НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

«На правах рукопису»
УДК _____

«До захисту допущено»
Завідувач кафедри
Сергій Стіренко
(підпис) (ініціали, прізвище)
"____" _____ 2022 р.

Магістерська дисертація

зі спеціальності: *121. Інженерія програмного забезпечення*
(код та назва напряму підготовки або спеціальності)

Спеціалізація: *121. Інженерія програмного забезпечення комп'ютерних систем*

на тему: «Спосіб виявлення раку мозку за допомогою глибинного навчання»

Виконав : студент *2* курсу, групи ІМ-11мп
(шифр групи)

Мещеряков Олександр Андрійович
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник *доцент каф. ОТ, к.т.н. Селіванов В Л*
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант *проф. д.т.н. Жабін В. І.*
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2022 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет (інститут) *Інформатики та обчислювальної техніки*

(повна назва)

Кафедра *Обчислювальної техніки*

(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність *121. Інженерія програмного забезпечення*
(код і назва)

Спеціалізація *121. Інженерія програмного забезпечення комп'ютерних систем* (код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Сергій Стіренко

(підпис) (ініціали, прізвище)

« » 2022 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Мещеряков Олександр Андрійович

(прізвище, ім'я, по батькові)

1. Тема дисертації Спосіб виявлення раку мозку за допомогою глибинного навчання

Науковий керівник дисертації *доцент каф. ОТ, к.т.н. Селіванов В.Л.*

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

затверджені наказом по університету від «*09*» *11* *2022* р. № *4115-с*

2. Строк подання студентом дисертації _____

3. Об'єкт дослідження *процес визначення ракових пухлин.*

4. Предмет дослідження: *методи та засоби визначення рокових пухлин за допомогою нейронних мереж.*

5. Перелік завдань, які потрібно розробити: дослідити с*пособ конструювання мереж для виявлення ракових пухлин засобами глибинного навчання.*

6. Консультанти розділів дисертації:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормконтроль	Жабін В. І.		
1			
2			
3			
4			

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів дисертації	Примітка
1	*Вивчення літератури*	*0.3.09.2022 - 11.09.2022*	
2	*Складання і узгодження технічного завдання*	*12.09.2022 – 20.10.2022*	
3	*Написання вступної частини та огляд рішень*	*22.10.2022-9.11.2022*	
4	*Моделювання розробленого способу*	*10.11.2022-19.11.2022*	
5	*Оформлення документації ДП*	*20.11.2022-29.11.2022*	
6	*Попередній захист та проходження нормативного контролю*		
7	*Подання ДП рецензенту*		

Студент _____ Мещеряков О.А.
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Селіванов В.Л.
(підпис) (ініціали, прізвище)

РЕФЕРАТ

на магістерську дисертацію

виконану на тему: Спосіб виявлення раку мозку за допомогою глибинного навчання

студентом: Мещеряков Олександр Андрійович

Робота складається із вступу та чотирьох розділів. Загальний обсяг роботи: 83 аркушів основного тексту, 25 ілюстрації, 26 таблиць. При підготовці використовувалася література з 16 різних джерел.

Актуальність. У сфері онкологічної рентгенографічної візуалізації штучний інтелект використовується для виявлення та діагностики пухлин. Комп'ютерне виявлення історично використовувалося для візуалізації раку. Візуалізація раку була головною мішенню для виявлення на основі штучного інтелекту. Модель знаходить та окреслює зони, підозрілі на рак, без будь-якого нагляду з боку людини.

Штучний інтелект допоможе менш досвідченим радіологам виявити рак коли він є, і відкинути все, що можна прийняти за рак зменшивши таким чином кількість неправильних діагнозів і позбавивши деяких людей від непотрібного стресу, подальших тестів і процедур. Від рентгенівських знімків цілих органів до мікроскопічних знімків ракових клітин лікарі використовують візуалізаційні тести різними способами: виявляючи рак на його ранніх стадіях, визначаючи стадію пухлини, перевіряючи ефективність лікування та відстежуючи, чи рак повернувся після лікування.

Мета і завдання дослідження. Метою магістерської роботи є розробка нейронної мережі для досягнення високої точності у визначенні ракових пухлин.

Для досягнення мети дослідження поставлено і вирішено такі завдання:

- дослідження структури та архітектури нейронних мереж;
- дослідження принципи побудови нейронних мереж;

- класифікація та узагальнення методів сторення мережі;
- ефективне навчання мережі

Об’єкт дослідження – процес виявлення ракових пухлин.

Предмет дослідження – методи та засоби визначення рокових пухлин за допомогою нейронних мереж.

Методи досліджень. Для досягнення поставлених в магістерській роботі задач, використано методи побудови загорткової нейронної мережі.

Наукова новизна одержаних результатів роботи полягає у наступному:

- запропоновано спосіб конструювання мережі для виявлення ракової пухлини.
- змодельовано запропонований спосіб засобами Tenserflow та keras.

Проведене дослідження дає можливість використання запропонованого способу для виявлення ракових пухлин за допомогою загорткової нейронної мережі.

Особистий внесок здобувача. Магістерське дослідження є самостійно виконаною роботою, в якій відображено особистий авторський підхід та особисто отримані теоретичні та прикладні результати, що відносяться до вирішення задачі конструювання нейронних мереж. Формулювання мети та завдань дослідження проводилось спільно з науковим керівником.

Практична цінність. Отримані результати можуть використовуватися у майбутніх дослідженнях за напрямками:

- розробка та вдосконалення методів створення нейронної мережі;
- навчання нейронних мереж.

Ключові слова: Глибинне навчання, нейронні мережі, згорткова мережа, виявлення пухлин, робота з даними.

Пояснювальна записка
до магістерської дисертації

на тему: «Спосіб виявлення раку мозку за допомогою глибинного навчання»

Київ – 2022

ЗМІСТ

ВСТУП

Використання штучного інтелекту для подолання складнощів медицини давно проголошувалося найближчим часом. Він має довгу історію, яка сягає 1970-х років, коли системи підтримки клінічних рішень (CDSS) , що вимагали від людей надавати правила для методів дерева рішень і вручну вибирати атрибути для включення в ці експертні системи.

У сфері онкологічної рентгенографічної візуалізації штучний інтелект використовується для виявлення та діагностики пухлин. Комп'ютерне виявлення історично використовувалося для візуалізації раку, але воно не продемонструвало високої клінічної цінності. Отже, візуалізація раку була головною мішенню для виявлення на основі штучного інтелекту. Модель знаходить та окреслює зони, підозрілі на рак, без будь-якого нагляду з боку людини. Штучний інтелект допоможе менш досвідченим радіологам виявити рак коли він є, і відкинути все, що можна прийняти за рак зменшивши таким чином кількість неправильних діагнозів. Від рентгенівських знімків цілих органів до мікроскопічних знімків ракових клітин лікарі використовують візуалізаційні тести різними способами: виявляючи рак на його ранніх стадіях, визначаючи стадію пухлини, перевіряючи ефективність лікування та відстежуючи, чи рак повернувся після лікування.

Штучний інтелект може краще відрізняти рак легенів від неракових змін на комп'ютерній томографії, потенційно зменшивши кількість помилкових спрацьовувань і позбавивши деяких людей від непотрібного стресу, подальших тестів і процедур. Наприклад, команда дослідників навчила алгоритм глибокого навчання, щоб знаходити рак легенів і спеціально уникати інших змін, які виглядають як рак. Під час лабораторних тестів алгоритм дуже добре ігнорував неракові зміни, схожі на рак і добре знаходив рак.

Існує значна кількість алгоритмів виявлення зображень злоякісних пухлин молочних залоз. Окрім раку молочної залози, виявлення та діагностика на основі штучного інтелекту зараз використовується для різних типів пухлин. Наприклад, відомо, що при раку передміхурової залози мультипараметрична МРТ покращує виявлення клінічно значущих злоякісних новоутворень, але такі проблеми, як варіабельність між спостерігачами, залишаються. Виявлення на основі штучного інтелекту має потенціал для подолання цих проблем за допомогою алгоритмів МЛ.

Алгоритми візуалізації, засновані на штучному інтелекті використовуються в клінічній практиці для виявлення та відстеження потенційно ракових уражень і для керівництва лікуванням. Наприклад, програмне забезпечення наразі дозволяє комплексно виявляти та відстежувати легеневі вузлики, прогнозувати злоякісність легенів серед виявлених уражень на КТ-зображеннях з низькою дозою та включати рекомендації щодо лікування. Глибока нейронна мережа також була розроблена для виявлення збільшених лімфатичних вузлів або поліпів товстої кишки на КТ-зображеннях і для покращення виявлення поліпів товстої кишки під час колоноскопії за допомогою системи комп'ютерного виявлення DL у реальному часі. Також розширена інтерпретація ендоскопічних зображень за допомогою штучного інтелекту постійно підвищує точність виявлення раку стравоходу.

Моделі візуалізації на основі ШІ також використовуються для характеристики пухлини. Характеристика може включати анатомічну сегментацію пухлин, яка дозволяє програмному забезпеченню ідентифікувати межі ураженої тканини серед нормальної анатомії, і класифікацію підтипу пухлини, яка використовує підказки щодо інтенсивності сигналу, текстури, форми та інших дескрипторів для встановлення діагнозу. Сегментація — двовимірна або об'ємна — може використовуватися в клінічній практиці для прийняття рішень про лікування, наприклад планування терапії з радіації, однак існує упередженість між

спостерігачами в ручній сегментації пухлини. Алгоритми на основі штучного інтелекту мають потенціал для подолання цих упереджень.

Одним із таких алгоритмів є продукт, що виявляє метастази в головному мозку та проводить сегментацію для стереотаксичної радіохірургії. Анотація даних обстежень, таких як КТ-сканування, може створювати об'ємні одиниці або вокселі, які схожі на 3D-пікселі. Ці екстраполяції даних можуть призвести до інформації на основі комп'ютерного зору. Радіомічний аналіз передбачає автоматизоване виділення клінічно значущої інформації з радіологічних зображень, і його можна використовувати для розробки радіомічних біомаркерів шляхом перевірки радіомічних підписів з використанням генетичних, гістологічних та інших форм даних.

Також через зростання обчислювальної потужності, збільшується доступність електронних медичних карток (EHR), мультиоміка та дані медичних зображень, а також прогрес у DL, зокрема згорткових нейронних мережах, зробили революцію в розробці та використанні алгоритмів ШІ та CDSS в аналізі зображень, пов'язаних із раком. Поточні дослідження для підтримки застосування штучного інтелекту в геноміці раку дозволять раннє виявлення мультиракових захворювань і визначення місця походження пухлини. Це може змінити системи первинного обстеження раку, особливо для менш поширених і рідкісних видів раку, і це може покращити стратегії спостереження за хворими на рак.

Постійний прогрес у ML на основі візуалізації також може призвести до розробки моделей, які оцінюють ризик різних типів раку, підвищують діагностичну точність раку або прогнозують пов'язані з цим результати захворюваності та смертності. Це може дозволити індивідуальні стратегії скринінгу та профілактики, підходи до лікування та спостереження за постраждалими, а також може підтримувати віртуальну біопсію для класифікації патологічних і геномних особливостей, пов'язаних з діагнозом раку.

Лікарі, що займаються онкологічними захворюваннями, зараз користуються перевагами перших інструментів штучного інтелекту, які тільки стають доступними в EHR, і немає недоліку в очікуванні майбутніх наслідків. Корисність штучного інтелекту в онкології можна оцінити, візуалізувавши здоров'я кожного пацієнта у вигляді цифрової фотографії. Медицина намагається розшифрувати це зображення попіксельно, використовуючи дослідження, що ґрунтуються на зрозумілій патофізіології, щоб точно з'ясувати окремі пікселі, розкидані повсюди. Штучний інтелект використовує комплексний підхід, починаючи з погано сфокусованого наближення, але з підвищенням чіткості в міру вдосконалення баз даних, алгоритмів і обчислювальної потужності.

