованих даних для навчання, можуть мати високі вимоги до обчислювальних ресурсів.

Ансамблі алгоритмів: Ансамбліві алгоритми використовують комбінації різних моделей для отримання більш точних прогнозів. Вони можуть включати комбінації різних типів моделей, таких як CNN і U-Net, або можуть використовувати ту саму модель з різними параметрами.

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		10

Переваги: Здатні підвищити точність прогнозування, зменшити вплив перенавчання. Ансамблі моделей, як правило, дають кращі результати, ніж окремі моделі. Недоліки: Потребують більше обчислювальних ресурсів, тривалий процес навчання, може бути складно інтерпретувати результати.

Техніки збільшення даних: Ці методи використовуються для створення додаткових тренувальних даних шляхом модифікації існуючих даних. Наприклад, зображення можуть бути повернуті, віддзеркалені або спотворені, щоб створити додаткові варіанти.

1.3 Постановка задачі

Враховуючи специфіку задачі та доступні ресурси, ми вибираємо комбінацію CNN і U-Net для створення моделі. Це дозволить нам використовувати переваги обох методів: ефективне витягування характеристик з CNN і точну локалізацію об'єктів за допомогою U-Net. Ми також використовуватимемо техніки збільшення даних для збільшення розміру нашого набору даних.

В даному проекті планується використовувати комбінацію цих технік. В основному, ми будемо використовувати модель, засновану на конволюційній нейронній мережі для визначення патологій на зображеннях. Модель буде навчена за допомогою набору зображень з відомими патологіями. Ми також плануємо використовувати техніки збільшення даних для збільшення кількості доступних для навчання даних.

Задачею даної роботи є оптимізація та автоматизація процесу навчання конволюційних нейронних мереж (CNN) для виконання задач класифікації зображень з визначеною точністю. Основна мета полягає в розробці ефективної методики навчання, яка забезпечить високу точність розпізнавання при мінімальній кількості ітерацій.

У ході дослідження планується використовувати передові техніки глибокого навчання та алгоритми оптимізації для адаптації та точне налаштування конволюційних нейронних мереж. Необхідно також реалізувати різні техніки

збільшення даних та оптимізації архітектури CNN для підвищення загальної продуктивності та точності моделі.

Результатом буде програмний продукт, що дозволить задавати архітектуру CNN, дані для навчання та тестування, бажану точність, максимальну кількість ітерацій, конфігурацію шарів мережі та різні параметри методів оптимізації. В результаті роботи програми користувач отримає навчену конволюційну нейронну мережу з визначеною точністю та візуалізацію процесу навчання.

Висновки до розділу

Глибоке вивчення предметної області показало, що CNN є одним із передових методів у галузі машинного навчання для аналізу візуальних даних. Використовуючи свої унікальні властивості - здатність автоматично і ієрархічно вчиться на різних рівнях абстракції - вони дозволяють виявляти складні шаблони безпосередньо з сировинних даних.

Проаналізувавши аналіз існуючих технологій та успішних ІТ-проектів в цій сфері підкреслив важливість інноваційного підходу до процесу навчання CNN. Сучасні методи, такі як генетичні алгоритми, використовуються для автоматичного підбору оптимальних ваг зв'язків, що може значно скоротити кількість потрібних ітерацій навчання, при цьому зберігаючи високий рівень точності прогнозування.

Спостерігаючи прийшов до висновку, що полягає у тому, що для ефективної роботи з CNN необхідно мати гнучкі інструменти для налаштування та конфігурації. Це стосується всіх аспектів процесу навчання, включаючи архітектуру мережі, параметри оптимізації, техніки аугментації даних та інше.

Таким чином, на основі проведеного аналізу вимог, виходить, що розробка програмного забезпечення, що задовольнятиме ці потреби, є не лише актуальною, але й має великий потенціал для внесення значного вкладу в галузь комп'ютерного зору та глибокого навчання, покращуючи можливості класифікації зображень та сприяючи прогресу в цій важливій області.

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		12

2 ПРОЄКТУВАННЯ СИСТЕМИ ТА ВИБІР ТЕХНОЛОГІЙ

2.1 Огляд технологій та інструментів для створення CNN

Створення та використання конволютивної нейронної мережі (CNN) вимагає розуміння різних технологій, інструментів, процесів і типів даних. Нижче наведено детальний огляд цих аспектів.

TensorFlow і Keras: TensorFlow є однією з найпопулярніших бібліотек для створення нейронних мереж, а Keras - це високорівневий API, який працює поверх TensorFlow, що спрощує створення та тренування моделей.

PyTorch: Це інша популярна бібліотека для глибокого навчання, яка дуже гнучка та має сильну підтримку досліджень.

Підготовка даних: Це включає збір, очищення, нормалізацію, збільшення та інші форми передпрацювання даних. Важливо враховувати, що дані мають бути максимально близькими до даних, на яких модель буде використовуватися.

Навчання моделі: Це включає вибір архітектури мережі, вибір функції втрат, оптимізатора, а також вибір гіперпараметрів.

Оцінка та тюнінг моделі: Після навчання моделі потрібно її оцінити на валідаційному наборі даних, а потім налаштувати гіперпараметри, щоб зменшити перенавчання або підвищити точність.

CNN може працювати з різними типами зображень. Вони можуть обробляти кольорові зображення (RGB), чорно-білі зображення (градації сірого), багатоканальні зображення (наприклад, зображення медичних знімків або супутникових зображень) або навіть 3D зображення.

Перед початком роботи важливо уважно придивитись до низки аспектів, які суттєво впливають на ефективність проекту. Нижче наведені декілька найбільш значущих.

Якість даних: Це критично важливий фактор. Незалежно від того, як потужна ваша модель, якщо дані, на яких вона навчається, є неякісними або недостатньо представницькими, результати можуть бути поганими.

Перенавчання: Це поширена проблема при навчанні нейронних мереж. Її можна контролювати за допомогою технік, таких як рання зупинка, регуляризація, використання випадкового відключення (dropout) та збільшення даних.

Обчислювальні ресурси: Навчання CNN вимагає значних обчислювальних ресурсів. Якщо доступ до таких ресурсів обмежений, може бути корисним використовувати предобучені моделі та тонку настройку.

Розуміння архітектури: Розуміння того, як працює CNN, дозволить вам зрозуміти, як вона інтерпретує зображення, а також допоможе вам при розробці або виборі архітектури моделі.

2.2 Архітектура програмного забезпечення

Опис архітектури програмного забезпечення дозволяє детально розглянути структуру та роботу проекту. Для проекту, що базується на згортковій нейронній мережі (CNN), можна розглянути архітектуру як набір з'єднаних компонентів, кожен з яких виконує визначену функцію.

Для розуміння цієї архітектури розглянемо основні компоненти CNN:

Вхідний шар: Цей шар приймає вхідні дані, в нашому випадку - це зображення. Він перетворює зображення у вектори пікселів, які можуть бути подані на вхід нейронній мережі.

Конволюційні шари: Ці шари відповідають за виявлення ознак на вхідних зображеннях. Вони виконують операції конволюції над вхідними даними за допомогою набору ваг, названих "фільтрами" або "ядрами". Результатом цього процесу є карта ознак, яка показує, де і які ознаки були знайдені на зображенні.

Шари активації: Ці шари використовують функції активації для введення нелінійності в модель. Це допомагає мережі вивчити складніші взаємозв'язки в даних.

Пулінгові шари: Ці шари використовуються для зменшення просторового розміру вхідних даних. Це допомагає зменшити кількість параметрів моделі і запобігає перенавчанню.

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		14

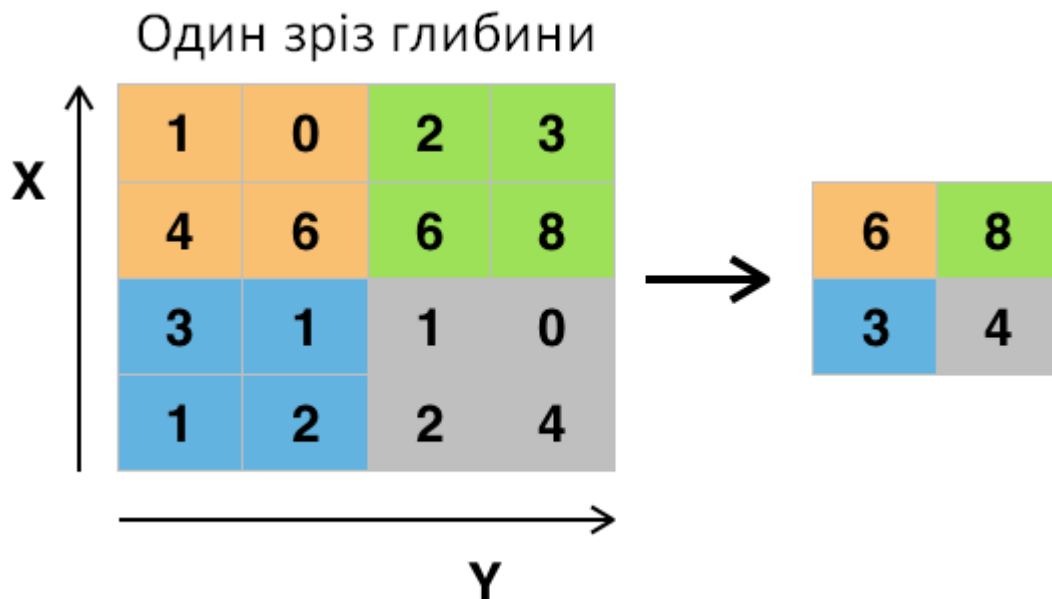


Рисунок 2.1 – Максимізаційне агрегування (англ. max pooling) із фільтром 2×2 та кроком = 2

Повністю з'єднані шари: На останній стадії, після кількох конволюційних і пулінгових шарів, вхідні дані проходять через один або кілька повністю з'єднаних шарів. Ці шари використовуються для вивчення не лінійних комбінацій ознак, виявлених на попередніх етапах, і виконують класифікацію або регресію на основі них.

Вихідний шар: Цей шар видає кінцевий результат мережі. Для задач класифікації, це зазвичай шар softmax, який видає ймовірності для кожного класу.

Архітектура нашого програмного забезпечення буде складатися з декількох вищезгаданих шарів, розташованих у певному порядку. Крім того, вона включатиме в себе модуль для автоматичного навчання та оптимізації з використанням генетичного алгоритму, що буде відповідати за підбір оптимальних ваг.

Таким чином, наша архітектура ПЗ буде інтегрованою системою для обробки вхідних зображень, їхнього аналізу за допомогою CNN, і подальшого навчання та оптимізації моделі.

2.3 Конструювання програмного забезпечення

З технічної точки зору, для розробки даного проекту ми обрали Python як основну мову програмування через її зручність і підтримку великої кількості наукових та машинного навчання бібліотек, таких як TensorFlow, Keras, PyTorch і Scikit-Learn. Для веб-інтерфейсу буде використовуватися Streamlit, що дозволяє швидко та ефективно створювати веб-застосунки для машинного навчання. Для роботи з зображеннями буде використана бібліотека OpenCV.

Python є найпопулярнішою мовою для наукових обчислень і машинного навчання. Це мова зі зрозумілим синтаксисом, яка підтримує різні парадигми програмування, такі як об'єктно-орієнтоване, процедурне та функціональне програмування. Python також має велику кількість бібліотек для обробки даних, візуалізації, машинного навчання та іншого, що робить його незамінним інструментом для нашого проекту.

TensorFlow – це відкритий фреймворк машинного навчання, який розроблений Google. Він дозволяє створювати складні моделі нейронних мереж з багатьма шарами.

Keras є високорівневою обгорткою над TensorFlow, що спрощує процес створення моделей машинного навчання. Разом вони надають гнучкість та зручність, необхідні для нашого проекту.

Pandas – це бібліотека Python для обробки та аналізу даних. Вона надає структури даних, які зручно використовувати для роботи з реляційними даними.

Numpy – це бібліотека для наукових обчислень в Python. Вона надає підтримку масивів високого рівня, що зручно використовувати для математичних обчислень..

Matplotlib – це бібліотека Python для створення статичних, анімованих і інтерактивних візуалізацій в Python.