РОЗДІЛ 1
АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1. 1 Опис завдання

Завдання даної роботи полягає в створенні штучного інтелекту , що дає змогу користувачам перевіряти потрібне їм фото мозку на наявність пухлини. Програма повинна надавати можливість швидкої та точної перевірки ряду завантажених фотографій.

Штучний інтелект, що створюється в даній магістерській роботі, повинен реалізувати зручний у користуванні інтерфейс що складається з клієнтської частини, серверної частини з загорткової мережі, що взаємодіють між собою для забезпечення попиту клієнта.

Нейромережа складається зі загорткових шарів в яких міститься найбільш ефективна кількість нейронів з певною кількістю вагів. Шари поділяються на вхідний шар, прихованих шарів з функцією активації та вихідний шар для отримання результату. Штучний нейрон у прихованому шарі працює як біологічний нейрон в мозку - він приймає в собі ймовірнісні вхідні сигнали, працює на них і перетворює їх у вихід.

Щоб ефективно класифікувати різні типи раку, необхідно виділити проміжні семантичні ознаки, які можуть описати його тип. Поточні функції переважно включають колір, текстуру тощо. Найбільш широко використовувана колірна функція — це вилучення колірних параметрів. Найменша одиниця – це кадр зображення. Нормалізований шар нормалізує об'єкти між реальними площинами об'єктів і знищує інформацію про становище об'єктів. Це може посилити різницю між площинами ознак. Тому нормалізований шар зазвичай використовується у поєднанні з шаром згортки.

1.1.1 Застосування моделі

Моделі ML на основі візуалізації можуть дозволити виявити проблемні зони на ранніх етапах, зменшити вірогідність помилки лікаря та передбачити майбутні результати для пацієнтів з раком, такі як локорегіональний рецидив, віддалений рецидив і смертність. Численні новітні моделі МЛ на основі візуалізації дозволяють передбачити клінічні результати для пацієнтів з раком, такі як загальне виживання та виживання без захворювань. Ця інформація може сприяти індивідуальному догляду за хворими на рак, включаючи спостереження та оптимізовану стратегії запобігання рецидиву. Радіомічний аналіз і новітні моделі ML на основі візуалізації продемонстрували потенціал для прогнозування пухлинної патології та геномних змін. Це може втілити в життя діагностику та інформацію про біомаркери без фактичного взяття зразків, що призведе до того, що називається «віртуальною біопсією». Наразі для гліобластоми розробляються моделі на основі неінвазивної візуалізації, що передбачають генетичні зміни в пухлині та впливають на клінічне лікування. Хоча потрібна додаткова робота для стандартизації даних візуалізації та створення відтворюваних моделей ML, цілком вірогідно, що в майбутньому ці моделі, які виходять за межі діагностики, суттєво вплинуть на лікування раку.

З прискореним зростанням цифрової патології існує кілька застосувань штучного інтелекту в аналізі патологічних зображень для діагностики, класифікації та інтерпретації прогностичних біомаркерів. Важливі досягнення були зосереджені на автоматизації трудомістких завдань для підвищення ефективності патологоанатомів, дозволяючи їм збільшити час, який вони виділяють на виконання завдань високого рівня, прийняття рішень, особливо тих, які пов’язані зі складністю та факторами, пов’язаними з проявами захворювання.

1.1.2 Проблеми розробки

Проблеми в розробці, впровадженні та підтримці моделей ШІ включають відсутність стандартизації збору та управління даними, властиве зміщення в навчальних наборах даних, обмежена проспективна клінічної перевірки моделей, потреба в бездоганній інтеграції в клінічні робочі процеси без додавання канцелярського чи когнітивного тягаря для постачальників, відсутність нормативно-правової бази і обмежена відтворюваність під час передачі між системами охорони здоров'я та населенням, враховуючи обмежену взаємодію з EHR та динамічний характер даних і вказівок щодо найкращої практики. Паралельні інвестиції в людей і ресурси, необхідні для вирішення цих проблем, повинні відбуватися паралельно з інвестиціями, які стимулювати інновації технології ШІ.

1.1.3 Штучний інтелект

Штучний інтелект відноситься до моделювання людського інтелекту в машинах, які запрограмовані думати як люди і імітувати їх дії.

Основною характеристикою штучного інтелекту є його здатність класифікувати та робити дії, з найкращими шансами для досягнення конкретної цілі. Машинне навчання, відноситься до концепції, згідно з якою програми без допомоги з боку людини мають можливість автоматично навчатися на нових даних та адаптуватися згідно нових умов. Надання та згодування величезних обсягів даних що не мають певної структури забезпечує автоматичне навчання методами глибокого навчання.

Штучний інтелект – це інстинктивний механізм, який виконує різні завдання, такі як спостереження, навчання та міркування, це метод імплантації інтелекту в машини, які здатні виконувати завдання, для яких історично були потрібні люди. Програми на основі нейронних мереж швидко розвиваються з точки зору розгортання, адаптації, швидкості обчислень та

можливостей.. Системи на основі нейронних мереж звели до мінімуму повторення дій людини і можуть давати результати в короткий час.

Штучний інтелект створив методи машинного та глибокого навчання для вирішення проблем мереж інтернету речей. Метод глибокого навчання використовується для вирішення проблеми енергоефективності інтернету речей. За допомогою методів ML програми інтернет речей виявляє шаблони, аномалії та роблять припущення на основі величезної кількості інформації. Основні проблеми штучного інтелекту полягають у наступному.

- Довіра у використанні: штучний інтелект — це наука, технології та алгоритми, про які більшість людей не знають, що ускладнює їхнє прийняття.

- Несправність програмного забезпечення: за допомогою штучного інтелекту, що управляє пристроями та алгоритмами, функції прийняття рішень негайно делегуються програмному забезпеченню керованому кодом. Автоматизація унеможливлює визначення джерел відмов і несправностей.

- Безпека даних: можливості машинного навчання та прийняття рішень у системах штучного інтелекту зосереджені на величезній кількості захищеної інформації, переважно критичного та особистого характеру. Це робить його вразливим для дуже серйозних проблем, таких як втрата даних та шахрайство з ідентифікацією.

- Упередженість алгоритму: штучний інтелект заснований на даних та алгоритмах. Стабільність здатності нейромережі приймати рішення залежить виключно від точності тестування та використання достовірних та неупереджених доказів.

1.2 Аналіз існуючих рішень

1.2.1 Google Lens

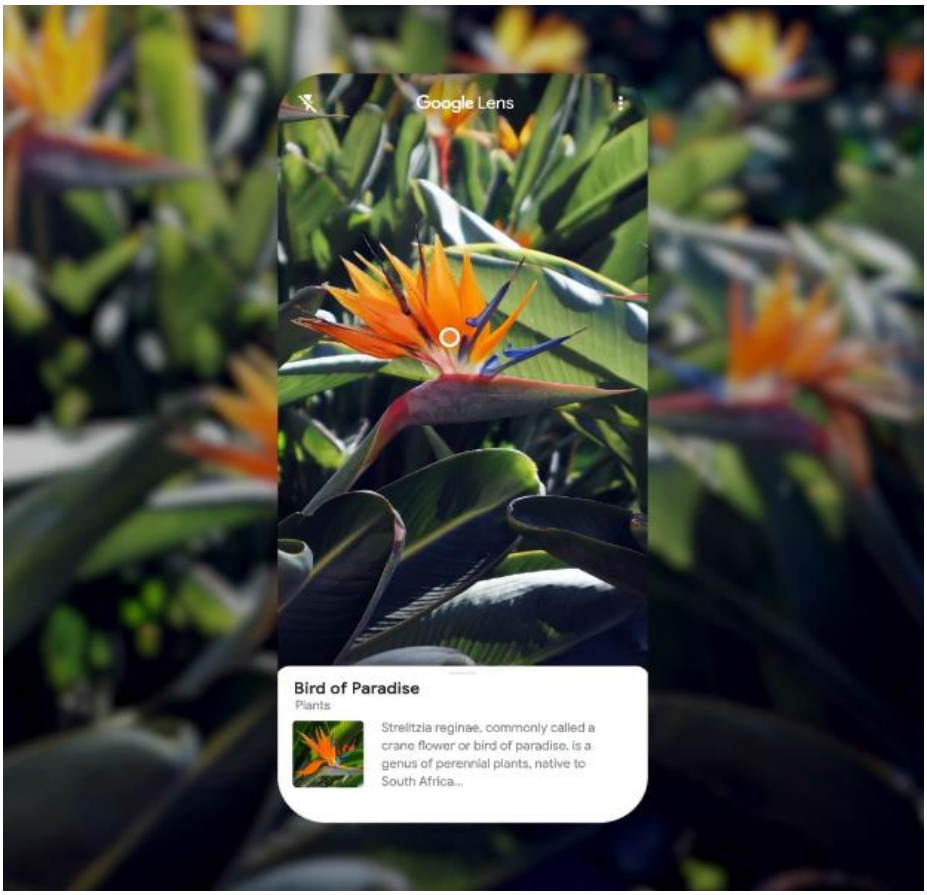

Рис. 1.1. Демонстрація використання Google Lens

Google Lens — це програма для розпізнавання зображень на базі штучного інтелекту, яка використовує камеру смартфона та глибоке машинне навчання для захоплення зображень та надає відповідну інформацію, пов'язану з об'єктами, які вона ідентифікує. Наприклад, якщо ви наведете телефон на квітку, він покаже інформацію про цю квітку, а також де можна купити ці та подібні квіти тощо. І тому він використовує візуальний аналіз з урахуванням нейронної мережі.

Програма намагається ідентифікувати об'єкт, зчитуючи етикетки та текст, QR-коди, штрих-коди тощо, та відображати відповідну інформацію.

В основі програми лежить технологія розпізнавання зображень (за допомогою камер смартфона). Програмне забезпечення стане в нагоді в багатьох ситуаціях, починаючи зі збереження контактної інформації з візитки і завершуючи визначенням породи собаки, яку ви зустріли на вулиці.

Також технологія має такі функції:

- перенесення відсканованих текстів в електронний формат для перегляду на комп'ютері або ноутбуці в браузері Chrome,
- автоматичний пошук формулювань для складних термінів і навіть сканування текстів для їхнього прослуховування без передруку або копіювання в окремі програми.
- визначення об'єктів та знаходження їх у мережі, а також «пізнавати» страви в ресторані по фотографії з меню, надаючи вам повну інформацію про інгредієнти і не тільки
- сканування та покупка товару
- переклад тексту
- сканування штрих-кодів
- копіювання тексту з телефону на комп’ютер

Google Lens інтегрований у Google Фото та Google Assistant, тому ви можете отримати до нього доступ із цих програм. Якщо ваш телефон може використовувати Google Lens, ви побачите значок у програмі Google Фото.

Рис. 1.2. Набір функцій

Коли ви використовуєте Google Lens, зображення завантажується з вашого телефону на сервери Google, і тоді починається використання штучного інтелекту. Використовуючи штучні нейронні мережі, Google Lens аналізує зображення, щоб визначити, що воно містить.

Як тільки Google Lens визначає зміст і контекст зображення, програма надає вам інформацію або дає можливість виконати контекстно-відповідну дію.

Наприклад, якщо ви бачите книгу на журнальному столику вашого друга, зробіть знімок і торкніться піктограми затвора Google Lens. Google Lens автоматично визначає автора та назву книги, а також надає вам огляди та іншу інформацію.

Google lens використовує комп'ютерний зір, машинне навчання та графік знань, щоб розпізнавати такі об'єкти, як рослини, тварини, а також виділяти, копіювати, вставляти, перекладати текст із зображень або документів.