Seaborn – це високорівнева бібліотека візуалізації, яка будується на базі Matplotlib і дозволяє створювати більш сучасні та стильні візуалізації.

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		16

Середовища розробки (IDE):

PyCharm: Це професійне середовище розробки Python, яке надає широкий набір інструментів для розробки високоякісного коду.

Jupyter Notebook: Jupyter Notebook – це інтерактивне середовище розробки, яке дозволяє створювати та ділитися документами, що містять живий код, формули, візуалізації та текст. Завдяки його інтерактивності та гнучкості, Jupyter особливо популярний серед науковців даних.

2.4 Аналіз безпеки даних

У контексті проекту, що використовує конволютивні нейронні мережі (CNN) та Streamlit для розробки додатка, що виявляє меланому на зображеннях шкіри, основні питання безпеки та вразливості пов'язані з наступними областями:

Зберігання даних: Важливо захищати дані, які використовуються для навчання та тестування моделей. Це могли би бути зображення шкіри пацієнтів, що містять особисту та чутливу інформацію. Необхідно забезпечити захист від несанкціонованого доступу до цих даних.

Передача даних: Передача даних між клієнтом (браузером) та сервером (додатком Streamlit) повинна бути захищена. Використання протоколу HTTPS для захисту даних під час передачі - це хороша практика.

Безпека програмного забезпечення:

Вразливості бібліотек: Бібліотеки, які використовуються у вашому проекті (наприклад, PyTorch, TensorFlow, Streamlit), можуть мати власні вразливості. Важливо регулярно оновлювати ці бібліотеки до найновіших версій, що виправляють відомі вразливості.

Завантаження моделей: Якщо ви завантажувате предобучені моделі з Інтернету, важливо переконаватися, що вони завантажуються з надійних джерел.

Безпека моделі:

Викривлення моделі: Модель CNN може бути змушена давати неправильні висновки через навмисно викривлені вхідні дані. Це є частиною більш широкої

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		17

проблеми з безпекою машинного навчання, що називається "завадливими атаками". Наразі існує багато методів для захисту від цих атак.

Конфіденційність моделі: Якщо модель навчалася на конфіденційних даних, є ризик того, що зловмисники можуть витягнути ці дані з моделі через "атаки на вивчення моделі". Це включає техніки, такі як обратне інженерне проектування або атаки типу "чорний ящик".

Загалом, важливо застосовувати найкращі практики в області безпеки при розробці та впровадженні вашого проекту, зокрема при роботі з чутливими даними.

Висновки до розділу

Було розглянуто основні етапи, технології та виклики, що стосуються розробки програмного забезпечення для проекту з використанням конволютивних нейронних мереж (CNN) та платформи Streamlit.

Було досліджено інструменти та бібліотеки, необхідні для створення та тренування моделей CNN. Ми розглянули важливість вибору відповідних даних для тренування і тестування моделі, а також особливості підготовки цих даних.

У підрозділі "Архітектура програмного забезпечення" ми розкрили структуру та архітектуру нашого програмного рішення, що включає в себе модель CNN та веб-додаток, створений за допомогою Streamlit.

Під час "Конструювання програмного забезпечення" було описано основні етапи розробки нашого додатку, включаючи налаштування середовища, реалізацію моделі CNN та інтеграцію моделі з веб-додатком.

В рамках "Аналізу безпеки даних" ми розглянули ключові аспекти безпеки, на які необхідно звернути увагу під час розробки та впровадження проекту. Особливу увагу ми приділили захисту даних, безпеці програмного забезпечення та безпеці моделі.

Моделювання та конструювання програмного забезпечення для даного проекту, що базується на використанні конволютивних нейронних мереж (CNN) та

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		18

платформи Streamlit, включало дослідження та впровадження ряду технологічних рішень.

Найсуттєвішим було підготовка та обробка даних для подальшого тренування моделі. Без правильної підготовки і відповідного розуміння даних, модель CNN може бути неефективною або навіть некоректною. Це підкреслює важливість таких етапів, як збір, обробка та аугментація даних.

Архітектура програмного забезпечення була ретельно розроблена для впевненості в його масштабованості та ефективності. Модель CNN та веб-додаток Streamlit були спроектовані таким чином, щоб гармонійно доповнювати один одного, що дозволило створити цілісний та зручний для користувача продукт.

Важливим елементом конструювання програмного забезпечення була інтеграція різних компонентів та впевненість у їхній згуртованості. Враховуючи складність і велику кількість технологічних компонентів, було важливо забезпечити їх правильне поєднання та взаємодію.

Безпека даних була визнана критично важливим аспектом у розробці та впровадженні цього проекту. Було проаналізовано та впроваджено стратегії для захисту даних, безпеки програмного забезпечення та безпеки моделі, що є важливим кроком для забезпечення надійності та захищеності системи.

Загалом, успіх в моделюванні та конструюванні програмного забезпечення, який включає в себе важливі елементи якості даних, архітектуру системи, злагодженість компонентів та безпеку даних, вимагає високого рівня розуміння та використання відповідних технологій та методик. Всі ці елементи були враховані при розробці даного проекту.

В цілому, дані етапи та аспекти є критично важливими для успішного впровадження та експлуатації нашого програмного рішення. Розробка і впровадження подібного проекту вимагає глибокого розуміння технологій машинного навчання, веб-розробки та принципів безпеки.

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		19

3 РОЗРОБКА ДОДАТКУ

3.1 Опис програмного забезпечення

У процесі розробки програмного продукту було враховано принципи чистого коду та розробки з орієнтацією на тестування, що дозволило створити продукт високої якості, що відповідає всім поставленим вимогам.

Загалом задача має масштабуватись до всіх існуючих видів онкологічних утворень, але реалізація буде прототипу на прикладі меланоми.

3.2 Навчання та побудови моделі

Початок побудови моделі це вибірка якісних даних та вибір інструментів для роботи. Навчання моделі буде за допомогою Python 3.9.5 та допоміжних бібліотек.

Опис бібліотек інтегрованих для навчання та побудови:

Numpy - це бібліотека Python для обробки масивів. Вона надає високопродуктивний багатовимірний масив об'єктів, а також інструменти для роботи з цими масивами. Numpy використовується в наукових обчисленнях, включаючи математичні, логічні, форми масиву, маніпуляції з масивами, сортування, вибірку, I/O, дискретне перетворення Фур'є та основні лінійні алгебраїчні операції.

Pandas - це бібліотека Python для аналізу та обробки даних, яка надає швидкі, гнучкі та виразні структури даних. Pandas призначений для роботи з "реальними" даними, які часто включають неповні, неоднорідні дані, а також даними, що змінюються у часі. Основні структури даних в pandas - це "DataFrame" (таблиця даних) та "Series" (одновимірний масив). З їх допомогою можна легко проводити такі операції, як сортування, фільтрація, групування, злиття, зміна форми, та інші види обробки та аналізу даних.

os - це стандартна бібліотека Python, яка надає функції для взаємодії з операційною системою. Вона дозволяє виконувати багато операцій з файловою системою, таких як читання з файлу, запис в файл, робота з шляхами до файлів,

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		20

створення та видалення директорій, отримання інформації про процес, оточення і т.д. Бібліотека os допомагає у випадках, коли вашому коду потрібно взаємодіяти з системними ресурсами або оточенням вашої програми.

tqdm - це бібліотека Python, що використовується для візуалізації прогресу циклів. Це допомагає стежити за тим, скільки часу може зайняти виконання великих циклів. tqdm є скороченням від "тахаддум" в арабському, що означає "прогрес". tqdm.notebook - це спеціальний підмодуль tqdm, оптимізований для використання в Jupyter Notebook. Він показує красиві інтерактивні віджети прогресу, які добре відображаються в блокнотах Jupyter.



```
from tqdm import tqdm
from time import sleep

fruits = tqdm(['apple', 'orange', 'grape'])
for fruit in fruits:
    sleep(0.3)
    fruits.set_description(f"Picking {fruit}")

"""
Picking apple: 33%|██████████          | 1/3
[00:05<00:10, 5.00s/it]
Picking orange: 67%|███████████      | 2/3
[00:10<00:05, 5.00s/it]
Picking grape: 100%|█████████████████| 3/3
[00:00<00:00, 3.27it/s]
"""
```

Рисунок 3.1 – Приклад візуалізації бібліотеки tqdm

torch: PyTorch є відкритою бібліотекою машинного навчання, що розробляється в основному командою Facebook's AI Research lab (FAIR). PyTorch є однією з найпопулярніших бібліотек для розробки та тренування нейронних мереж та інших моделей машинного навчання. Вона відрізняється гнучкістю та інтуїтивно

зрозумілим і динамічним програмуванням, що робить її особливо корисною для прототипування та дослідницьких проектів.

`torchvision`: ця бібліотека служить для роботи з даними для комп'ютерного зору в `PyTorch`. Вона включає популярні набори даних, архітектури моделей та звичайні зображення трансформації для комп'ютерного зору. Це допомагає спростити завантаження даних, передопрацювання і використання предтренованих моделей, що включають `ResNet`, `VGG`, `AlexNet` тощо.

`torchvision.transforms`: цей підмодуль `torchvision` містить класи, які використовуються для передопрацювання та перетворення даних під час завантаження. Передопрацювання даних є критично важливим кроком у будь-якому проекті машинного навчання, і `torchvision.transforms` надає різні методи для цього, включаючи перетворення тензорів, нормалізацію, обрізку, зміну розміру тощо.

`torch.nn`: Цей підмодуль `PyTorch` містить класи для побудови нейронних мереж. Він містить різні модулі та класи для створення власних архітектур нейронних мереж, включаючи класи для різних типів шарів (наприклад, лінійні, згорткові, рекурентні шари), також функції втрат, функції активації та інші утиліти для роботи з нейронними мережами.

`torch.nn.functional`: Цей підмодуль містить функції, які використовуються в процесі побудови та тренування нейронних мереж. Він містить багато важливих функцій, таких як функції активації (`ReLU`, `sigmoid`, `tanh`), функції втрат (`cross-entropy`, `MSE`), операції згортки та пулінгу, і таке інше. На відміну від `nn.Module`, які є класами, `nn.functional` містить стану-вільні (`stateless`) версії тих самих операцій.

`torch.optim`: Цей підмодуль містить реалізації різних алгоритмів оптимізації, які можна використовувати для тренування нейронних мереж. Він включає популярні оптимізатори, такі як `SGD`, `RMSprop`, `Adam` і так далі. Ці оптимізатори використовуються під час тренування для оновлення ваг моделі на основі обчислених градієнтів.

`torch.utils.data` це підмодуль PyTorch, який надає набір утиліт для завантаження та препроцесингу даних. Модулі які юдуть використані в роботі:

- `DataLoader`: цей клас використовується для створення ітерабельних об'єктів, які ефективно завантажують дані під час тренування моделі. Він дозволяє зручно керувати пакетами даних, автоматичним перемішуванням даних, паралельною обробкою даних, автоматичним завантаженням даних в пам'ять, і таке інше.

- `SubsetRandomSampler`: цей клас використовується для виборки випадкових підмножин даних для завантажувача даних. Це може бути корисним, коли вам потрібно розділити набір даних на тренувальний і тестовий набори.

- `TensorDataset`: цей клас дозволяє швидко створювати набір даних з тензорів PyTorch. Це корисно, коли у вас є дані, які вже знаходяться у форматі тензора, і вам потрібно створити набір даних для використання з `DataLoader`.

- `random_split`: ця функція використовується для розбиття набору даних на випадкові підмножини. Це корисно, коли вам потрібно розділити набір даних на тренувальний і тестовий набори.

- `ConcatDataset`: цей клас дозволяє об'єднати декілька наборів даних у один. Це може бути корисним, коли у вас є кілька наборів даних, які ви хочете об'єднати та використовувати разом.

Далі необхідно знайти багато якісних фото, які допоможуть натренувати модель з гарними показниками. Для цього я скористався відкритими датасетами на платформі для ML-ентузіастів «Kaggle».

Kaggle — платформа для змагань з аналітики та передбачувального моделювання, в рамках якого статистики та добувачі даних конкурують у створенні найкращих моделі для прогнозування та опису даних, запропонованих компаніями або користувачами. Цей краудсорсинговий підхід ґрунтується на тому, що є безліч стратегій, які можуть бути застосовані до будь-якого завдання з передбачувального моделювання, і наперед не відомо, яка методика або аналітичний підхід буде найбільш ефективним.

Competitions

Your Work

Q cancer

Filters

Results

Recently Launched







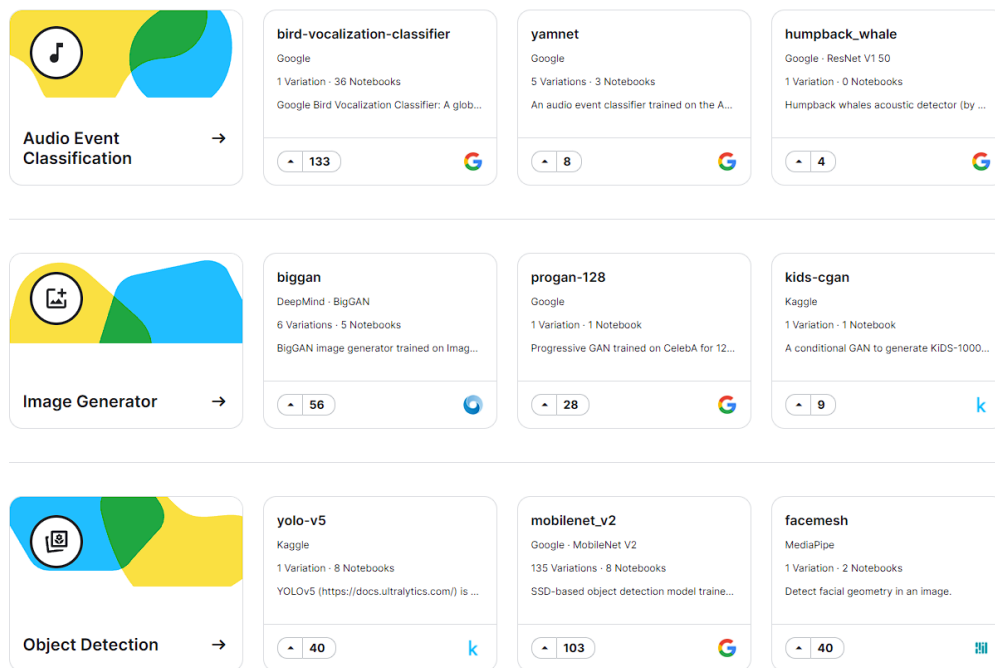
	RSNA Screening Mammography Breast Cancer Detection Find breast cancers in screening mammograms Featured - Code Competition - 1687 Teams - 3 months ago	\$50,000 ...
	UW-Madison GI Tract Image Segmentation Track healthy organs in medical scans to improve cancer treatment Research - Code Competition - 1548 Teams - a year ago	\$25,000 ...
	RSNA-MICCAI Brain Tumor Radiogenomic Classification Predict the status of a genetic biomarker important for brain cancer treatment Featured - Code Competition - 1555 Teams - 2 years ago	\$30,000 ...
	Prostate cANcer graDe Assessment (PANDA) Challenge Prostate cancer diagnosis using the Gleason grading system Featured - Code Competition - 1010 Teams - 3 years ago	\$25,000 ...
	Histopathologic Cancer Detection Identify metastatic tissue in histopathologic scans of lymph node sections Playground - 1149 Teams - 4 years ago	Knowledge ...
	Personalized Medicine: Redefining Cancer Treatment Predict the effect of Genetic Variants to enable Personalized Medicine Research - 1386 Teams - 6 years ago	\$15,000 ...

Рисунок 3.2 – Змагання на платформі Kaggle з темою дослідження онкології



Audio Event Classification	bird-vocalization-classifier Google 1 Variation - 36 Notebooks Google Bird Vocalization Classifier: A glob...	yamnet Google 5 Variations - 3 Notebooks An audio event classifier trained on the A...	humpback_whale Google - ResNet V1 50 1 Variation - 0 Notebooks Humpback whales acoustic detector (by ...
Image Generator	biggan DeepMind - BigGAN 6 Variations - 5 Notebooks BigGAN image generator trained on Imag...	progan-128 Google 1 Variation - 1 Notebook Progressive GAN trained on CeleBA for 12...	kids-cgan Kaggle 1 Variation - 1 Notebook A conditional GAN to generate KIDS-1000...
Object Detection	yolo-v5 Kaggle 1 Variation - 8 Notebooks YOLOv5 (https://docs.ultralytics.com/) is ...	mobilenet_v2 Google - MobileNet V2 135 Variations - 8 Notebooks SSD-based object detection model traine...	facemesh MediaPipe 1 Variation - 2 Notebooks Detect facial geometry in an image.

Рисунок 3.3 – Категорії розроблених моделей та їх приклади

Тому для інтеграції датасету потрібно виконати наступні кроки:

Встановлення трансформацій даних:

transform - це послідовність операцій, які будуть виконані над зображеннями.

Перший крок - зміна розміру зображення до розміру 227x227 пікселів. Другий крок перетворює зображення у тензор. Третій крок нормалізує тензор зображення,

Змін.	Арк.	№ докум.	Підп.	Дата.
-------	------	----------	-------	-------

IA94.170БАК.004 ПЗ

Арк.

24

використовуючи передбачені середні значення та стандартні відхилення для кожного з трьох каналів зображення (RGB).

Встановлення розміру пакета (batch size):

batch_size - це кількість зображень, які будуть проходити через нейронну мережу за одну ітерацію.

Завантаження навчальних і тестових даних:

Використовується ImageFolder з torchvision.datasets, що дозволяє просто завантажувати датасети зображень з файлової системи. Тут навчальний і тестовий датасети завантажуються з вказаних шляхів, трансформуються за допомогою визначеного раніше transform.

Створення навчального завантажувач даних:

DataLoader використовується для створення навчального загрузчика, який впорядковує навчальні дані, розподіляє їх на пакети з заданим розміром, допомагає впорядковувати дані випадковим чином на кожній ітерації та використовує два потоки робочих для завантаження даних.

Визначення класів:

Отримується список класів з навчального набору даних.

Ми імплементували трансформації для роботи з нашими зображеннями, включаючи зміну розміру, конвертацію в тензори та нормалізацію.

Далі ми встановили розмір пакету, що відповідає кількості зразків, які будуть передаватися в модель за одну ітерацію. Завантаження даних було розділено на два датасети - навчальний і тестовий.

Організація даних була зроблена з використанням функції ImageFolder, яка спрощує роботу з даними в структурі папок. Для подальшої оптимізації процесу навчання ми використали DataLoader, що значно полегшує ітерацію по даних під час тренування.

В результаті, ми успішно підготували наші дані для навчання моделі, враховуючи всі необхідні параметри та вимоги до даних, що є важливим кроком на шляху до побудови ефективної системи для виявлення раку шкіри.

Далі обираємо модель для навчання моделі. Я зупинився на AlexNet.

					ІА94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		25

AlexNet — згортова нейронна мережа, що застосовується для розпізнавання зображень. Розроблена українцем Алексом Крижевським у співпраці з Іллею Суцкевером та Джефрі Гінтоном, який був радником Крижевського.

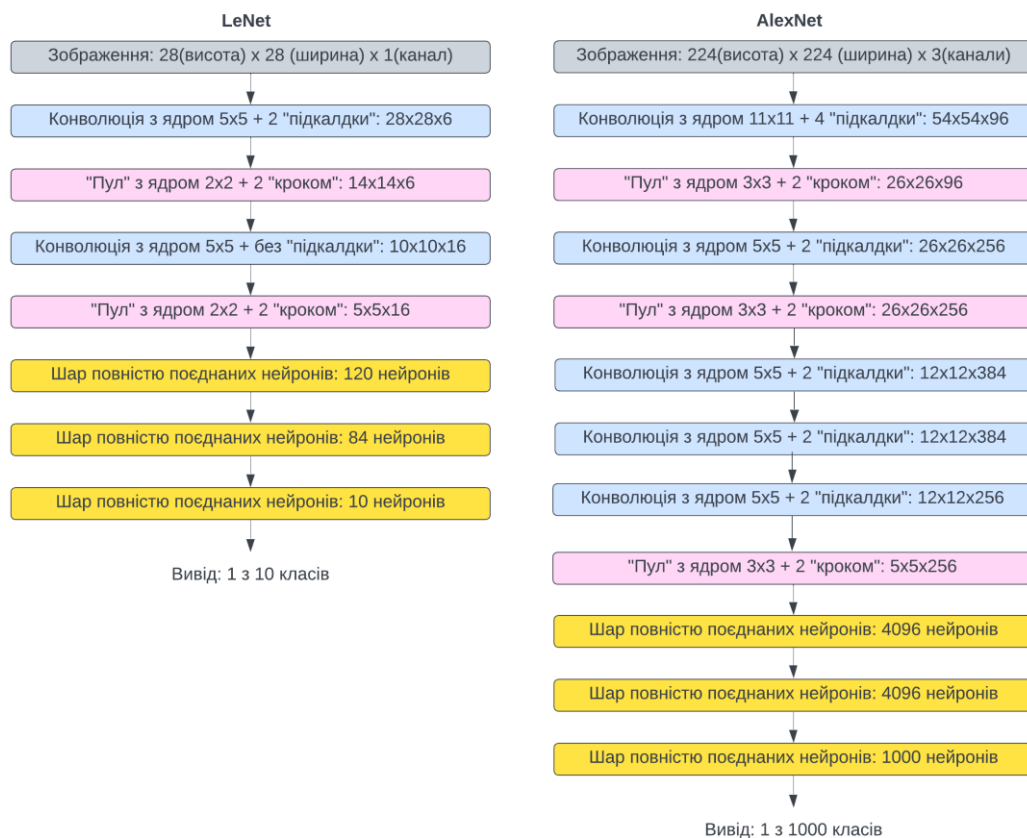


Рисунок 3.4 – Порівняння моделей LeNet та AlexNet пошарово

У бібліотеці PyTorch є готова модель AlexNet, але для свого проекту я хотів показати структуру та натренувати її самостійно.

Основні елементи моделі:

features: Ця послідовність з'єднує шари конволюції, активації (ReLU), пулінгу (MaxPool2d) та адаптивного пулінгу (AdaptiveAvgPool2d). Ці шари відповідають за витягнення основних ознак зображення.

classifier: Ця частина мережі складається з повнозв'язних шарів (Linear), шарів відкидання (Dropout) та останнього повнозв'язного шару для класифікації. Вона використовує ознаки, витягнуті в частині features, для передбачення класу зображення.

forward: Ця функція описує, як дані проходять через модель. Вона передає вхід через features і classifier для отримання передбачення.

Основна відмінність цієї моделі від вбудованої моделі AlexNet в PyTorch полягає в налаштуванні шарів і виходів для конкретної задачі. В оригінальній моделі AlexNet вихідний шар налаштований для класифікації 1000 класів, тоді як у цій моделі вихідний шар налаштований на два класи, що відповідає задачі бінарної класифікації для детектування меланоми.

Також в цій моделі використовуються шари Dropout, що додають регуляризацію до моделі та допомагають запобігти перенавчанню, що не є стандартною особливістю оригінального AlexNet.

Загалом, ця модифікація AlexNet адаптує класичну архітектуру для конкретної проблеми, що підкреслює гнучкість і адаптивність глибокого навчання.

Таблиця 3.1 – Технічні характеристики моделі AlexNet

Структура	Кількість параметрів	Розмір вихідного тензору
Conv2d	96	55x55x96
ReLU	-	55x55x96
MaxPool2d	-	27x27x96
Conv2d	256	27x27x256
ReLU	-	27x27x256
MaxPool2d	-	13x13x256
Conv2d	512	13x13x512
ReLU	-	13x13x512
MaxPool2d	-	6x6x512
Conv2d	512	6x6x512
ReLU	-	6x6x512
Conv2d	256	6x6x256
ReLU	-	6x6x256
MaxPool2d	-	3x3x256
AdaptiveAvgPool2d	-	6x6x256
Linear	-	4096
Dropout	-	4096
Linear	-	4096
Dropout	-	4096
Linear	-	2

Далі необхідно зробити обчислення loss функції (втратна функція).