Region proposal network – це повністю згорткова мережа, яка одночасно передбачає межі об'єктів та оцінки об'єктності в кожній позиції. RPN навчений генерувати високоякісні пропозиції регіонів, які

використовуються Fast R-CNN для виявлення. Цей компонент повідомляє об'єднаної мережі, де шукати.

Оптичне розпізнавання символів (OCR) використовує мережу регіональних пропозицій для рамок, що обмежують, на рівні символів, які можна об'єднати в рядки для розпізнавання тексту.

Граф знань - об'єктив використовує граф знань для надання контекстуальних підказок, наприклад, чи є слово, ймовірно, власним ім'ям і не повинно бути виправлено правопис, та інші подібні деталі.

Convolutional neural network - об'єктив використовує CNN для виявлення зв'язкових текстових блоків, таких як стовпці або тексту в єдиному стилі або кольорі. А потім усередині кожного блоку він використовує такі сигнали, як вирівнювання тексту, мова та геометричне співвідношення абзаців, щоб визначити їхній остаточний порядок читання.

Об'єктив використовує алгоритми нейронного машинного перекладу Google Translate для перекладу цілих пропозицій за раз, а не послівно, щоб зберегти правильну граматику та дикцію.

Перелічені алгоритми працюють разом, щоб об'єктив Google працював з найбільш ефективно можливою швидкістю.

1.2.2 Ai-derm

Ai-derm— це веб-додаток для дерматології на основі штучного інтелекту, який запустили щоб було легше з’ясувати, що може відбуватися з вашою шкірою. Після запуску додатку для його використання потрібно скористайтеся камерою свого телефону, щоб зробити три зображення шкіри, волосся чи нігтів під різними кутами. Потім вам зададуть запитання про тип вашої шкіри, як довго у вас є проблема та інші симптоми, які допоможуть інструменту звузити можливості.

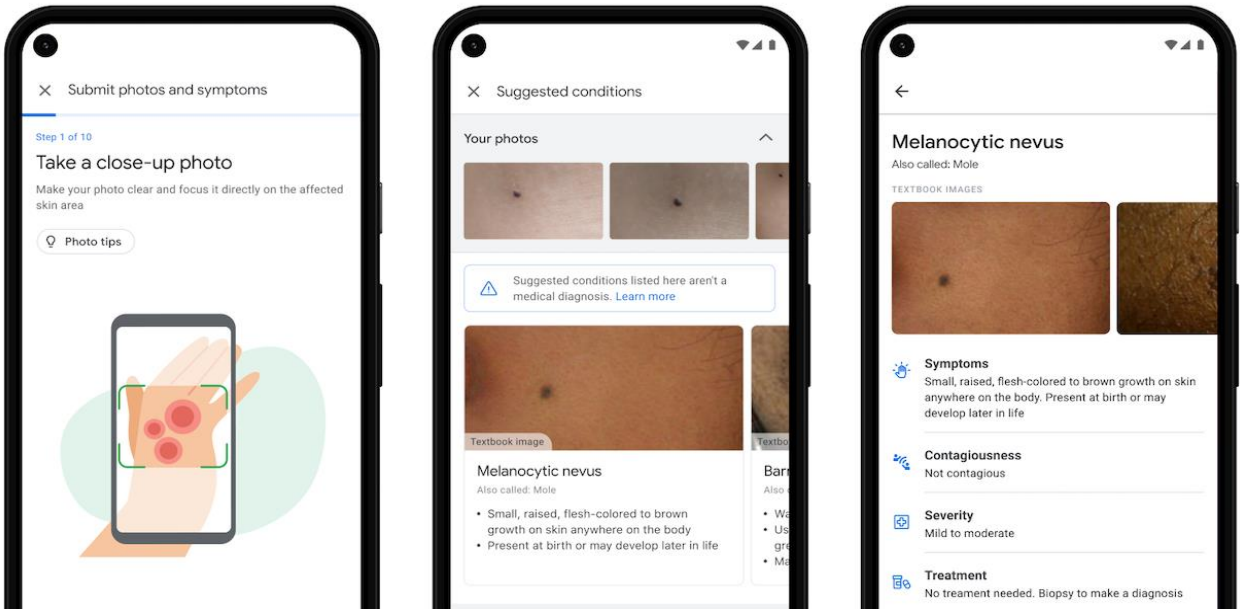

Рис. 1.3. Демонстрація використання Ai-derm

Модель штучного інтелекту аналізує цю інформацію та спирається на свої знання про 288 умов, щоб надати список можливих умов відповідності, які можна потім дослідити далі. Для кожного відповідного стану інструмент покаже інформацію, перевірену дерматологами, і відповіді на поширені запитання, а також схожі відповідні зображення з Інтернету. Інструмент не призначений для встановлення діагнозу чи заміни медичної консультації, оскільки багато захворювань вимагають огляду клініцистом, особистого огляду або додаткового тестування, наприклад біопсії..

Модель враховує такі фактори, як вік, стать, раса та тип шкіри — від блідої шкіри, яка не засмагає, до коричневої шкіри, яка рідко обгорає. Було розроблено та налаштувано модель із зашифрованими даними, що охоплюють близько 65 000 зображень і даних про діагностовані захворювання шкіри, мільйони підібраних зображень проблем шкіри та тисячі прикладів здорової шкіри — для різних демографічних груп.

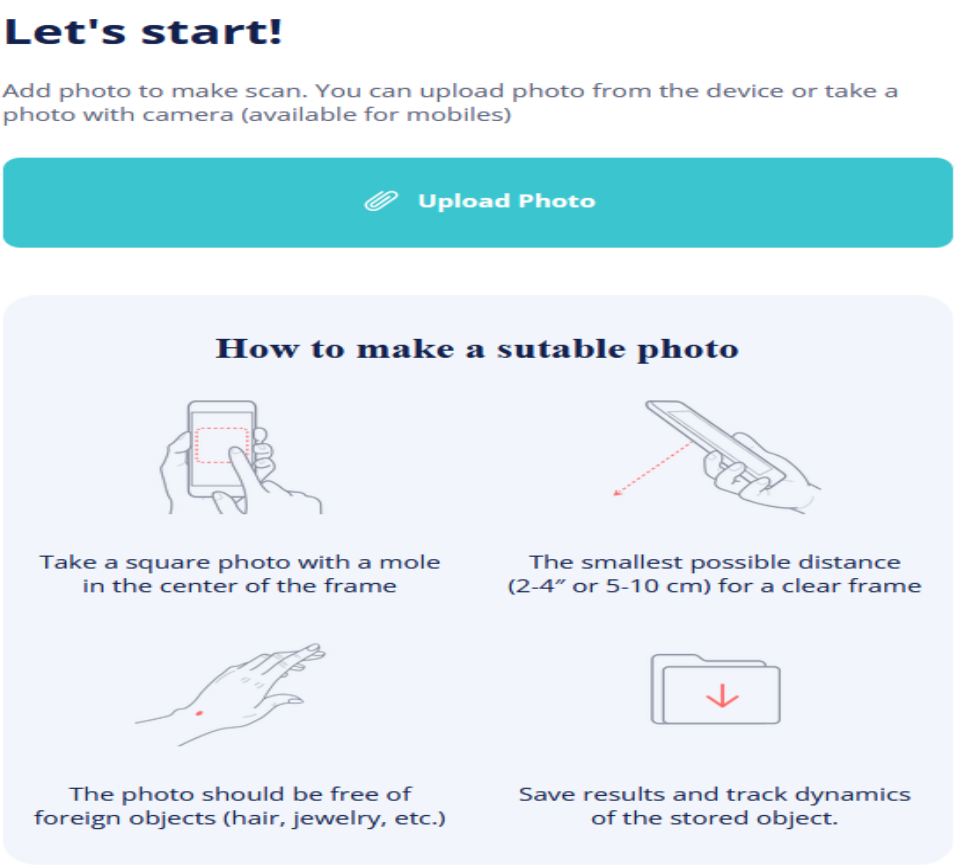

Рис. 1.4. Інтерфейс завантаження файлу Ai-derm

Коли ви використовуєте Ai-derm, зображення завантажується з вашого телефону на сервери, і тоді починається використання штучного інтелекту. Використовуючи згорткову нейронну мережу (CNN), навчену із зображень безпосередньо використовуючи лише пікселі та мітки захворювань як вхідні дані, Ai-derm аналізує зображення, щоб визначити, що воно містить. Після аналізу Ai-derm визначає присутні на фото проблемні ділянки шкіри та якщо так, то вказую яка саме це хвороба її симптоми та проблеми.

Таблиця 1.1

Основні характеристики існуючих рішень

Назва	Призначення	Засоби
Google lens	Використовується для розпізнавання зображень на наступних дій на основі даних використовуючи глибоке машинне навчання.	Region proposal network, Convolutional neural network
Ai derm	веб-додаток для дерматології на основі штучного інтелекту, який запустили щоб було легше з’ясувати, що може відбуватися з вашою шкірою	Convolutional neural network

ВИСНОВОК ДО РОЗДІЛУ 1

В першому розділі були наведені приклади вже створених додатків з використання згорткових нейронних мереж як основного інструменту для класифікації та розпізнавання об'єктів. Впродовж розділу були розглянуті елементи розробленого інтерфейсу та деталі роботи кожного з розглянутих додатків.

Процес оптичного розпізнавання складається з згорткової мережі, яка одночасно передбачає межі об'єктів та оцінки об'єктності в кожній позиції. RPN навчений генерувати високоякісні пропозиції регіонів, які використовуються Fast R-CNN. Модель враховує значну кількість факторів, що були визначенні завчасно щоб надати список можливих умов відповідності, які можна потім дослідити далі, також модель охоплюють значну кількість зображень і даних про діагностовані захворювання для різних демографічних груп для збільшення точності передбачування.

Також були розглянуті проблеми розробки і впровадження з якими може зтикатися модель для цієї сфери, основними є потреба в бездоганній інтеграції в клінічні робочі процеси та збір доцільних даних.

Були розглянуті застосуваня моделі у сфері охорони здоров’я. Моделі ML можуть дозволити виявити проблемні зони на ранніх етапах, зменшити вірогідність помилки лікаря та передбачити майбутні результати для пацієнтів з раком, такі як локорегіональний рецидив, віддалений рецидив і смертність Таким чином, актульною є задача розробки способу виявлення ракових пухлин за допомогою засобів нейронних мереж.

РОЗДІЛ 2

Аналіз інструментів для створення штучного інтелекту

2.1 Python

Python — високорівнева мова програмування основною концепцією якої є об’єктно орієнтоване програмування. Мова має вкладені в неї структури даних, які у зв’язці з перевіркою типів даних під час виконання програми та зв’язування викликуваних методів роблять необхідні умови для пришвидшення розробки додатків, а також для використання як мови сценаріїв або для з’єднання існуючих компонентів.

Інтерпритатор python надає широку вибірку стандартних бібліотек для різних задач які доступні для користування задарма для всіх існуючих платформ та систем. Мова надає підтримку модульності коду та повторного його використання

Python не має етапу компіляції, шо забезпечує підвищену продуктивність, та цикл редагування програмного коду засобами від проблемних шаблоні, тестування програми на наявність багів та налагодження для поз буття багів відбувається швидше за більшість інших мов програмування.

Мова має спрощений механізм налагоджування програм ніж на інших мовах. Однією з особливостей мови є те що помилка чи хибний вхід не спричиняє помилку сегментації, в цьому випадку інтерпретатор викликає виняток. У тому випадку коли програма не може знайти виняток, тоді інтерпретатор пересилає трасування стека. Іншою особливістю є те що налагоджував вихідного коду надає змогу перевірити змінні у локальній та глобальній видимості, надавати оцінку виразам, установлювати контрольні точки для дебагу коду та крок за кроком виконувати код програми.

Переваги:

- Простий синтаксис, що досить просто сприймається. Це пришвидшує створювання проекту та його поліпшення. Також це допомагає легше його вивчати.
- Універсальність мови. Python можна використовувати для розробки веб-додатків, аналітики данних, тестування ПЗ, машинного навчання.
- Відкритий код, який дозволяє безкоштовно використовувати та поширювати його.
- Архів модулів і бібліотек Python — пакетів коду, створених сторонніми користувачами для розширення можливостей Python — величезний і постійно зростає.
- Інтерпритатор реалізований практично на всіх платформах та системах.
- У Python є велика та активна спільнота, яка вносить свій внесок у пул модулів і бібліотек Python і є корисним ресурсом для інших програмістів. Значна кількість користувачів мови є ознакою, що якщо програмісти стикаються з проблемою для якої складно знайти рішення, це рішення вже могло бути знайдено кимось хто стикався з такою ж задачею раніше, що пришвидшить її вирішення.

Недоліки:

- Низька ефективність пам'яті щоб спростити життя розробнику, Python потребує багато пам’яті, це може бути трохи проблематично, якщо ви розробляєте програми, де вам потрібно оптимізувати пам’ять

- Повільна швидкість. Після високого споживання пам’яті недостатня швидкість є одним із найбільших недоліків Python. Оскільки він виконує код один рядок за раз, швидкість виконання часто знижується. Там, де швидкість важлива для проекту, Python не можна використовувати для кодування.
- Доступ до бази даних. Хоча на Python легко програмувати, рівень доступу до бази даних недостатньо розвинений порівняно з іншими технологіями, такими як ODBC.
- Слабкий у мобільних обчисленнях, через високе використання пам’яті та низьку швидкість, він зазвичай не використовується для програмування інтерфейсу чи розробки мобільних додатків.
- Помилки виконання.Оскільки python це мова з динамічною типізацією то типи даних можуть раптово змінюватися. Змінна, що містить рядок, пізніше може містити цілим числом, що без належного догляду та коментування може призвести до помилок виконання.
- Несумісність версії. Багато фреймворків і інструментів працюють лише для певних версій і не можуть використовуватися з іншими. Наприклад, Python 2, який вважається застарілою версією, має кілька фреймворків, створених спеціально для нього, вони несумісні з 3 версією мови.
- Мова не має попередньо створених статистичних моделей або тестів. Щоб полегшити представлення та аналіз величезних обсягів даних програмою Python, необхідно використовувати сторонні бібліотеки.

2.1.1 Використання python

Основним застосуванням мови програмування Python є розробка веб-сервісів і програмного забезпечення, автоматизації завдань та робота з данними, а саме їх аналіз та візуалізація. Оскільки його відносно легко вивчити, Python був використаний для спрощення повсякденних задач у сферах фінансів, научних сферах та сферах обчислень.

Python використовую для:

- Навчання нейромереж
- Створення веб-сервісів
- Автоматизація або програмування сценаріїв
- Тестування програмного забезпечення
- Повсякденні задачі

2.1.2 Аналіз даних і машинне навчання

Python став основним продуктом науки про дані, адже мова надає можливість програмістам у сфері аналітики даних та іншим професіоналам використовувати цю мову для проведення масивних обчислень даних використовуючи внутрішні бібліотеки, створення візуалізацій даних, створення алгоритмів машинного навчання, обробки та аналізу даних, а також виконання інших завдань, пов’язаних із даними.

Python може створювати широкий спектр різних візуалізацій даних, наприклад лінійні, секторні та стовпчасті діаграми , 3D графіки. Мова надає низку бібліотек, які дозволяють програмістам створювати ПО для аналізу значних об’ємів даних і створення нейромереж швидше й ефективніше, наприклад бібліотеками TensorFlow і Keras.

2.1.3 Тестування програмного забезпечення

При розробці ПО виникає потреба у тестуванні написаного коду використання функцій на різних даних та перевірка стабільності роботи програми на різних сбірках. За домогою мови програмування Python можна вирішувати такі завдання як контроль якості продукту, відслідковування і налагоджування помилок при розробці та тестування створенного програмного продукту для виявлення помилок. Мова надає можливість створення автоматизованих тестувальних наборів для нового функціоналу.

2.1.4 Датасет

Вибірка данних з дата сету складається з прикладів наявності та відсутності якоїсь риси. В нашому випадку це є пухлина та немає.

В якості дата сету було обрано Br35H :: Brain Tumor Detection 2020 що складається з 3000 Цей набір даних містить МРТ-зображення пухлини головного мозку. Існує дві папки папки, одна з яких представляє зображення нормального мозку – 1500 зображень, а інша - зображення пухлини – 1500 зображень. Відношення між категоріями 1 до 1. Всього в обох папках знаходиться 3000 зображень в обох цих папках. Зображення які мають різний розмір, наприклад 320 x 240, 620 x 620 у форматі JPEG. Загальна вага складає 100мб .

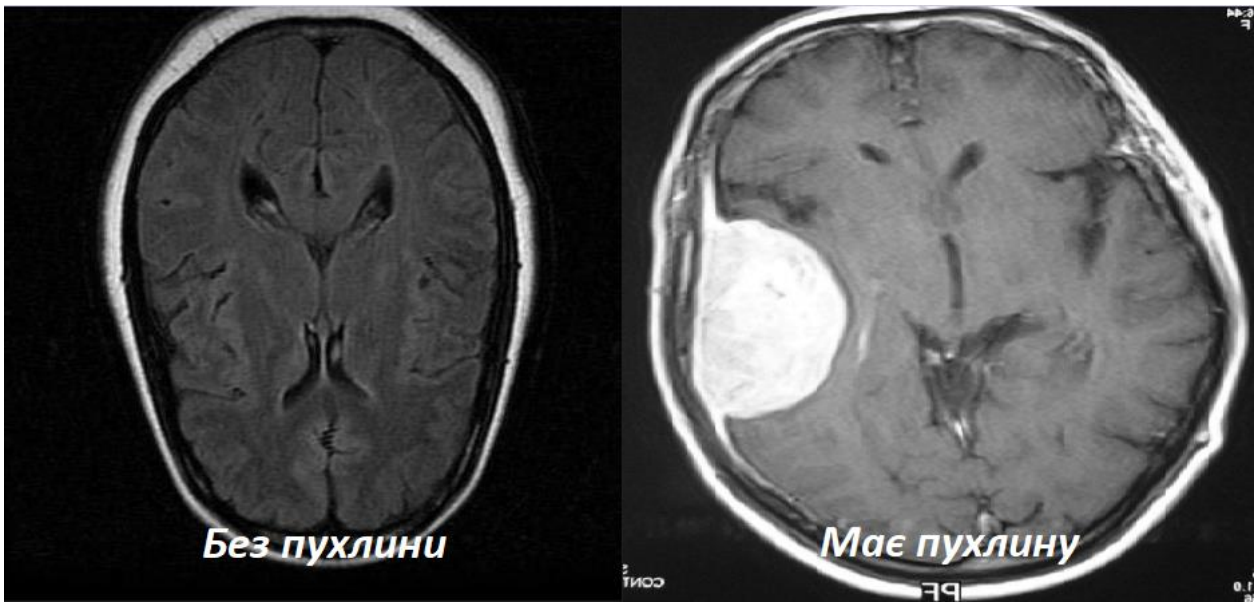

Рис. 2.1. Приклад нормального та пухлинного зображення головного мозку

Зображення поділені таким чином 2137 зображення для навчання, 579 зображень для валідації та 284 зображень для тестування. З 2137 навчальних зображень - 1199 зображень є зображеннями пухлин і 938 зображень – без пухлинними зображеннями., з 579 валідаційних зображень відібрано 201 пухлинних та 378 непухлинних зображення. Серед 284 тестових зображень 100 зображень пухлин і 184 без пухлинних зображень.

Таблиця 2.1

Розподіл датасету

Назва папки	Без пухлини	З пухлиною	Всього
Test	184	100	284
Train	938	1199	2137
Validation	378	201	579

2.2 Алгоритм k-середніх

Алгоритми машинного навчання бувають різних видів, кожен з яких призначений для певної задачі що стоїть перед ним. Кластеризація К-середніх — один із методів, корисних для сегментації.

У неконтрольованих алгоритмах відсутні розмічені дані, за домогою яких є можливість оцінити їх продуктивність.

Дані які мають найбільший відцоток схожості один між одним за групою ознак складають пари та групи в кластер це концепція яка закладена в основу алгоритму к-середніх.

Для розрахунків схожості між кластерними групами можуть використовуватися різниця між вікок, заробітною платою, витратами на споживання споживачів

Для використання потрібно кількість кластерів, на які хочете розділити дані при навчанні моделей k-середніх. Початок кожної моделі – центроїв, представляє собою розміщені у випадковому порядку змінні, які формують центральну частину кожного кластера.

Модель перевіряє навчальні дані і присвоює їх кластеру з найближчим центром тяжкості. Після проходження навчання приклади були класифіковані, параметри центроїдів повторно налаштовуються, щоб помістити їх у середину їх кластерів.

У доповненні до кластеризації k-середніх локтевий підхід є ефективним методом сегментації даних для визначення ідеальної кількості кластерів сегментації машинного навчання. Оцінюючи їх відстань від кожного з кластерів центроїдів, ваша модель машинного навчання може вирішити, до якого сегменту належать нові клієнти після її навчання. Для цього є безліч додатків. Модель машинного навчання допоможе вам визначити ваш сегмент клієнтської бази та найбільші продукти, пов’язані з цим сегментом.

Кластеризація К-средніх — це швидкий і ефективний метод сегментації машинного навчання. Однак для цього повинна бути обрана цільова аудиторія для маркетингових зусиль та елементи, які будуть для них найбільш важливими. Якщо ваша реклама буде орієнтована, наприклад, на певні місця, географічне положення не матиме значення, і вам краще відфільтрувати дані за цим регіоном.

Таким же чином, якщо ви просуваєте продукт цілеспрямований на чоловічу аудиторію, ви повинні відфільтрувати дані про своїх клієнтів, щоб включити тільки чоловіків і відмовитися від використання поля в якості одного з атрибутів вашої моделі машинного навчання. За інших обставин ви захочете надати додаткову інформацію, наприклад, про речі, які вони вже купили. У цій ситуації вам потрібно створити матрицю клієнт-продукту, яка представляє собою таблицю з клієнтами у вигляді рядків і речей у вигляді столбців, а також кількість речей, куплених на пересіченні кожного клієнта та товару.

При надто великій кількості продуктів, надається можливість створення вкладень, у яких продукти представлені у вигляді значень у великому векторному просторі.

2.3 Tenserflow

TensorFlow є бібліотекою з відкритим кодом для обчислень і великомасштабного машинного навчання. Бібліотека об’єднує веливу кількість різних моделей і алгоритмів машинного та глибокого навчання для створення нейронних мереж. Він використовує мови Python або JavaScript, щоб забезпечити зручний інтерфейсний API для створення додатків, одночасно виконуючи ці додатки на високопродуктивній C++. .

TensorFlow також має широку кількість модулів навчених моделей, з можливістю застосування їх у власних проектах. Має засоби щоб малювати графіки потоків даних, вони описують переміщення даних через графік або ряд вузлів обробки. Кожен з вузлів являє собою математичну операцію, а кожне з’єднання між вузлами є багатовимірним масивом даних або тензором.

Програми TensorFlow придатні до використання на більшості середовищ виконання та на всіх системах: локальній машині, хмарному кластері, пристроях на базі систем iOS і Android, операційних системах Windows та Linux, центральних або графічних процесорах. Якщо ви

використовуєте власну хмару Google, ви можете використати спеціальні засоби для обробки TensorFlow Google для прискорення обробки моделі. Однак отримані моделі, створені TensorFlow, можна розгорнути на будь-якому пристрої, де вони використовуватимуться для надання прогнозів.