Для цього було використано `CrossEntropyLoss()`

`nn.CrossEntropyLoss()` це втратна функція (або цільова функція) в PyTorch, яка часто використовується для задач класифікації. Конкретно, вона призначена для багатокласової класифікації.

Сама `nn.CrossEntropyLoss()` вимірює втрату перехресної ентропії між вихідними даними та цільовими мітками. Це включає в себе два основних кроки:

Логарифмічний софтмакс. Вихідні дані перетворюються на ймовірності за допомогою функції `softmax`, а потім застосовується логарифм. Функція `softmax` перетворює вектор вихідних даних на вектор ймовірностей, який сумується до одиниці, а логарифм використовується для шкалювання цих ймовірностей.

Негативна логарифмічна правдоподібність. Втрата перехресної ентропії вимірює різницю між цими логарифмічними ймовірностями та цільовими мітками.

Основна мета використання `nn.CrossEntropyLoss()` полягає в тому, щоб мінімізувати різницю між передбаченими ймовірностями моделі та дійсними мітками. Коли модель правильно передбачає мітку, втрата перехресної ентропії нижча. Навпаки, коли модель неправильно передбачає мітку, втрата перехресної ентропії вища.

Ця втратна функція є досить поширеною для задач класифікації, тому що вона добре працює в задачах, де є взаємовиключні класи, і має вбудовані механізми для шкалювання ймовірностей й покарання моделей, які зробили невірні передбачення.

Після підготовки основних елементів можна переходити до підготовки навчання моделі, а саме кросс-валідації.

Кросс-валідація (`cross-validation`) є методом оцінки точності моделі машинного навчання та використовується для визначення здатності моделі до узагальнення на нових, раніше невиданих даних.

Основна ідея кросс-валідації полягає у тому, що замість того, щоб розділяти доступні дані на окремий тренувальний та тестовий набори, дані розділяються на декілька підвибірок (фолдів). Модель тренується на одній підвибірці і оцінюється на іншій. Цей процес повторюється декілька разів з різними комбінаціями

тренувальних та тестових підвбірок. За результатами оцінок точності на різних підвбірках обчислюється середнє значення точності, що дозволяє отримати оцінку загальної точності моделі.

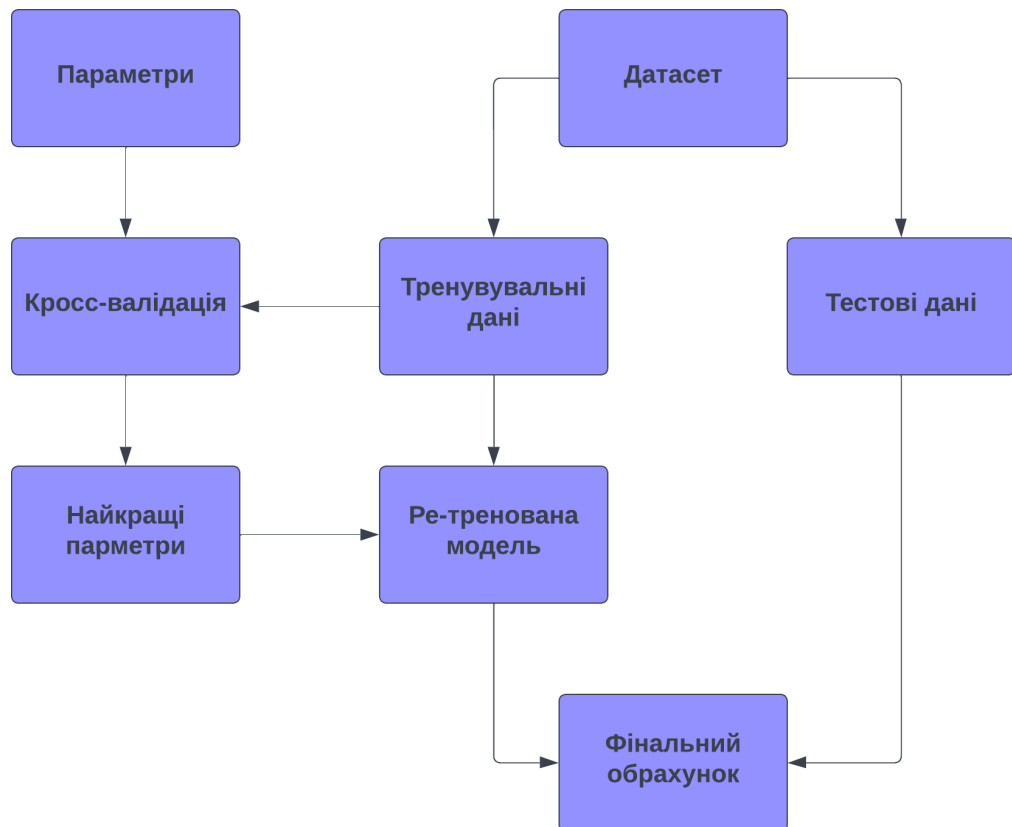


Рисунок 3.2. – Блок-схема кросс-валідації для загального прикладу

Основні переваги кросс-валідації:

Оцінка точності моделі: Кросс-валідація дозволяє отримати оцінку точності моделі, яка враховує його здатність до узагальнення на нових наборах даних. Це допомагає уникнути перенавчання (overfitting) або недонавчання (underfitting) моделі.

Ефективне використання даних: Кросс-валідація використовує всі доступні дані для тренування та тестування моделі. Це особливо важливо в ситуаціях, коли ви маєте обмежену кількість даних.

Виявлення переносимості: Кросс-валідація дозволяє оцінити, наскільки добре модель може переносити свою точність на різні набори даних. Це особливо

корисно, коли ви плануєте застосовувати модель на нових, раніше невиданих даних.

Вибір кращої моделі: Кросс-валідація може використовуватись для порівняння різних моделей та визначення найкращої з них. Це допомагає вибрати модель з найвищою точністю для вашої конкретної задачі.

Для позбавлення від перенавчання моделі я вирішив обрати метод кросс-валідації, а саме K-Fold Cross Validation.

K-Fold Cross Validation (крос-валідація з k-розбиттями) є одним з методів оцінки ефективності моделей машинного навчання. Він дозволяє оцінити якість моделі та зробити узагальнену оцінку її точності, уникнувши проблем підвищеної довірчості або перенавчання.

Процес K-Fold Cross Validation включає наступні кроки:

Розділіть дані на k підвибірок однакового розміру. Це може бути здійснено випадковим чином або зберегти баланс класів у кожній підвибірці, якщо ви працюєте з задачею класифікації.

Для кожної з k ітерацій, використовуйте одну підвибірку як тестовий набір даних, а інші k-1 підвибірки - як тренувальні дані.

Для своєї моделі я обрав 30 епох, та 10 фолдів(кількості складок у кросс-валідації K-складок (KFold). Це означає, що набір даних буде розділено на 10 рівних частин або "складок".).

Крім стандартного K-Fold Cross Validation, існують його варіації, такі як Stratified K-Fold (збереження балансу класів), Leave-One-Out Cross Validation (LOOCV, коли використовується лише одна спостереження як тестовий набір), і інші. для яких задач краще використовувати ці види кросс-валідації.

Stratified K-Fold Cross Validation: Цей підхід забезпечує збереження балансу класів у кожній з кросс-валідаційних підвбірок. Він часто використовується в задачах класифікації, де даним є незбалансовані класи, щоб забезпечити адекватну оцінку точності для кожного класу.

Leave-One-Out Cross Validation (LOOCV): У цій варіації кожна точка даних використовується як окремий тестовий набір, а всі інші точки - як тренувальний

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		30

набір. LOOCV корисний, коли набір даних дуже малий, і ми хочемо максимально використовувати наявні дані для оцінки моделі. Однак цей підхід може бути вимогливим за обчислювальними ресурсами.

Time Series Cross Validation: Цей підхід призначений для обробки часових рядів, де послідовність даних має важливе значення. Він враховує хронологічний порядок даних та забезпечує правильну оцінку моделі на майбутніх періодах шляхом використання "скользящего вікна" для тренування та тестування моделі.

Вибір конкретного виду крос-валідації залежить від характеристик даних, розміру вибірки, проблеми, яку ви вирішуєте та доступних обчислювальних ресурсів

Але було обрано K-Fold Cross Validation, бо:

- 1) Розмір вибірки був достатній (не малий та не занадто великий)
- 2) Тип даних, як фото, показує високі та стабільні результати точності
- 3) Це стандартний варіант який працює з багатьма типами даних, тому бажання подальшого масштабування на інші види онкологічних утворень, допоможе отримувати високу точність без додавання додаткових типів валідації даних.

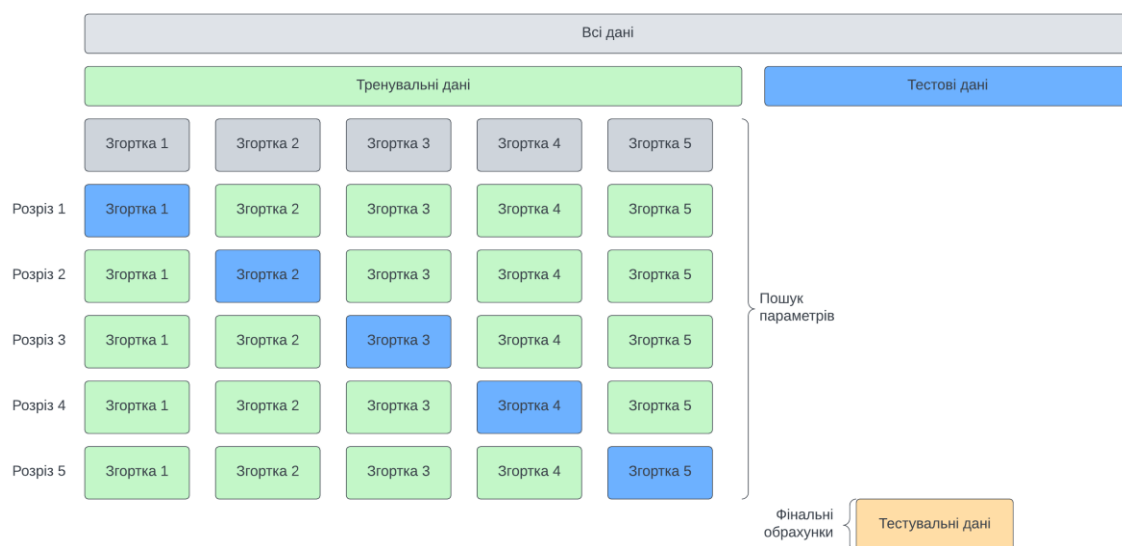


Рисунок 3.3. – Діаграма K-Fold кросс-валідації

Опис функцій `train_epoch()` та `valid_epoch()`

Ці функції відповідають за тренування моделі протягом одної епохи та перевірку її на валідаційному наборі даних.

`train_epoch` - ця функція відповідає за тренування моделі. Її параметри включають:

`model` - саму модель, яку треба навчити.

`device` - об'єкт `PyTorch`, що визначає, чи будуть обчислення виконуватись на CPU або GPU.

`dataloader` - об'єкт `DataLoader`, що надає дані для тренування.

`loss_fn` - функція втрат, яка використовується для оцінки точності прогнозів моделі.

`optimizer` - оптимізатор, який використовується для оновлення ваг моделі.

`valid_epoch` - ця функція відповідає за перевірку моделі на валідаційному наборі даних. Її параметри подібні до `train_epoch`, але без оптимізатора, оскільки в процесі валідації ваги моделі не оновлюються.

Функції працюють за схожим принципом. Вони проходять через всі дані в наборі (для тренування або валідації), роблять прогнози за допомогою моделі, розраховують втрати і оновлюють ваги моделі (тільки у процесі тренування). Також вони ведуть облік загальної втрати та кількості правильних прогнозів для підрахунку точності.

В кінці кожної епохи функції повертають загальні втрати та кількість правильних прогнозів.

Як висновок, цей фрагмент коду визначає основні функції для тренування та валідації моделі. Це важлива частина будь-якого процесу машинного навчання, оскільки вона відповідає за оновлення ваг моделі на основі даних і їх міток, а також за перевірку моделі на нових, невідомих їй даних.

Після налаштування всіх попередніх етапів, ми переходимо до ключового – тренування самої моделі. Така довга підготовка необхідна аби отримати якісну та багатофункціональну модель.

Нижче буде покроковий опис тренування з використанням усіх етапів описаних вище:

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		32

- 1) Кожний цикл в цьому циклі відповідає одному "fold" (або розділу) у K-кратній перехресній перевірці.
- 2) Індеси для тренувального та валідаційного наборів для кожного розділу генеруються за допомогою `splits.split(np.arange(len(dataset)))`.
- 3) Створюється випадковий вибіркового піднабір для тренувального та тестового наборів за допомогою індесів, отриманих з `KFold.split`.
- 4) За допомогою отриманих випадкових вибіркового піднаборів створюються тренувальний та тестовий загрузчики даних.
- 5) Оптимізатор створюється для оновлення параметрів моделі. В цьому випадку використовується стохастичний градієнтний спуск (SGD).
- 6) Запускається цикл для кожної епохи тренування.
- 7) Виконується тренування моделі за допомогою тренувального набору даних.
- 8) Проводиться валідація моделі за допомогою тестового набору даних.
- 9) Втрати та точність обчислюються для тренувального та тестового наборів даних.
- 10) Після кожної епохи виводиться статистика навчання, включаючи середні втрати навчання та тестування, а також точності.
- 11) Збереження моделі

Оцінка натренованої моделі на реальних даних за допомогою Predict Test Set Predict Test Set означає застосування навченої моделі машинного навчання до тестового набору даних для отримання прогнозів або передбачень. Тестовий набір даних є окремим набором даних, який не використовувався під час тренування моделі, і використовується для оцінки точності та ефективності моделі на реальних даних.

Головна мета Predict Test Set полягає в тому, щоб оцінити, наскільки добре модель може узагальнювати свої прогнози на нові, раніше невидані дані. Порівнюючи прогнози моделі з фактичними значеннями у тестовому наборі даних, можна оцінити точність, чутливість, специфічність та інші метри моделі.

Використання тестового набору даних допомагає уникнути перенавчання (overfitting) моделі, тобто ситуації, коли модель дуже добре пристосовується до навчальних даних, але не вміє здатися з новими даними. Через те, що тестовий набір даних не використовується під час тренування моделі, він надає об'єктивну оцінку ефективності моделі на реальних ситуаціях.

Predict Test Set також дозволяє порівняти ефективність різних моделей машинного навчання та вибрати найкращу модель для конкретної задачі. За допомогою метрик, які визначаються на основі прогнозів і фактичних значень в тестовому наборі даних, можна порівнювати та ранжувати моделі за їхньою точністю та ефективністю.

Узагальнюючи, Predict Test Set є важливим етапом в оцінці та перевірці моделі машинного навчання на реальних даних, що допомагає визначити її здатність до узагальнення та надійність в реальних умовах.

Визначається кількість класів (`nb_classes = 2`). У цьому випадку у нас є два класи (злаякісний меланома або здорова шкіра).

Завантажується тестовий набір даних за допомогою `DataLoader` з `PyTorch`. Це об'єкт, який генерує пакети тестових даних.

Створюється нульова матриця, що відповідає розміру матриці помилок (кількість класів x кількість класів).

Модель переводиться в режим оцінки (`evaluation mode`), викликаючи `model.eval()`. Це змінює поведінку деяких слоїв моделі, наприклад, слоїв "`dropout`" та "`batch normalization`".

В циклі `for`, що ітерує по тестових даних, робиться наступне:

- 1) Тестові вхідні дані та мітки переносяться на CPU.
- 2) Вихідні дані моделі визначаються за допомогою `model(test_inputs)`.
- 3) Виконується класифікація, вибираючи клас з найвищим вихідним значенням.
- 4) Оновлюється загальна кількість міток та кількість правильних передбачень.

5) Заповнюється матриця помилок: для кожного пари реальної мітки та передбаченої мітки збільшується відповідний елемент матриці.

б) Після того, як всі тестові дані були оброблені, виводиться точність тестування, розрахована як відсоток правильних передбачень від загальної кількості.

Нарешті, виводиться матриця помилок, яка показує, як модель класифікувала різні класи.

Висновок: Цей блок коду оцінює модель на тестовому наборі даних та допомагає визначити, наскільки добре модель впоралася з класифікацією обох класів. Матриця помилок - це корисний інструмент для визначення, які помилки модель схильна робити.

Цей проект представляє собою цілісну систему для тренування та валідації моделі глибокого навчання для класифікації зображень меланоми. Використовується модифікована модель AlexNet для розпізнавання меланоми на зображеннях.

В першу чергу, було імпортовано всі необхідні бібліотеки, такі як PyTorch, torchvision, numpy і pandas. Далі було проведено передобробку даних за допомогою torchvision.transforms, що допомагає виконувати аугментації та перетворення зображень.

Модель була змінена відносно оригінального AlexNet: після послідовності згорткових та пулінгових шарів, було додано три повнозв'язні шари з функцією активації ReLU та Dropout для зменшення перенавчання.

Для тренування моделі було використано крос-ентропійну втрату (Cross-Entropy Loss), яка є стандартною для задач класифікації. Оптимізатор - стохастичний градієнтний спуск (SGD) з розгоном (momentum).

Код також включає в себе використання K-блокової перехресної перевірки (K-Fold Cross Validation) для надійної оцінки моделі. Це важливо, оскільки дає нам змогу оцінити, як модель виступить на невідомих їй даних.

Після навчання моделі було проведено тестування та використано матрицю помилок для детального аналізу результатів класифікації.

					ІА94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		35

Загалом, цей проект є глибоким і досить детальним вивченням процесу побудови, навчання та оцінки моделі глибокого навчання для задачі класифікації зображень. Використання PyTorch та інших бібліотек мови Python демонструє, як засоби машинного навчання можуть бути застосовані для розв'язання конкретних задач.

3.3 Front-end частина

Реалізація Front-end частини буде за допомогою вбудованої системи бібліотеки Streamlit, яка не потребує розгортання класичних фронтенд інструментів, такі як HTML + CSS + JS, а використовує localhost та мову програмування Python.

Опис бібліотек інтегрованих для навчання та побудови:

Streamlit: Streamlit - це бібліотека для створення інтерактивних веб-застосунків з використанням Python. Вона забезпечує простий у використанні набір функцій, що дозволяє додавати інтерактивність, а також виконувати обробку та візуалізацію даних у веб-додатках.

Pillow (PIL - Python Imaging Library): Pillow - це бібліотека для обробки зображень у Python. Вона підтримує широкий спектр форматів зображень та надає набір інструментів для роботи з зображеннями, включаючи зміну розмірів, обрізку, перетворення та інше. Image - це основний клас, що представляє зображення у Pillow.

Опис процесу:

Після імпорту бібліотек потрібно завантажити модель:

- 1) Створення новий екземпляр моделі AlexNet, використовуючи клас AlexNet, який був визначений раніше.
- 2) Завантажує навчені параметри моделі з файлу melanoma_state_CNN.pt.
load_state_dict() - це метод PyTorch для завантаження параметрів моделі.
torch.load() - це функція для завантаження об'єктів PyTorch з файлу.

`map_location=device` дозволяє перенести вантаження моделі на відповідний апаратний пристрій (CPU або GPU) в моєму випадку CPU.

- 3) Переміщує модель на відповідний апаратний пристрій (CPU або GPU), визначений змінною `device`.
- 4) Змінює режим моделі на режим оцінювання (`evaluation mode`). Це важливо для моделей, що використовують техніки, які ведуть себе по-різному під час навчання і оцінки, наприклад `Dropout` і `BatchNormalization`.
- 5) Повертається завантаженої модель.
- 6) Отже, з усіх цих кроків у цій функції створюється і повертається навчена модель `AlexNet()`, яка готова до використання для передбачення.

Препроцесінг переданих файлів:

- 1) Створює послідовність трансформацій за допомогою `transforms.Compose`. Цей компонент `PyTorch` використовується для створення послідовності трансформацій, які будуть виконані на зображенні.
- 2) Зміна розміру вхідного зображення до зазначених розмірів (256, 256). Це важливо для подальшого використання моделі, яка очікує зображення певного розміру.
- 3) Обрізання зображення з центру до 224x224 пікселів.
- 4) Конвертація `PIL Image` або `NumPy ndarray` в `torch.*Tensor`.
- 5) Нормалізування тензорів зображень з вказаними середніми значеннями та стандартними відхиленнями по каналах.
- 6) Виконання попередньо визначені трансформації на вхідному зображенні.
- 7) Додавання додаткового розміру для відповідності формату вхідних даних моделі `PyTorch` і переміщує зображення на відповідний апаратний пристрій (CPU або GPU).

В процесі обробки зображень для глибокого навчання, функція, яка відповідає за перетворення зображень в придатний для моделі формат, має важливе значення. Це починається зі зчитування зображення з диска або з іншого джерела.

Вхідні зображення також можуть мати різні розміри, тому їх потрібно привести до єдиного розміру. Це робиться за допомогою операцій зміни розміру або обрізки зображень.

На останньому етапі зображення перетворюється в тензори. Тензори - це основний тип даних, які використовуються в бібліотеках глибокого навчання, таких як PyTorch. Вони схожі на масиви у NumPy, але, на відміну від них, тензори можуть використовуватися на графічних процесорах, що дозволяє значно прискорити обчислення.

Таким чином, перетворення вхідних зображень до формату, який придатний для моделі PyTorch, включає в себе декілька кроків і є важливим етапом у процесі підготовки даних для глибокого навчання.

Завантаження зображення через веб-інтерфейс Streamlit:

- 1) Streamlit надає вбудовану функцію `file_uploader`, яка дозволяє користувачам завантажувати файли через веб-інтерфейс. У цьому випадку, функція використовується для завантаження зображення.
- 2) Перевіряє, чи було завантажено файл. Якщо файл не завантажено, функція повертає `None`.
- 3) Завантажені бінарні дані перетворюються на об'єкт зображення за допомогою модулів `PIL Image` та `io.BytesIO`.
- 4) Завантажене зображення відображається на веб-сторінці Streamlit з підписом 'Uploaded Image.'
- 5) Завантажене зображення повертається як результат функції.

Також викликається функція `load_model()`, яку ви раніше визначили, щоб завантажити натреновану модель AlexNet.

Створення веб-додатку:

- Встановлення заголовків веб-сторінки на 'Melanoma Prediction App' за допомогою вбудованої функції `title` в Streamlit.
- Виклик функції `load_image()`, яку ми раніше визначили, щоб завантажити зображення, яке було надіслано користувачем через веб-інтерфейс.

- Перевірка, чи було завантажено зображення. Якщо зображення було завантажено, воно обробляється та використовується для передбачення.

- Виклик функції `preprocess_image()`, яку ми раніше визначили, щоб попередньо обробити завантажене зображення перед введенням його в модель.

- Виконується передбачення за допомогою моделі на попередньо обробленому зображенні без обчислення градієнтів, оскільки ми не збираємося оновлювати ваги моделі.

- Вирахування ймовірності для кожного класу (здорова шкіра та меланома) за допомогою функції `softmax`.

- Вивід відсотків ймовірності для здорової шкіри та меланоми на веб-сторінці.

Створено веб-додаток для передбачення меланоми, використовуючи Streamlit.

Streamlit є важливим інструментом в проєкті, оскільки він дозволяє створювати прості та інтерактивні веб-інтерфейси. Це бібліотека Python, орієнтована на наукові дослідження та аналіз даних. Використання Streamlit надає можливість розробити веб-додаток із мінімальними зусиллями з боку програміста, що значно прискорює процес розробки.

Додаток, створений з використанням Streamlit, представляє собою інтуїтивно зрозумілий веб-інтерфейс, що дозволяє користувачам завантажувати зображення для аналізу. Цей інтерфейс простий у використанні, а його простота сприяє взаємодії користувача з системою, роблячи процес завантаження зображень швидким та зручним.

Після завантаження зображення, воно передається в натреновану модель глибокого навчання. В даному випадку, ми використовуємо варіацію моделі AlexNet, однієї з найвідоміших моделей у сфері глибокого навчання та комп'ютерного зору. Ця модель аналізує зображення та повертає ймовірність наявності ознак меланоми, надаючи користувачу цінну інформацію, що може допомогти в ранньому виявленні цього стану.