TensorFlow версії 2.0, багато в чому оновив функції та можливості фреймворка, зробивши його легшим та швидшим для роботи (Keras API для навчання моделей) і більш продуктивним. Розроблене нове API робить простішим проведення розподіленого повторення. Підтримка TensorFlow Lite дає змогу розгортати моделі на телефонах що збільшує кількість платформ на яких працює бібліотека.

Навчену модель можна використовувати для доставки прогнозів як служби через контейнер Docker за допомогою мережі зі з'єднання між клієнтом та сервером або gRPC API. Kubernetes підійде для складніших сценаріїв технічної підтримки

Усі вузли та тензори в TensorFlow є об'єктами мови програмування Python, а програми TensorFlow самі є програмами Python. Фактичні математичні операції, однак, не виконуються в Python, це можливо тому що Python перенаправляє трафік між частинами модулів, використовуючи бібліотеки що надає TenserFlow, які були написані на мові C++ та являють собою двійкові файли з підвищеною продуктивністю та за допомогою абстракцій з'єднує їх в одне ціле .

2.3.1 Keras

З Бібліотека Keras використовується для глибокого навчання та написана на мові програмування Python. Вона надає можливість пришвидшено та більше легко будувати нейромережеві моделі, використовуючи мінімум коду, що робить можливим швидку побудову прототипі. Також позитивню стороною бібліотеки є зручний інтерфейс та досягнута за рахунок використання низькорівневого API на C++ надійні

обчислювальні потужності, які використовують обчислювальний граф в якості бекенду. Keras пропонує перечень API, які ви можете використовувати для визначення нейронних мереж TensorFlow шляхом вказівки потрібних атрибутів, функцій і шарів:

- Послідовний (Sequential) API - це список шарів, що обмежений стеками шарів, через що модель має одну вхід та один вихід. Він дозволяє створювати модель шар за шаром підвищуючи точність та можливості моделі. Використовується для побудови класифікаторів та регресивних моделей. Модель йде від вхідного шару до прихованих шарів і згодом йде до вихідного шару, тобто один шар послідовно веде до іншого[7] .

- Функціональний API, який є повнофункціональним API, що підтримує довільні архітектури моделей. Він більш гнучкий і складний, ніж послідовний API.

- Model Subclassing, який дозволяє реалізувати все з нуля. Підходить для досліджень і дуже складних випадків використання, але рідко використовується на практиці.

Для визначення нейронної мережі за допомогою Sequential API в бібліотеці Keras потрібно завантажити у файл виконання програми з бібліотеки керас Sequential та необхідні шари таким чином:

```
from keras.models import Sequential
from keras.layers import Dense, Activation
```

Рис. 2.2. Завантаження бібліотек

Наступним кроком потрібно створити моделі використовуючи Sequential Api наступним чином:

```
model = Sequential()
```

Рис. 2.3. Створення моделі

Останнім кроком, потрібно додати обрані шари до нашої моделі та встановити їх параметри:

```
model.add(Dense(32, input_dim=784))
model.add(Activation('relu'))
```

Рис. 2.4. Додавання шарів

2.4 Навчання нейронних мереж.

Нейронні мережі потрібно навчити, щоб вони почали функціонувати та навчатися самостійно. Потім вони можуть вчитися на результатах отримуваної інформації, але це має з чогось початися. Існує кілька процесів, які можна використовувати, щоб допомогти нейронним мережам почати навчання.

Навчання. Нейронні мережі, які навчаються, отримують випадкові числа або ваги для початку навчання. Вони навчаються або під наглядом, або без нагляду для навчання з боку розробника або під наглядом ще однієї нейромережі. Контрольоване навчання передбачає механізм, який дає мережі оцінку або виправлення. Навчання без нагляду змушує мережу працювати, щоб визначити вхідні дані без сторонньої допомоги. Більшість нейронних мереж використовують контрольоване навчання, щоб допомогти йому швидше навчатися.

Трансфер навчання. Трансферне навчання — це техніка, яка включає в себе вирішення завдання нейронною мережею подібної задачі, яку потім можна повторно використовувати повністю або частково для прискорення навчання та підвищення ефективності розв’язання проблеми, що цікавить. **[8]**

Вилучення ознак. Вилучення функцій передбачає отримання всіх даних, які надсилаються на вхід, перевірку та видалення будь-яких зайвих даних і об’єднання їх у більш керовані сегменти. Це скорочує пам’ять і

обчислювальну потужність, необхідну для виконання проблеми через нейронну мережу, надаючи мережі лише абсолютно необхідну інформацію.

2.4.1 Розповсюдження, втрати та епохи (Backpropagation, Loss and Epochs)

Кожен нейрон в нейронній мережі приймає вхідні значення, помножені на вагу, щоб відобразити силу цього зв'язку. Розповсюдження знаходить правильні ваги, які слід застосовувати до вузлів нейронної мережі, порівнюючи поточні виходи мережі з бажаними, або правильними, виходами.

Різниця між бажаним виходом і поточним виходом обчислюється за допомогою функції втрат. Кажучі інакше, функція втрат демонструє нам, наскільки точно наша нейронна мережа робить прогнози щодо заданих навчальних даних на вхідному шарі.

Backpropagation в нейронній мережі - це скорочена форма для зворотного поширення помилок. Це стандартний метод навчання штучних нейронних мереж. Цей метод допомагає обчислити градієнт функції втрат по відношенню до всіх ваг в мережі.[7]

Функцією втрат (помилок) яку використовують для нейромережвого класифікатора, пов'язана з класичною статистичною лінійною регресією. Ця функція втрат являє собою середнє квадратичне значення різниці між правильними значеннями які ми отримаємо на виході з моделі і розрахунковими значеннями, які залежать від нахилу та інтервалу між цими даними. Тобто її принципи побудовані на різниці між вірними значаннями на виході з даними які були розраховані.

Існує три загально використовувані функції втрат для пободови нейронних мереж: середня квадратична помилка, перехресна ентропійна помилка та бінарна перехресна ентропійна помилка.

2.4.2 Функція вартості або втрат

Алгоритм коригує кожну вагу, щоб мінімізувати різницю між обчисленим значенням і правильним значенням. Термін "зворотне поширення" походить від того, що алгоритм повертається назад і коригує ваги та упередження після обчислення відповіді. Чим менші втрати для мережі, тим точнішою вона стає.

Процес навчання, таким чином, може бути кількісно визначений як мінімізація виходу функції втрат. Кожен цикл прямого поширення і корекції зворотного поширення для зменшення втрат називається епохою. Зворотне розповсюдження - це пошук найкращих вхідних ваг та зміщень для отримання більш точного результату або "мінімізації втрат". Потребує значних ресурсів для обчислювання. Лише в останні десятиліття доступна обчислювальна потужність дійшла тієї межі, щоб зробити цей процес практичним для широкого використання.

2.4.3 Типи нейронних мереж

Більшість моделей глибокого навчання використовують один з цих трьох типів нейронних мереж в якості своєї основи: Штучна нейронна мережа (ШНМ), Згорткова нейронна мережа (ЗНМ) або Рекурентна нейронна мережа (РНМ).

Штучна нейронна мережа або мережа прямого поширення складається зі штучних нейронів, згрупованих на різних рівнях, відомих як шари.

Нейронна мережа з одним шаром здатна робити лише приблизні прогнози, тому додаткові приховані шари допомагають оптимізувати точність результатів, отриманих системою. По мірі того, як мережа навчається на інформації, вона модифікує свої нейрони до тих пір, поки не зможе зробити правильні висновки для досягнення заздалегідь встановленої мети.[1]

У структурованих штучних нейронних мережах застосовуються 3 шари:

- Вхідний шар

Складається з нейронів, які асимілюють вхідні дані, наприклад, зображення або таблицю даних. Складається з реальних даних, які живлять нейронну мережу.

- Прихований шар

У нейронній мережі може бути один або кілька прихованих шарів. Чим більше інформації вона має, тим складніший аналіз вона виконує. Наприклад, цей шар може визначати вміст зображення, такий як його колір, об'єкти, літери та інше. Крім того, він може створювати сотні комбінацій для обробки інформації та виконання необхідних обчислень.

- Вихідний шар

Це шар, який приймає рішення, надає висновки або виконує необхідну діяльність в залежності від даних, наданих прихованим шаром.

Рекурентні нейронні мережі (RNN) використовуються в додатках розпізнавання природної мови і мови шляхом обробки послідовних даних. Цей механізм наділяє мережу пам'яттю. Вони можуть обробляти різні типи даних в різні моменти часу.[16]

Рекурентні нейронні мережі можуть обробляти вхід і вихід послідовності незалежно від її розміру, враховуючи кореляцію між різними елементами цієї послідовності.

Згорткові нейронні мережі (ЗНМ) використовуються в програмах розпізнавання зображень і комп'ютерного зору. Вони можуть вловлювати просторові характеристики зображення. Тобто, вони аналізують розташування пікселів на зображенні та їх відношення один до одного. Вони використовуються для виділення найбільш важливих характеристик зображення. [12]

Згорткові нейронні мережі мають спільні параметри, які застосовуються до різних ділянок вхідних даних для створення карти ознак.

Ця карта дозволяє класифікувати різні повторювані ознаки на зображенні і згодом ідентифікувати їх автоматично.

Мережа використовує випадково визначені змінні для вхідних даних на початку і модифікує їх підвищуючи або понижаючи результат в процесі навчання. Після завершення навчання мережа використовує ці модифіковані змінні для прогнозування та перевірки результату в процесах тестування та валідації.

Згорткові нейронні мережі досягли значних успіхів в задачі класифікації зображень, оскільки визначена сутність CNN відповідає розподілу точок даних на зображенні. В результаті, багато задач обробки зображень адаптують штучні нейронні мережі для автоматичного виділення ознак. CNN часто використовується для сегментації зображень, а також для обробки медичних зображень. [11] Нейронна мережа також вимагає більше часу для навчання у порівнянні з іншими підходами до навчання під наглядом та без нагляду.

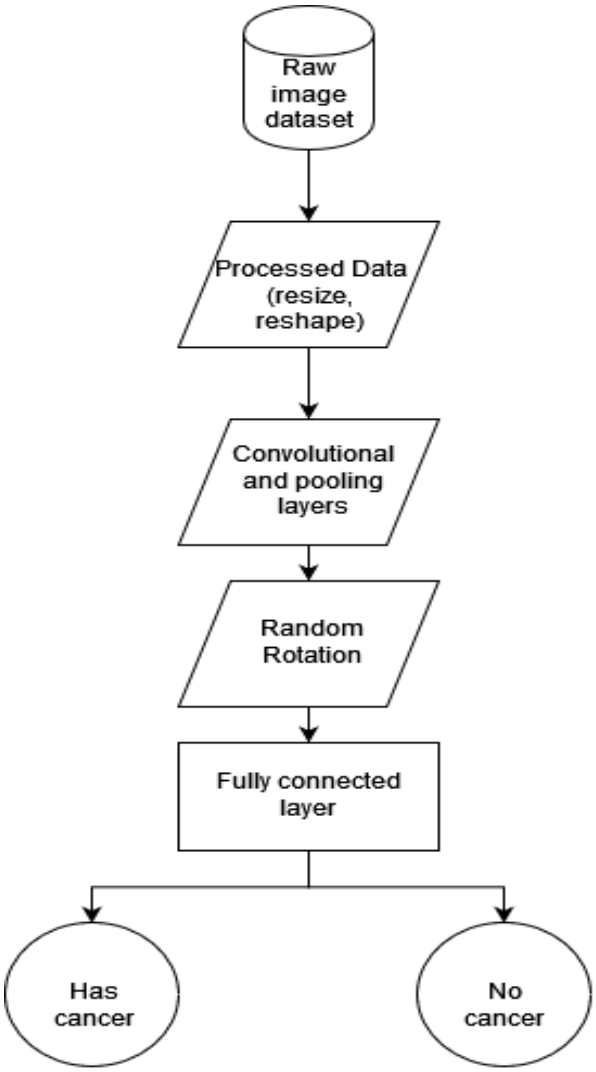

Рис. 2.5. Архітектура згорткової мережі

Для створення нашої моделі потрібно обрати архітектуру моделі. Оскільки CNN використовується для розпізнавання зображень, аналізуючи пікселі для виділення найбільш важливих характеристик і створення карти ознак на основі якої класифікує повторювальні ознаки і ідентифікує їх досягаючи високої точності у класифікації. Незважаючи на те що згорткова мережа вимагає значної кількості даних для навчання, позитивні сторони мережі переважують їх. Тому для створення моделі обрано за основу згорткову нейронну мережу (CNN).