Проект демонструє силу машинного навчання в реальному світі, особливо в контексті медичної діагностики. Цей додаток - це більше, ніж просто теоретичний приклад застосування алгоритмів глибокого навчання, це реальний інструмент, що може допомогти людям виявити ризик меланоми на ранніх стадіях.

Весь цей процес підтверджує, наскільки важливо надавати доступ до таких інструментів широкій аудиторії. Простота веб-інтерфейсу, розробленого за допомогою Streamlit, дозволяє людям без технічного фону використовувати складні алгоритми глибокого навчання для аналізу своїх власних зображень, що може виявитися життєво важливим у виявленні меланоми.а фокусуватися на основній області інтересу (шкіра) і мінімізувати вплив несуттєвого фону.

3.4 Опис контрольного прикладу

Після запуску файлу з фронтенд частиною на основі Streamlit за допомогою команди

```
streamlit run D:\Eugene\kpi\diploma\diploma\front_visualiazation.py
```

Після цього відкривається веб-застосунок розташований на локалхості мого ПК.

```
Warning: to view this Streamlit app on a browser, run it with the following
command:

streamlit run D:\Eugene\kpi\diploma\diploma\front_visualiazation.py [ARGUMENTS]
D:\Eugene\kpi\diploma> streamlit run D:\Eugene\kpi\diploma\diploma\front_visualiazation.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.0.110:8501
```

Рисунок 3.4 – Скріншот з терміналу з виводом параметрів локалхосту

Melanoma Prediction App

Upload an image for prediction

Drag and drop file here
Limit 200MB per file

Browse files

Рисунок 3.5 – Стартовий стан веб-застосунку

У веб-додатках, в тому числі і в додатках, які створені за допомогою Streamlit, однією з поширених функцій є завантаження файлів з локальної системи користувача. Це може бути реалізовано за допомогою використання HTML елемента `<input type="file">`, який відкриває вікно вибору файлу на клієнтському комп'ютері, коли користувач натискає на кнопку «Browse files».

Цей діалогове вікно відображає файлоу систему на локальному комп'ютері користувача, включаючи всі директорії, файли і диски, які доступні для користувача. Це дозволяє користувачеві навігуватися по своїй файлової системі та вибирати файл, який він хоче завантажити в додаток.

Для більшої зручності, часто надається можливість додавання файлів через перетягування - метод «Drag and drop». У цьому випадку, користувач може взяти файл зі своєї файлової системи, перетягнути його на вікно веб-браузера, і цей файл буде автоматично завантажений в додаток. Ця функція забезпечує додаткову зручність для користувачів, особливо коли їм потрібно завантажити велику кількість файлів.

Отже, як кнопка «Browse files», так і метод «Drag and drop» надають користувачам гнучкі та зручні способи завантаження файлів з їхньої локальної системи в веб-додаток.

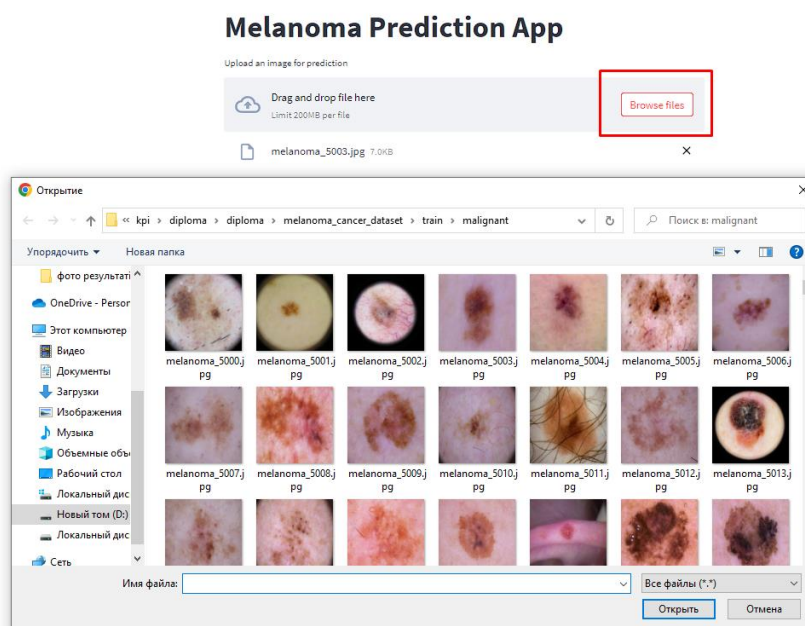


Рисунок 3.6 – Інтерфейс при виборі файлу

Користувачі повинні знати, що при завантаженні файлів у веб-додаток кожне вибране зображення проходить початкову обробку. Цей процес включає перегляд зображення і його відображення в межах, які були визначені програмним кодом. Ця функція є дуже корисною, оскільки вона дозволяє користувачам адаптувати розмір відображення зображення у веб-додатку, враховуючи їхні власні потреби та обмеження інтерфейсу. Зокрема, можливе масштабування зображення до певного розміру, встановлення максимального розміру зображення, або навіть зміна орієнтації зображення для оптимального відображення.

Щодо вибору фотографій для аналізу, то важливо зазначити, що обмеження практично відсутні. Зображення можуть бути отримані з різних джерел: вони можуть бути взяті з датасету, що використовувався для тренування моделі, або зроблені за допомогою камери смартфона або будь-якого іншого пристрою. Однак, важливо дотримуватися декількох ключових критеріїв при виборі зображення для аналізу. Наприклад, область шкіри, яка аналізується, має бути рівномірно освітленою, гамма кольорів на фотографії має бути збережена, а також область шкіри повинна займати більше 70% від усього зображення. Ці вимоги допомагають забезпечити максимальну точність та надійність результатів аналізу зображень за допомогою моделі глибокого навчання, що на важливо для ефективної та своєчасної діагностики.

Щодо тестування, додаток не має жодних обмежень на типи файлів або джерела зображень. Це означає, що можна завантажити зображення як з датасету для тренування, так і зроблені на камеру телефону. Головне, щоб зображення було високої якості і відповідало декільком важливим вимогам:

1. Рівномірне освітлення ділянки шкіри: Це важливо для забезпечення точного визначення характеристик шкіри, таких як колір, текстура, і форма. Нерівномірне освітлення може спотворити ці характеристики і призвести до неточних передбачень.
2. Зберігання гамми на фото: Коли зображення відправляється через веб, часто застосовуються алгоритми стиснення, що можуть знизити якість зображення, зокрема, змінити його гаму. Для найкращих результатів

необхідно, щоб гамма зображення була збережена в максимально можливій мірі.

3. Ділянка з шкірою повинна займати 70%+ усього зображення: Це необхідно, щоб модель могла фокусуватися на основній області інтересу (шкіра) і мінімізувати вплив несуттєвого фону.

Melanoma Prediction App

Upload an image for prediction



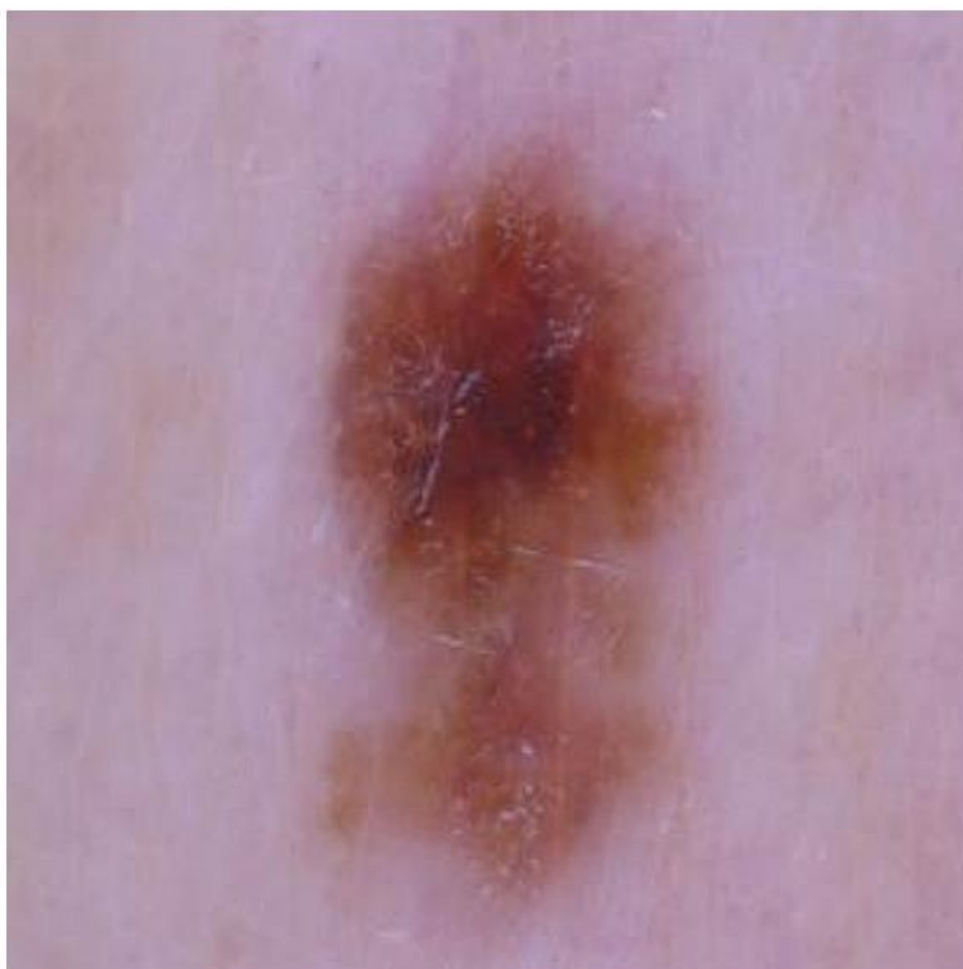
Drag and drop file here

Limit 200MB per file

Browse files



melanoma_5003.jpg 7.0KB



Uploaded Image.

Probability of healthy skin: 0.01%

Probability of melanoma cancer: 99.99%

Рисунок 3.7 – Фінальний вигляд після обробки зображення

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		43

Після завантаження об'єкту одразу відбувається обробка фото за допомогою натренованої моделі. Під фото виводиться аналітична інформація про вірогідність меланоми в такому форматі:

Probability of healthy skin: 0.01%

Probability of melanoma cancer: 99.99%


Рисунок 3.8 – Вірогідності після аналізу фото

Користувачі повинні знати, що при завантаженні файлів у веб-додаток кожне вибране зображення проходить початкову обробку. Цей процес включає перегляд зображення і його відображення в межах, які були визначені програмним кодом. Ця функція є дуже корисною, оскільки вона дозволяє користувачам адаптувати розмір відображення зображення у веб-додатку, враховуючи їхні власні потреби та обмеження інтерфейсу. Зокрема, можливе масштабування зображення до певного розміру, встановлення максимального розміру зображення, або навіть зміна орієнтації зображення для оптимального відображення.

Після проходження процесу аналізу, додаток виводить результати у вигляді ймовірностей для кожного з потенційних станів шкіри. Ці результати представлені у вигляді відсотків, які відображають вірогідність того, що шкіра на зображенні здорова або має ознаки меланоми. Наприклад, додаток може вивести результати в такому форматі: "Вірогідність здорової шкіри : xx.xx%, Вірогідність раку меланоми : xx.xx%". Ці відсотки вираховуються на основі аналізу зображення моделлю глибокого навчання.

Після відображення результатів, додаток також надає декілька прикладів зображень для ілюстрації різних можливих результатів. Перший приклад показує зображення здорової шкіри. Наступний приклад демонструє зображення, яке модель не змогла класифікувати з повною впевненістю. Це може відбутися, коли модель виявляє ознаки обох станів на зображенні або коли якість зображення не дозволяє моделі чітко ідентифікувати ознаки стану шкіри.