2.4.4 Відмінності між штучним інтелектом, машинним навчанням та глибоким навчанням

Основні відмінності між штучним інтелектом, машинним навчанням і глибоким навчанням полягають в їхньому обсязі і спрямованості. В той час як штучний інтелект охоплює багато галузей інформатики, таких як хмарні обчислення і інтернет речей, машинне навчання і глибоке навчання більше пов'язані з моделями, які дозволяють машинам навчатися, як випливає з їхніх назв.

Короткий огляд цих трьох напрямків:

- Штучний інтелект: Область комп'ютерних наук, яка вивчає, як створювати комп'ютерні програми зі здатністю міркувати, як людина, вирішуючи проблеми швидше і ефективніше, ніж люди. ШІ дозволяє технологічним системам сприймати своє оточення, співвідноситися з ним, вирішувати проблеми і діяти з певною метою.[1]
- Машинне навчання: Підрозділ ШІ, присвячений створенню алгоритмів, які дозволяють системам навчатися без необхідності програмування. Робочий процес цього інструменту починається з ручного вилучення відповідних ознак для створення моделі, яка може класифікувати або обробляти таку інформацію.
- Глибоке навчання: Це підмножина машинного навчання, орієнтована на створення штучних нейронних мереж, які є системами, що імітують людський мозок шляхом адаптації та навчання на великих обсягах даних. На відміну від машинного навчання, цей інструмент витягує ознаки безпосередньо з баз даних без ручних міток.

2.4 .5 Методи покращення моделей Deep Learning

Існує кілька методів досягнення більшої надійності та ефективності моделей Deep Learning. Ось найважливіші з них:

- Зменшення швидкості навчання

Це регульований параметр, який дозволяє контролювати процес навчання моделі, цей метод визначає систему і встановлює умови для її правильної роботи до того, як вона навчиться і ідентифікує інформацію.

Його мета - оцінити і контролювати зміни, яких зазнає модель у відповідь на оцінену помилку кожного разу, коли вона модифікується.

- Навчання з перенесенням

Цей метод полягає в уточненні моделі, яка вже була навчена на даних. Після того, як ця дія виконана, можна робити нові дії з більш конкретними можливостями категоризації, ніж раніше. Його основна перевага полягає в тому, що метод потребує меншу кількість даних, ніж інші моделі, що скорочує час обчислень.

- Відсіювання

Ця модель намагається вирішити проблему перебору в мережах з великою кількістю параметрів. Крім того, вона випадковим чином усуває зв'язки нейронної мережі під час навчання.

Було показано, що цей метод покращує продуктивність нейронних мереж в задачах керованого навчання, таких як розпізнавання мови і класифікація документів.

2.4.6 Основні обмеження і виклики глибокого навчання.

Технологія вимагає потужного апаратного забезпечення, щоб мати можливість обробляти мільйони даних.

Ці технології зможуть робити речі самостійно, але дії і рішення, які вони приймають, повинні часто контролюватися людиною для забезпечення належного функціонування та коректування роботи.

Вона навчається через спостереження. Отже, якщо надані дані не відтворюють в достатній мірі необхідну класифікацію, вона не буде розуміти всебічно

Він дорогий через необхідні обчислювальні блоки та споживання великої кількості енергії.[1]

2.4.7 Згорткова нейронна мережа

У машинному навчанні класифікатор присвоює мітку класу точці даних. Наприклад, класифікатор зображень створює мітку класу для об’єктів що існують на зображенні. Згорткова нейронна мережа, або скорочено CNN, є типом класифікатора, який ідеально підходить до цього завданням через особливості її роботи.

Тензор можна розглядати як n-вимірну матрицю. У ШНМ тензори будуть 3-вимірними, за винятком вихідного шару.

Нейрон можна розглядати як функцію, яка отримує декілька входів і видає один вихід. Виходи нейронів представлені у вигляді карт активації.

Шар - це просто сукупність нейронів з однаковими операціями, включаючи однакові гіперпараметри.

Ваги та зміщення ядра, хоча і є унікальними для кожного нейрона, налаштовуються на етапі навчання і дозволяють класифікатору адаптуватися до задачі та наданого набору даних. [9]

ШНМ передає диференційовану функцію оцінки, яка представлена у вигляді оцінок класів у візуалізації на вихідному шарі. ШНМ використовують спеціальний тип шару, названий згортковим шаром, що робить їх добре пристосованими для навчання на основі зображень і даних, подібних до зображень. Що стосується даних зображень, то ШНМ можуть використовуватися для багатьох різних завдань комп'ютерного зору, таких як обробка зображень, наприклад маски снепчату, класифікація, сегментація і виявлення об'єктів.

2.4.7 Шари мережі

Вхідним шаром називають вхідні данні в згорткову мережу, наприклад зображення, метрики, тощо.

Згорткові шари

Згорткові шари є основою CNN, оскільки вони містять вивчені ядра (ваги), які виділяють ознаки, що відрізняють різні зображення одне від одного - це те, що нам потрібно для класифікації. Взаємодіючи зі згортковим шаром, між попередніми шарами та згортковим шаром утворюються зв’язки.

Кожен зв'язок являє собою унікальне ядро, яке використовується для операції згортки, щоб отримати вихідні дані або карту активації поточного згорткового нейрона. Зв'язок між згортковим шаром і попереднім шаром є проектним рішенням при побудові мережевої архітектури, яке впливатиме на кількість ядер на згортковий шар.

Згортковий нейрон виконує поелементний точковий добуток з унікальним ядром і виходом відповідного нейрона попереднього шару. Це дасть стільки проміжних результатів, скільки є унікальних ядер. Згортковий нейрон є результатом всіх проміжних результатів, об'єднаних разом з вивченим зміщенням.[15]

Розмір цих ядер є гіперпараметром, який задається розробниками архітектури мережі. Для того, щоб отримати вихід згорткового нейрона (карту активації), ми повинні виконати поелементний точковий добуток з виходом попереднього шару і унікальним ядром, вивченим мережею. Операція точкового добутку гіперпараметр, який розробник архітектури мережі може налаштувати, щоб краще відповідати своєму набору даних. Наприклад використовує крок 1 що означає, що ядро зсувається на 1 піксель за точковий добуток. Ця операція використовується для усіх наявних ядер та надає проміжні результати.

Потім виконується поелементне підсумовування, що містить всі проміжні результати разом зі зміщенням, яке засвоїла мережа. Після цього

отриманий двовимірний тензор буде картою активації. Ця ж операція повинна бути застосована для отримання карти активації кожного нейрона.

2.4.8 Гіперпараметр

- Ми використовуємо заповнення у випадках коли за межі карти активації виходить ядро. Використовується для покращення продуктивності, що досягається шляхом зберігання данних на межах карт активації. Розробленно багато методів заповнення. Найпопулярнішим підходом є нульове заповнення через його продуктивність, простоту та обчислювальну ефективність. Цей метод передбачає додавання нулів симетрично по краях вхідних даних. Він прийнятий багатьма високопродуктивними CNN, такими як AlexNet.

- Розмір ядра або розмір фільтра. Цей гіперпараметр має великий вплив на навчання класифікації зображень. Наприклад, малі розміри ядра здатні виділити з вхідних даних більшу кількість інформації, що представляє собою локальні ознаки об’єкту.

Чим менший розмір фільтра тим менше зменшення розмірів шарів, що дозволяє створити глибшу архітектуру мережі. Також це працює в іншу сторону великий розмір ядра витягує менше інформації, що призводить до швидшого зменшення розмірів шарів, що часто призводить до гіршої продуктивності нейромережі. Великі ядра показують кращі результати при використанні їх для вилучення функцій, які мають більший розмір шару.

Вибір підходходящого розміру ядра залежить від особливостей поставленого завдання і набору даних, але, як правило, менші розміри ядра призводять до кращої продуктивності для завдання класифікації зображень, тому що дизайнер архітектури може складати все більше і більше шарів разом, щоб вивчити все більше і більше складних ознак.

- Крок вказує на те, на скільки пікселів ядро має бути зсунуте за один раз. Наприклад, Tiny VGG використовує крок 1 для своїх шарів згортки, що

означає, що точковий добуток виконується на вікні вхідних даних 3x3 для отримання вихідного значення, а потім зсувається вправо на один піксель для кожної наступної операції. Вплив кроку на CNN подібний до впливу розміру ядра. Зі зменшенням кроку, більше ознак вивчається, тому що витягується більше даних, що також призводить до більших вихідних шарів. І навпаки, зі збільшенням кроку, це призводить до більш обмеженого вилучення ознак і менших розмірів вихідного шару. Одним з обов'язків проектувальника архітектури є забезпечення симетричного ковзання ядра по входу при реалізації CNN.

2.4.9 Вибір класифікації моделі

Для створення нашої моделі потрібно вибрати класифікацію моделі, існує кілька типів алгоритмів класифікації, які можна використати в залежності від набору даних. Розглянемо п’ять найпоширеніших алгоритмів класифікації в машинному навчанні.

Формула Баєс полягає в обчисленні шансу відношення точки вхідних даних до певної категорії. Використовується в аналізі тексту для класифікації слів або фраз як таких, що належать до заданої класифікації. Дає найкращій результат у порівнянні з іншими алгоритмами класифікації, особливо в таких додатках як фільтри спаму та аналіз настроїв, що включають дані у вигляді тексту. [11]

Логістична регресія – розрахунок, що прогнозує шансову можливість результату з двома можливими варіантами: щось або відбувається, або ні. Найбільш ефективний з категоріями наприклад істина чи хибність. Це працює таким чином що незалежні змінні аналізуються для визначення бінарного результату, а результати співставляються з однією з двох категорій. Незалежні змінні можуть бути категоріальними або числовими, але залежна змінна завжди є категоріальною.

K-найближчих сусіди (k-NN) – це алгоритм розпізнавання образів, який використовує навчальні набори даних для пошуку k найближчих родичів у майбутніх прикладах. Коли k-NN використовується в класифікації, обчислення зводяться до того щоб помістити вхідний сигнал в категорію найближчого сусіда. Якщо k = 1, то він буде поміщений в клас, найближчий до 1. Тобто K класифікується дивлячись на найближчу точку даних і якщо більшість сусідів належать до даних умовної групи А алгоритм класифікує данні як А. Використовується для розпізнавання тексту написаного від руки, розпізнавання зображень за категорією. K-NN найбільш корисний коли використовується для створення моделі з високою точністю але низькою читабельністю.[14]

Дерево рішень – це алгоритм керованого навчання, який ідеально підходить для проблем класифікації, оскільки він здатний впорядковувати класи на точному рівні. Працює подібно до роботи блок-схеми, розділяючи точки даних на дві схожі категорії одночасно від основи дерева до гілок та наступним кроком до «листя», де категорії стають схожими з більшою точністю. Це створює категорії всередині категорій, що дозволяє органічно класифікувати дані з обмеженим людським наглядом.