Під кожним з цих зображень відображається відсоток, що вказує на результат аналізу моделлю. Це дозволяє користувачам бачити, як система інтерпретує та аналізує різні типи зображень, а також розуміти, наскільки впевненою є модель у своїх передбаченнях для кожного конкретного випадку.

 melanoma_10108.jpg 9.3KB

X



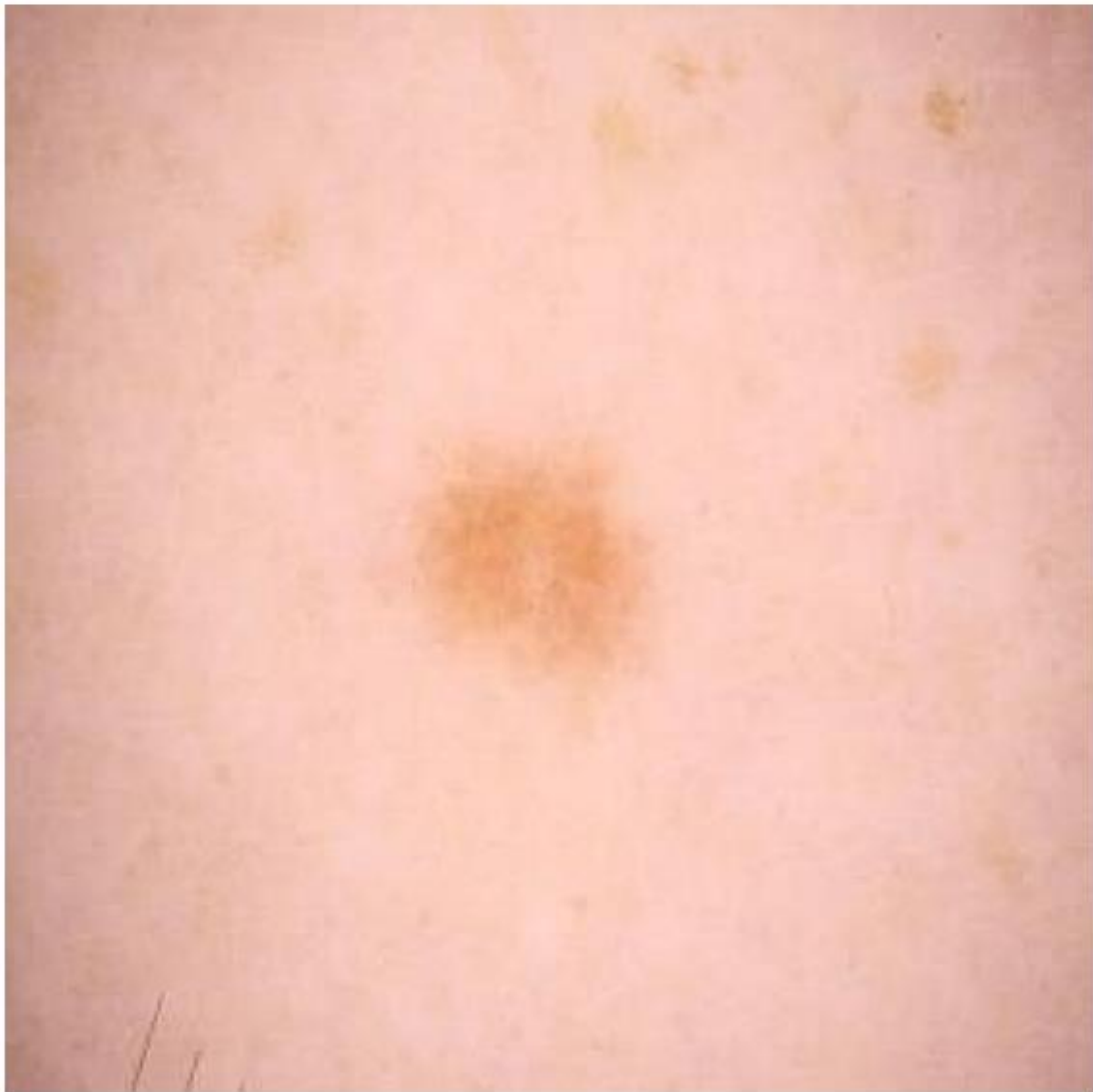
Uploaded Image.

Probability of healthy skin: 44.19%

Probability of melanoma cancer: 55.81%

Рисунок 3.9.1 – Не класифікована

melanoma_9617.jpg 6.0KB



Uploaded Image.

Probability of healthy skin: 100.00%

Probability of melanoma cancer: 0.00%

Рисунок 3.9.2 – Здорова

Останній приклад включає зображення, зняте на телефон. Це важливо, тому що демонструє можливість використання реальних, "живих" зображень, які не були частиною датасету, використововуваного для тренування моделі. Це підтверджує універсальність і практичність цієї системи, показуючи, що вона може бути використана в реальному світі з зображеннями, отриманими від користувачів, не залежно від умов зйомки або типу камери.



Uploaded image.

Probability of healthy skin: 100.00%

Probability of melanoma cancer: 0.00%

Рисунок 3.9.3 – Фотографія з телефону

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		47

Висновки до розділу

Розроблено програмний додаток для діагностики меланому, який використовує технології глибокого навчання та веб-розробки. Застосовуючи модифіковану версію моделі AlexNet і успішно навчили систему розпізнавати меланому на зображеннях шкіри.

Використання Python як основної мови програмування і бібліотек PyTorch і Streamlit сприяло ефективності розробки, дозволяючи створити надійний механізм навчання та інтуїтивний веб-інтерфейс. В результаті, користувачі можуть легко використовувати цей додаток для швидкого та зручного виявлення меланому, що може сприяти ранньому виявленню та лікуванню цього захворювання.

Детально дослідили процес побудови, тренування та оцінки моделі, а також розробили інтерфейс користувача, який зручний для використання. Мій додаток демонструє потужні можливості машинного навчання в медичній сфері, а також важливість доступного програмного забезпечення для реалізації цих технологій.

В цілому, робота є важливим кроком у напрямку використання передових технологій для покращення здоров'я людей. Цей додаток може стати основою для подальших досліджень і розробок у цій області, оскільки він демонструє, як високоточні алгоритми машинного навчання можуть бути інтегровані в корисні і доступні медичні додатки.

4 ТЕСТУВАННЯ СИСТЕМИ

Для перевірки оцінки ефективності я буду відштовхуватись від показників recall, precision та F1- score. Recall (повнота), Precision (точність) та повнота f1-міри (F1-score) є ключовими метриками для оцінки якості класифікаційної моделі, особливо у задачах бінарної класифікації. У випадку з прогнозуванням меланому, бажано максимізувати і повноту, і точність. Ви не хочете пропустити жодного випадку меланому (висока повнота), але ви також не хочете лякати людей, заявляючи, що в них меланома, коли насправді це не так (висока точність). Тому повнота f1-міри є хорошою метрикою, яка бере до уваги обидві ці величини.

4.1 Повноти

Recall, також відома як повнота, є метрикою, яка вимірює здатність класифікаційної моделі виявити усі випадки певного класу. Вона важлива в ситуаціях, де пропуск позитивних випадків може призвести до серйозних наслідків. Наприклад, в медичному контексті модель діагностики меланому з низьким показником повноти може пропустити випадки раку шкіри, що може призвести до затримки у лікуванні.

З математичної точки зору повнота визначається так:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}, \quad (4.1)$$

де True Positives - це кількість правильно класифікованих позитивних випадків, а False Negatives - це кількість позитивних випадків, які були помилково класифіковані як негативні.

Градація оцінки Recall може бути такою:

- Висока (>0.9): Модель дуже добре виявляє позитивні випадки. Це може бути особливо важливо для критичних застосувань, таких як медична діагностика.

					ІА94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		49

- Середня (0.5-0.9): Модель відома помірно добре виявляти позитивні випадки, але все ще може пропустити деякі випадки.

- Низька (<0.5): Модель погано виявляє позитивні випадки. Це може бути недостатнім для багатьох застосувань, особливо тих, що вимагають високої повноти.

Важливо зазначити, що Recall не повинна розглядатися ізольовано, а має бути розглянута разом з іншими метриками, такими як точність (Precision) і повнота f1-міри (F1-score), щоб отримати більш комплексне уявлення про ефективність моделі.

4.2 Precision

Precision (Точність) - це метрика, що вимірює, скільки з позитивних класів, які ми прогнозували, дійсно були позитивними. Ця метрика важлива в ситуаціях, де важливо уникнути помилкових позитивних прогнозів. Наприклад, у медицині, якщо модель діагностує рак, коли його немає, це може призвести до непотрібного стресу та ненужних медичних процедур.

Математично точність визначається наступним чином:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (4.2)$$

де True Positives - це кількість правильно класифікованих позитивних випадків, а False Positives - це кількість негативних випадків, які були помилково класифіковані як позитивні.

Градація оцінки Precision може бути такою:

- Висока (>0.9): Модель дуже добре уникає помилкових позитивних прогнозів. Це може бути особливо важливо в контекстах, де помилкові позитивні результати приносять значні витрати або небезпеку.

- Середня (0.5-0.9): Модель відносно добре уникає помилкових позитивних прогнозів, але все ще може час від часу робити помилки.

- Низька (<0.5): Модель часто робить помилкові позитивні прогнози. Це може бути недостатнім для багатьох застосувань, особливо тих, що вимагають високої точності.

4.3 F1-score

Повнота f1-міри (F1-score) є гармонійним середнім між точністю (Precision) та повнотою (Recall). Ця метрика є важливою, тому що вона забезпечує збалансовану оцінку точності та повноти моделі.

В основі повноти f1-міри лежить ідея, що нам не просто потрібна модель з високою точністю або високою повнотою, нам потрібна модель, яка добре справляється з обома цими аспектами. Якщо модель має високу точність, але низьку повноту, це означає, що вона може пропустити багато позитивних випадків. З іншого боку, якщо модель має високу повноту, але низьку точність, це означає, що вона може помилково класифікувати багато негативних випадків як позитивні.

Математично повнота f1-міри визначається наступним чином:

$$F1\ Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 \times Precision \times Recall}{Precision + Recall}, \quad (4.3)$$

Градація оцінки повнота f1-міри може бути такою:

- Висока (>0.7): Модель має високу точність і повноту, що означає, що вона добре справляється з класифікацією позитивних випадків і ефективно уникає помилкових позитивних прогнозів.

- Середня (0.3-0.7): Модель має помірну точність та повноту. Це може бути прийнятно для деяких застосувань, але для інших може знадобитися більш точна модель.

- Низька (<0.3): Модель має низьку точність та повноту. Це часто свідчить про те, що модель не може ефективно класифікувати позитивні випадки або має багато помилкових позитивних прогнозів.

Отже, повнота f1-міри є важливою метрикою для оцінки ефективності класифікаційних моделей, особливо в ситуаціях з нерівністю класів, коли проста точність може бути оманливою.

Висновки до розділу

Після перевірок ефективності моделі використаної у веб-додатку можна зробити деякі висновки про кожен з перевірок на основі отриманих даних.

Модель показала точність у 90%, що означає, що 90% випадків, які модель вважає позитивними (тобто меланома), справді є позитивними. У медичному середовищі, особливо при діагностиці серйозних захворювань, як-от меланома, ідеальною метою є наближення точності до 100%. Це пов'язано з тим, що помилка у визначенні може мати серйозні наслідки.

Точність 90% вже є хорошим показником, однак, зважаючи на важливість інтенсивності завдання, бажано покращити цей показник. Є різні способи, як можна це зробити:

Збільшити розмір набору даних: Моделі глибокого навчання відомі своєю здатністю вчитися на великих обсягах даних. Збільшення розміру набору даних може допомогти моделі краще узагальнювати та впоратися з шумом у даних.

Data augmentation: Це техніка, яка використовується для створення нових тренувальних випадків шляхом модифікації існуючих даних. Наприклад, можна зробити зображення більш яскравими або темними, повернути їх, збільшити чи зменшити масштаб, тощо.

Гіперпараметричний підбір: Модель може бути подальше оптимізована шляхом підбору гіперпараметрів, таких як швидкість навчання, розмір партії, кількість епох і т.д.

Енсембль моделей: Інколи, комбінування прогнозів від різних моделей може допомогти покращити загальну точність. Такий підхід використовує прогнози від кількох моделей для отримання кінцевого прогнозу.

Всі ці техніки потребують додаткового часу та ресурсів, але вони можуть допомогти покращити точність моделі, що, в свою чергу, може врятувати життя.

Повнота (Recall): Модель має високу повноту 0.936, що означає, що модель здатна виявити 93.6% випадків меланоми з усіх дійсних випадків меланоми. Висока повнота свідчить про те, що модель робить мало помилок типу II (False Negatives), коли вона помилково визначає наявність меланоми як відсутність.

F1-оцінка (F1 Score): F1-оцінка моделі є дуже високою, 0.918. Це свідчить про те, що модель забезпечує високий баланс між точністю та повнотою. Зокрема, це важливо в ситуаціях, коли нерівновага класів може спричинити заниження або завищення оцінок точності та повноти.

Загалом, ці показники свідчать про те, що ваша модель має високу ефективність для задачі виявлення меланоми. Це показує, що модель може надійно використовуватися для виявлення меланоми на основі зображень шкіри, відображаючи не тільки високу здатність до коректної класифікації позитивних випадків, але й високу здатність уникнення помилкового визначення негативних випадків як позитивних. З урахуванням серйозності медичних наслідків помилок в обох напрямках, висока точність, повнота та F1-оцінка є важливими досягненнями.

ВИСНОВКИ

У ході даного проекту було розроблено та впроваджено ефективний інструмент для виявлення меланоми на основі глибокого навчання. Застосування технологій штучного інтелекту і глибокого навчання в медицині відкриває нові можливості для раннього виявлення та лікування захворювань, таких як меланома.

Модель глибокого навчання, навчена на збірці даних з меланою, продемонструвала високу точність у виявленні меланоми. Результати виявилися високими: точність - 90%, повнота - 93.6%, повнота f1-міри - 91.8%, що свідчить про високий рівень якості моделі та її здатності правильно класифікувати випадки меланоми.

Ці результати відповідають сучасному рівню наукових та технічних знань у галузі глибокого навчання та медичної діагностики. Створена модель може бути використана як важливий інструмент для підтримки медичних спеціалістів у діагностуванні меланоми, що може врятувати життя та поліпшити якість життя пацієнтів.

Цей проект з використанням глибокого навчання включає розробку веб-додатку, що робить його більш доступним для використання як спеціалістами у галузі охорони здоров'я, так і пацієнтами. Саме тут ми бачимо, як технологія глибокого навчання може бути впроваджена у повсякденні медичні практики.

Веб-додаток було розроблено за допомогою бібліотеки Streamlit, яка відома своєю здатністю створювати інтуїтивно зрозумілі веб-інтерфейси. Це означає, що навіть користувачі, які не мають глибоких технічних знань, можуть використовувати цей додаток для аналізу зображень шкіри і отримання висновків, що генеруються моделлю глибокого навчання.

Майбутнє цього проекту видається дуже обіцяючим. З одного боку, можливе подальше вдосконалення моделі, наприклад, шляхом навчання на більшому наборі даних. Це може привести до ще більшої точності діагностики. З іншого боку, можна розширити функціональність веб-додатку, додавши нові функції, такі як звіти про діагностику або інтеграцію з електронними медичними системами.

					IA94.170BAK.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		54

Щодо потенційних сфер використання, дана технологія має величезний потенціал в галузі охорони здоров'я. Медичні заклади різного типу, включаючи лікарні, клініки, центри дерматології, можуть застосовувати цей інструмент у своїй щоденній практиці. Наприклад, дерматологи можуть використовувати цей веб-додаток як додатковий інструмент при постановці діагнозу меланоми. Це не тільки покращує точність діагностики, але і дозволяє лікарям робити це швидше, що може бути життєво важливим для пацієнтів з підозрою на рак шкіри. Іншим можливим застосуванням є інтеграція цієї моделі в мобільні додатки для самостійної перевірки здоров'я шкіри, що може бути корисним для широкої аудиторії.

На більш загальному рівні, ця робота демонструє потенціал технології глибокого навчання в медичній діагностиці та може слугувати прототипом для розробки подібних систем для інших медичних умов.

Наукова значущість цієї роботи полягає в застосуванні передових технологій машинного навчання та глибокого навчання для вирішення важливих медичних проблем. Це підкреслює роль штучного інтелекту у сучасному науковому дослідженні та демонструє, як AI може бути використано для підтримки діагностики хвороб.

З науково-технічної точки зору, ця робота включає розробку конволюційної нейронної мережі для класифікації зображень шкіри на предмет меланоми.

Соціальна значущість проекту полягає в можливості поліпшити раннє виявлення меланоми, що є критично важливим для підвищення шансів на успішне лікування. Розроблений веб-додаток може бути доступним та зручним інструментом для широкого загалу.

З економічної точки зору, впровадження таких систем може знизити медичні витрати, пов'язані з пізнім виявленням та лікуванням меланоми. Він також може збільшити продуктивність медичних працівників, дозволяючи їм зосередитися на більш складних випадках.

Основаючись на результаті цього проекту, з'являється чітка доцільність продовження досліджень у цій області. Моделі машинного навчання і глибокого навчання продовжують розвиватися та вдосконалюватися, пропонуючи все більше

					IA94.170BAK.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		55

можливостей для покращення діагностування і лікування хвороб, таких як меланома.

По-перше, можна розширити датасет для тренування та валідації моделі, включаючи дані про пацієнтів з різних етнічних груп та з різними типами шкіри. Це може допомогти покращити здатність моделі коректно виявляти меланому у різних демографічних групах.

По-друге, можна розширити функціонал додатку, наприклад, додати можливість відслідковувати динаміку змін шкірних утворень користувача чи інтегрувати цей інструмент з електронною медичною системою, що дозволить лікарям отримувати доступ до інформації про пацієнтів та їх діагностику в реальному часі.

По-третє, можна продовжувати дослідження з метою покращення алгоритмів глибокого навчання та оптимізації моделей для швидкішого та точнішого виявлення меланоми.

Всі ці напрямки покращення та розвитку можуть забезпечити значне підвищення ефективності і точності діагностики меланоми, що є важливим кроком до зниження смертності від цього захворювання. В результаті виконання дипломного проекту було спроектовано.

У цьому проекті використано різноманітні технології, що спільно дозволили створити ефективний та корисний інструмент для діагностики меланоми. Вибір середовища розробки був здійснений на основі потреб специфічного завдання та загальних принципів розробки програмного забезпечення.

Python було обрано як основна мова програмування через його широку підтримку у науковій та дослідницькій областях, особливо в галузі машинного навчання та обробки зображень. Python має розширений набір бібліотек, що робить його відмінним вибором для цього проекту.

PyTorch було використано для реалізації моделі глибокого навчання. PyTorch дозволяє легко та інтуїтивно збудувати нейронні мережі та тренувати їх на графічних процесорах, що є важливим для ефективного тренування складних моделей.

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		56

Для створення веб-додатку було використано Streamlit - це високорівневий фреймворк, що дозволяє розробникам легко створювати інтерактивні веб-додатки з можливостями машинного навчання.

Цей проект став піонером у використанні технологій глибокого навчання для діагностики меланоми, вносячи вагомий вклад у медичну сферу. Спираючись на використання сучасних методів штучного інтелекту та обробки зображень, був створений ефективний інструмент для діагностики цієї потенційно смертельної хвороби.

Перше, що варто відзначити, - це новий формат взаємодії між лікарем та пацієнтом, який запропоновано в цьому проекті. Використання веб-додатку робить цей процес більш зручним та ненав'язливим, адже пацієнтам тепер не потрібно особисто відвідувати лікаря для первинної діагностики. Вони можуть просто завантажити зображення шкіри в веб-додаток, отримати результати та, при необхідності, звернутися до спеціаліста.

Такий підхід може значною мірою сприяти вчасному виявленню меланоми, а також інших шкірних захворювань. Все це може привести до покращення статистики виживаності пацієнтів, оскільки раннє виявлення - це ключовий фактор успішного лікування меланоми.

В іншому ключовому аспекті проекту використовується модель глибокого навчання AlexNet, що є відомою своєю високою ефективністю в задачах комп'ютерного зору. Це перший випадок, коли такі методи були адаптовані для аналізу зображень шкіри в пошуках меланоми. Використання такої сильної моделі дозволило досягти високої точності в розпізнаванні меланоми, що може значно покращити якість діагностики.

Загалом, цей проект демонструє, як технології глибокого навчання можуть бути ефективно використані в медицині, створюючи потужні інструменти для підтримки вчасної та точної діагностики. Однак, це лише початок - з потенціалом глибокого навчання, можливості для подальшого використання та вдосконалення таких інструментів є майже необмеженими. Окрім впровадження нового веб-застосунку та використання глибокого навчання для діагностики, існуючу систему

					IA94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		57

було модифіковано та адаптовано для роботи з веб-додатком. Це робить додаток доступним для широкої аудиторії та зручним у використанні.

В результаті, стандартні методи діагностики меланому, які зазвичай вимагають візуального огляду шкіри лікарем, були значно розширені за допомогою автоматизованих алгоритмів глибокого навчання. Таке впровадження не лише може сприяти ранньому виявленню меланому, але і покращити загальні результати лікування.

Проект, в основі якого лежить використання глибокого навчання для діагностики меланому, є важливим кроком у використанні технологій штучного інтелекту в медицині. Цей проект показує, як технології штучного інтелекту можуть відіграти важливу роль у виявленні та діагностиці серйозних захворювань шкіри, що можуть забезпечити вчасне лікування та покращити прогнози для пацієнтів.

Глибоке навчання пробуджує нові можливості для трансформації медицини в багатьох відносинах. Цей проект є свідченням цього потенціалу, але це тільки початок. Давайте розглянемо деякі із цих можливостей більш детально:

Використовуючи стратегії, які були використані в цьому проекті, ми можемо розробити алгоритми для виявлення різних типів шкірних захворювань або навіть різних форм раку. Використовуючи медичні зображення, такі як рентгенівські знімки, МРТ або ПЕТ-сканування, можливо виявити ознаки хвороби, які ще не проявились настільки, щоб стати помітними на клінічному рівні. Такі техніки можуть надати нам можливість діагностувати захворювання набагато раніше, що у свою чергу може дозволити почати лікування раніше та збільшити шанси на успішне відновлення.

Щодо персоналізованого лікування, глибоке навчання може допомогти у розробці алгоритмів, які здатні аналізувати генетичні дані, життєвий стиль, історію хвороб пацієнта та інші важливі фактори. Ця інформація може допомогти у визначенні найефективнішого плану лікування для кожного конкретного пацієнта, що може значно покращити його шанси на успішне відновлення.

У контексті прогнозування та попередження, глибоке навчання може відігравати критично важливу роль. Моделі глибокого навчання мають здатність виявляти патерни в великих наборах даних, які можуть вказувати на початкові ознаки захворювання або на ризик розвитку певних станів. Завдяки цьому ми можемо виявити потенційні проблеми раніше та розробити стратегії для їхнього попередження або лікування.

Все вищезгадане свідчить про те, що глибоке навчання має неймовірний потенціал у сфері медицини. Цей проект являє собою важливий крок на шляху до цієї мети, але це лише вершина айсберга. Майбутні дослідження та розробки в цій області мають надзвичайно великий потенціал, і ми можемо очікувати бачити значні поліпшення в якості та доступності медичної допомоги завдяки цим технологіям.

					ІА94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		59

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Brownlee, J. (2019). "Deep Learning With Python"
- 2) Chollet, F. (2018). "Deep Learning with Python". Manning Publications.
- 3) Raschka, S., & Mirjalili, V. (2019). "Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2"
- 4) Goodfellow, I., Bengio, Y., & Courville, A. (2016). "Deep Learning". MIT Press.
- 5) Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks". In Advances in neural information processing systems (pp. 1097-1105).
- 6) Streamlit. "Streamlit Documentation"
- 7) PyTorch. "PyTorch Documentation"
- 8) Scikit-Learn. "Scikit-Learn Documentation"
- 9) Pandas. "Pandas Documentation"
- 10) Matplotlib. "Matplotlib Documentation"
- 11) Wikipedia. "Precision and Recall"
- 12) Wikipedia. "F1 Score"
- 13) LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep learning". Nature, 521(7553), 436-444.
- 14) Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). "Dermatologist-level classification of skin cancer with deep neural networks". Nature, 542(7639), 115–118.
- 15) Ronneberger, O., Fischer, P., & Brox, T. (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (pp. 234–241). Springer International Publishing.

ДОДАТКИ

					ІА94.170БАК.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		61