Машина опорних векторів (SVM) використовує алгоритми для навчання та класифікації даних у межах ступенів полярності. Алгоритм розподіляє дані з класифікацією на умовні мітки а результат передбачення залежить від відстані між мітками. Для максимізації навчання, найкращою гіперплощиною є та, яка має найбільшу відстань між кожною міткою. З ростом складності наборів даних зростає можовість коли не вдастся провести єдину лінію для класифікації даних. Використовуючи SVM, чим складніші дані, тим точнішим буде передбачення. [13]

Таблица 2.2

Переваги класифікацій

Назва класифікації	Перевага класифікації
Формула Баєс	Дає найкращий результат за будь-які інші алгоритми при роботі з текстом.
Логістична регресія	Має перевагу в класифікації бінарної кількості класів.
K-найближчих сусіди	Досягає значних результатів у додатках які вимагають високої точності але не потребують зрозумілої для людини моделі
Дерево рішень	Має перевагу в класифікації при значній кількості різних класів.
Машина опорних векторів	Зі збільшенням складності даних, збільшується точність передбачень. Можливість працювати у трьохвимірній площині.

Оскільки наш набор даних обмежений двома категоріями та складається з зображень, то найбільш підходящою класифікацією для моделі є логістична регресія, адже вона використовується для того щоб проаналізувати змінні, що надходять до моделі і на основі результату аналізу визначають до якої категорії відносяться змінні та має перевагу у класифікації з бінарним результатом.

2.4.10 Функція активації нейронної мережі

Функція активації виносить рішення щодо необхіності активувати нейрон. Вона вирішуватиме на скільки нейрон на вході в мережу сильно буде впливати на прогнозування, реалізуючи це через прості математичні операції.

Основна задача функції полягає у отриманні вихіднинх метрик з навчального набору

Основною ролью функції є перетворення входу вузла у вихідне знічення, що використовуються у настопному прихованому шарі.

Функції активації - функції, які здійснюють нелінійне перетворення вхідного сигналу і роблять його придатним для виконання більш складних завдань. Ми розглянули основні функції активації, ці функції активації використовуються з однією і тією ж метою, але в різних умовах.

Метою функції активації є додавання нелінійності до нейронної мережі. Це досягається за рахунок того що функції на кожному з шарів моделі впроваджують ще один крок під час прямого поширення. Якщо не використовувати активаційної функції для мережі, це призведе до того що нейрони будуть змінюватися лише під дією лінійного перетворення наданих даних використовуючи ваги та зсув. Причиною цьому є те що у випадку відсутності функції активації всі шари поводяться однаково незалежно від кількості шарів. В такому випадку мережа навчається швидше та простіше але навчити таку модель виконувати комплексні задачі неможливо через те що в результаті ми отримаємо модель з лінійною моделю.

Нейронні мережі поширені в сучасних технологіях. Найефективніші штучні нейронні мережі складаються з величезної кількості шарів, які здатні вивчати все більше і більше функцій. Частково причина, чому ці новаторські CNN здатні досягти такої величезної точності, полягає в їх нелінійності. ReLU застосовує нелінійність в моделі.

Нелінійність необхідна для отримання нелінійних границь рішень, так що вихід не може бути записаний як лінійна комбінація входів. Якби нелінійна функція активації була відсутня, глибокі архітектури ШНМ перетворилися б в один еквівалентний згортковий шар, який не працював би майже так само добре.

Функція активації ReLU спеціально використовується як нелінійна функція активації, на відміну від інших нелінійних функцій, таких як

сигмоїд, оскільки емпірично було помічено, що ШНМ, які використовують ReLU, швидше навчаються, ніж їхні аналоги. [10]

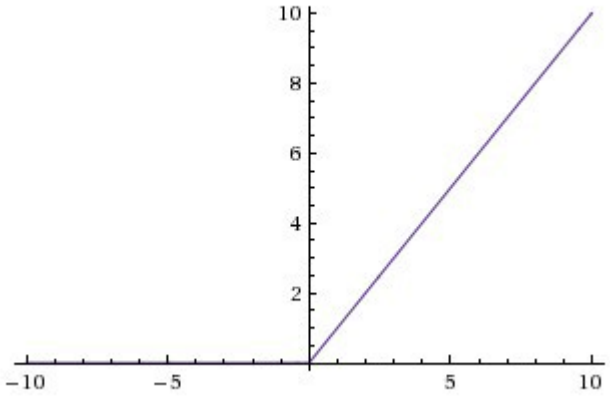

Рис. 2.6. Графік функції активації ReLU, який не враховує всі негативні дані

Ця функція активації застосовується поелементно до кожного значення з вхідного тензора та виконується після виконання кожного з шарів згортки в мережі. При використанні Функція випрямленої лінійної активації (ReLU) до будьякого значеня що більше за нуль у результаті буде отримано це значення.

2.4.11 Використання активаційних функцій

Без функції активації вага і зсув мали б тільки лінійне перетворення, або нейронна мережа була б просто лінійною регресійною моделлю, лінійне рівняння - це поліном тільки одного ступеня, який простий у вирішенні, але обмежений з точки зору здатності вирішувати складні завдання або поліноми більш високих ступенів.

На противагу цьому, додавання функції активації до нейронної мережі виконує нелінійне перетворення вхідних даних і робить її здатною вирішувати складні завдання, такі як мовні переклади та класифікація зображень. [2]

Активаційні функції є диференційованими, завдяки чому вони можуть реалізовувати зворотні розповсюдження, оптимізувати стратегію при виконанні зворотних розповсюджень для вимірювання градієнтних функцій втрат в нейронних мережах.

Типи активаційних функцій

Всі основні нейронні мережі можна поділити на три основні частини:

A. Бінарна функція кроку

B. Лінійна функція

C. Нелінійна функція активації

2.4.12 Згладжування (Flatten Layer)

Цей шар перетворює тривимірний шар в мережі в одновимірний вектор, щоб відповідати входу повністю з'єднаного шару для класифікації. Наприклад, тензор з розмірністю 5x5x2 буде перетворений у вектор розміром 50. Попередні згорткові шари мережі виділяли ознаки з вхідного зображення, але тепер настав час класифікувати ознаки. Ми використовуємо функцію softmax для класифікації цих ознак, яка вимагає одновимірного входу. Ось чому необхідний шар згладжування.

2.5 Проблеми навчання глибоких нейронних мереж

Існує дві проблеми, з якими ви можете зіткнутися при навчанні глибоких нейронних мереж.

- Зникаючі градієнти (Vanishing Gradients)

Подібно до сигмоїдної функції, певні функції активації стискають великий вхідний простір у невеликий вихідний простір між 0 і 1. Тому велика зміна на вході сигмоїдної функції спричинить невелику зміну на виході. Отже, похідна стає малою. Для неглибоких мереж з декількома шарами, які використовують ці активації, це не є великою проблемою.

Однак, коли використовується більше шарів, це може призвести до того, що градієнт буде занадто малим для ефективної роботи навчання. [3]

- Вибухові градієнти (Exploding Gradients)

Вибухові градієнти - це проблеми, коли накопичуються значні градієнти помилок, що призводять до дуже великих оновлень вагових коефіцієнтів моделі нейронної мережі під час навчання.

Нестабільна мережа може призвести до нестабільної роботи, коли є градієнти, що вибухають, і навчання не може бути завершене.

Значення вагових коефіцієнтів також можуть стати настільки великими, що можуть переповнитися і призвести до того, що називається NaN-значеннями.

2.5.1 Запобігання перенавчання

Як і у випадку будь-якої моделі машинного навчання, ключовою проблемою під час навчання згорткової нейронної мережі є перенавчання: модель настільки налаштована на специфіку навчальних даних, що її неможливо узагальнити на нові приклади. Два методи запобігання перенавчання при побудові CNN:

- Збільшення даних : штучне збільшення різноманітності та кількості навчальних прикладів шляхом виконання випадкових перетворень наявних зображень для створення набору нових варіантів .Збільшення даних особливо корисне, коли вихідний набір навчальних даних відносно невеликий.
- Регуляризація відсіву : випадкове видалення одиниць із нейронної мережі під час кроку градієнта навчання.

2.5.2 Відсіювання (Dropout)

Загальна ідея відсіювання полягає в тому, що якщо ви додасте його до моделі, то вона буде випадковим чином ігнорувати деяку підмножину вершин в даному шарі під час навчання з шансом p, тобто на кожному етапі навчання окремі вузли або викидаються з мережі з ймовірністю 1- p, або залишаються з ймовірністю p, так що залишається скорочена мережа, вхідні та вихідні ребра до вузла, що вибув, також видаляються. Звідси і назва "відсів".[3] Це дозволить запобігти участі цих відсіяних вузлів у створенні прогнозу на основі даних. Цей метод також може допомогти нашій моделі краще узагальнювати дані, яких вона раніше не бачила.

2.5.3 Перенавчання

Це поширена помилка в алгоритмах глибокого навчання, коли модель намагається повністю відповідати навчальним даним і в кінцевому підсумку запам'ятовує шаблони даних, а також шум і випадкові коливання.

Ці моделі не здатні узагальнювати і добре працювати в разі невидимих сценаріїв даних, що суперечить меті моделі.

Високе значення розсіяння вихідних значень результатів роботи моделі є індикатором проблеми надмірного приёёстосування.[5]

Час навчання моделі або її архітектурна складність можуть призвести до того, що модель буде надмірно пристосована. Якщо модель тренується занадто довго на навчальних даних або є занадто складною, вона засвоює шум або нерелевантну інформацію в наборі даних.

Рис.2.7. Ознаки перенавчання

Ключові визначення перенавчання:

Упередженість: Упередженість вимірює різницю між прогнозом моделі та цільовим значенням. Якщо модель надмірно спрощена, то прогнозоване значення буде далеким від істини, що призведе до більшої похибки.

Дисперсія: Дисперсія - це міра неузгодженості різних прогнозів на різних наборах даних. Якщо продуктивність моделі перевіряється на різних наборах даних, то чим ближче прогноз, тим менша дисперсія. Вища дисперсія є ознакою надмірної підгонки, при якій модель втрачає здатність до узагальнення.

Компроміс між зміщенням та дисперсією: Очікується, що проста лінійна модель матиме високе зміщення та низьку дисперсію через меншу складність моделі та меншу кількість параметрів, що піддаються навчанню. З іншого боку, складні нелінійні моделі, як правило, демонструють протилежну поведінку. В ідеальному сценарії модель повинна мати оптимальний баланс зміщення та дисперсії.

Узагальнення моделі: Узагальнення моделі означає, наскільки добре модель навчена витягувати корисні шаблони даних і класифікувати невидимі зразки даних.

Відбір ознак: Передбачає вибір підмножини ознак з усіх вилучених ознак, які роблять найбільший внесок у продуктивність моделі. Включення всіх ознак без необхідності збільшує складність моделі, а надлишкові ознаки можуть значно збільшити час навчання.

Перенавчання відбувається тоді, коли:

- Дані, що використовуються для навчання, не очищені і містять сміттєві значення. Модель фіксує шум у навчальних даних і не може узагальнити навчання моделі.
- Модель має високу дисперсію.
- Недостатня кількість даних для навчання моделі.
- Архітектура моделі має декілька нейронних шарів, складених разом. Глибокі нейронні мережі є складними і потребують значного часу для навчання, а також часто призводять до перенавчання навчальної вибірки.

2.5.4 Операція максимального об’єднання (max pooling)

Ця операція обчислює найбільше значення досягнутих числе у визначеній дільці карт, що представляє собою визначену матрицю, і використовує ці значення створення нової карти об’єктів з пониженою вибіркою. Основним застосуванням є використання в архітектурі загорткових нейронних мереж (CNN) після шару згортки.

Основною ідеєю шару є накопичення особливості з карт, створених шляхом згортки фільтра над зображенням. Його функція полягає у поступовому зменшенні просторового розміру представлення, що у свою чергу зменшує кількість параметрів та обчислень у мережі. Найпоширенішою формою об'єднання є максимальне об'єднання. Інші форми об'єднання: середнє, загальне. [6]

Максимальне об'єднання допомагає зменшити надмірне пристосування, надаючи абстраговану форму представлення. Також зменшує обчислювальні витрати за рахунок зменшення кількості параметрів, що вивчаються мережею. Максимальне об’єднання додає певну інваріантність перекладу, це означає, що переведення зображення на невелику величину не має суттєвого впливу на значення більшості об'єднаних результатів. Здійснюється шляхом застосування максимального фільтра до субрегіонів початкового представлення, що не перетинаються.

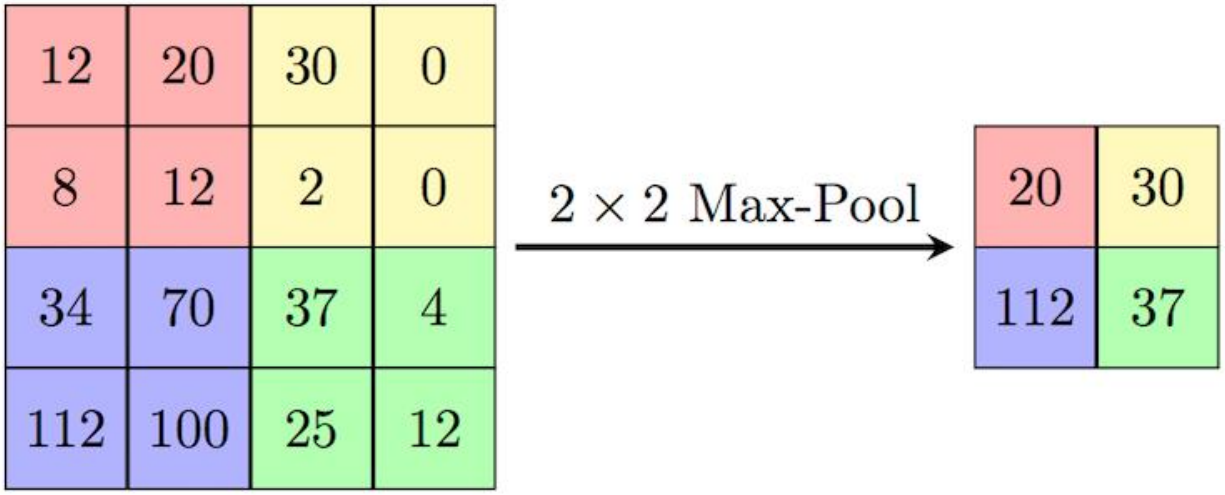

Рис. 2.8. Ілюстрація операції максимального об’єднання

2.5.5 Виявлення перенавчання моделі

Виявити надмірну пристосованість без попереднього тестування даних або аналізу графіків точності і втрат між валідаційним та тренувальним наборами технічно неможливо. Їх розходження є ознаками перенавчання. Одним з провідних індикаторів перенастроювання моделі є її нездатність узагальнювати набори даних. Перший спосіб що приходить на думку для виявлення перенавчання нашої моделі це поділити наші данні на під категорії. Це робиться для того, щоб ми могли дослідити продуктивність

моделі на кожному наборі даних, щоб виявити перенавчання, коли воно відбувається, і побачити, як працює процес навчання.

K-кратна перехресна перевірка - одна з найпопулярніших методик, яка зазвичай використовується для виявлення перенавчання.

При K-кратній перехресній перевірці ми розбиваємо точки даних на k однакових за розміром підмножин, які називаються "складками". Одна з розбитих підмножин виступає в якості тестової вибірки, а решта фолдів будуть навчати модель. [5]

Модель тренується на обмеженій вибірці, які не були задіяні під час навчання моделі, щоб оцінити, що модель буде працювати в цілому, коли вона буде використовуватися для прогнозування на даних.

2.5.6 **Ваги та зміщення(Weights and Bias)**

Вхідні дані, отримані з вхідного шару, перемножуються з присвоєними їм вагами. Перемножені значення додаються, щоб сформувати зважену суму. Зважена сума входів та їх відповідні ваги потім застосовуються до відповідної функції активації. Функція активації відображає вхідні дані на відповідний вихід.

Як тільки вхідна змінна подається в мережу, випадково вибране значення призначається в якості ваги цього входу. Вага кожної точки вхідних даних вказує на те, наскільки важливим є цей вхід для прогнозування результату.

Параметр зміщення, з іншого боку, дозволяє налаштувати криву функції активації таким чином, щоб забезпечити найбільш можливу точність.

ВИСНОВОК ДО РОЗДІЛУ 2

В другому розділі були розглянуті засоби, які були обрані для розробки штучного інтелекту для розпізнавання ракових пухлин. Обраними засобами для створення мережі є середовище python та бібліотека Tenserflow.

Моделью була обрана згорткова нейронна мережа. Класифікацією була обрана логістична регресія оскільки наш набор даних обмежений двома категоріями та складається з зображень.

Основними плюсами обраних засобів були полегшення розробки та значна кількість документації для проекту.

Набором даних було обрано дата сет з caggle Br35H :: Brain Tumor Detection 2020 та поділено між папками train, test та validation для подальшого навчання моделі на визначенних даних.

Засоби програмування були обрані для реалізації поставленої цілі, а саме створення нейронної мережі з високим коефіцієнтом точності передбачування.

РОЗДІЛ 3 Створення моделі та реалізація нейронної мережі

3.1 Структура датасету

У даній роботі було використано дата сет з kaggle Br35H :: Brain Tumor Detection 2020 який складається з даних зібраних з різних знімків пухлин та поділених на папки yes - в яких є пухлина та no – в яких її немає. Дані поділено на частини у відношенні 70% : 20% : 10% для тренування, валідації та тестування відповідно.

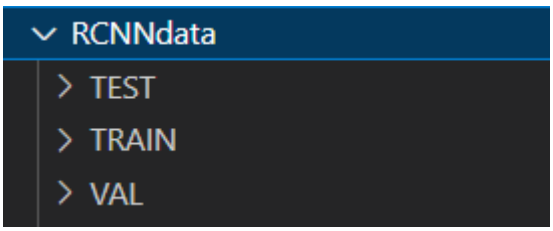

Рис. 3.1. Структура каталогу дата сету

3.2 Побудова моделі

Далі для використання даних застосуємо метод бібліотеки keras image_dataset_from_directory який повертає пакети зображень з підкаталогів (в нашому випадку каталогу yes та no) з мітками 0 і 1. У функцію ми передаємо три аргументи: set, image_size та batch_size. Set це шлях до папки з якої беремо данні що повинен містити підкаталоги, кожен з яких містить зображення для класу. Image_size це розмір до якого перетвориться зображення після зчитування приймає аргументи висоти та ширини до яких буде змінюватися зображення. Batch_size - кількість елементів, які будуть поширюватися через мережу на кожному кроці (за замовчуванням: 32). В більшості випадків мережі навчаються швидше з меншої розмірністю batch size це виникає через те що ми оновлюємо ваги після кожного кроку. Але

через це виникає проблема з тим що чим менша кількість елементів для поширення, тим менш точною буде оцінка градієнта.

```
def set_from_dir(set): return tf.keras.utils.image_dataset_from_directory(
    set,
    image_size=(img_height, img_width),
    batch_size=batch_size
)
```

Рис. 3.2. Функція зчитування

Далі приведемо дані до нормованого вигляду викликом функції normilise в яку передамо данні з попереднього кроку. Ця функція перетворює дані для вирівнювання значень даних до загальної шкали. Це робиться для того щоб дані з великими значеннями не мали надмірного впливу на змінні з малими значеннями зближаючи нейронну мережу до оптимального набору параметрів та зменшує кількість обчислювальних кроків для пошуку ефективної моделі.

```
def normalise(set): return set.map(lambda x, y: (x/255, y))
```

Рис. 3.3. Функція нормалізації

Наступним кроком використаємо keras.Sequential який групує лінійний стек шарів у tf.keras.Model. Всередині створюємо вхідний шар, приховані шари, вихідний шар та використовуємо відсів (Dropout), операція максимального об’єднання (Max pooling) та випадкове обертання. Для створення вхідного шару використовуємо Conv2d з функцією активації тангенсу. Цей шар створює ядро згортки, яке згортається з вхідними даними шару для отримання тензора виходів. Оскільки цей шар ми використовуємо для як вхідний потрібно вказати аргумент з ключовим словом input_shape та передати в нього формат для зображень на яких тренуємо модель. Dense – це шар який містить щільно з’єднані нейрони таким чином що кожен з окремих нейронів отримує вхідні дані від усіх інших нейронів.

При операції максимального об’єднання здійснюється вибірка вхідних даних вздовж їх розмірів (висоти та ширини), беручи максимальне значення у вікні вхідних даних для кожного каналу вхідних даних. Вікно зсувається на кроки вздовж кожного виміру.

Відсіювання застосовується до шару для зменшення перенавчання, відсіюючи частину вихідних ознак під час навчання моделі на кожній з епох. В аргумент приймає показник відсіву - це частка функцій, які обнуляються, в більшості випадків його встановлюють між 0,2 і 0,5. Випадкове обертання застосовується для обробки кожного зображення що попадає в модель застосовуючи до неї випадкове обертання в діапазоні що передається в аргументі factor. Випадкове обертання використовується з метою зменшення перенавчання моделі.

```
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(16, 3, padding='same', activation='tanh', use_bias=True,
                           kernel_constraint=min_max_norm(
                               min_value=-1., max_value=1.),
                           input_shape=(128, 128, 3)),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.experimental.preprocessing.RandomRotation(
        factor, fill_mode='reflect', interpolation='bilinear',
        seed=None, name=None, fill_value=0.0
    ),
    tf.keras.layers.Conv2D(32, 3, padding='same', activation='relu', use_bias=True,
                           kernel_constraint=min_max_norm(min_value=-1., max_value=1.)),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(128, 3, padding='same', activation='relu', use_bias=True,
                           kernel_constraint=min_max_norm(min_value=-1., max_value=1.)),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(975, activation='relu', use_bias=True,
                          kernel_constraint=min_max_norm(min_value=-1., max_value=1.)),
    tf.keras.layers.Dropout(.3),
    tf.keras.layers.Dense(325, activation='relu', use_bias=True,
                          kernel_constraint=min_max_norm(min_value=-1., max_value=1.)),
    tf.keras.layers.Dense(1)
])
```

Рис. 3.4. Створення Архітектури моделі бібліотекою keras

Наступним кроком налаштовуємо модель на навчання встановлюючи оптимізатор, функцію втрат та метрики які будуть оцінюватися моделлю під

час навчання та тестування. Найбільш популярним оптимізатором є Адама - це метод стохастичного градієнтного спуску, який базується на адаптивному оцінюванні моментів першого та другого порядку який приймає аргументом швидкість навчання (За замовчуванням дорівнює 0.001). Використовується для додатків з двійковою класифікацією. Функція обчислює втрату ентропії між істинними та передбачуваними мітками. Метрикою було обрано “accuracy”. Ця метрика створює дві локальні змінні для підрахування частоти з якою прогнози збігаються міткам.

```
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.0006),
    loss=tf.losses.BinaryCrossentropy(from_logits=True),
    metrics=['accuracy']
)
```

Рис. 3.5. Налаштування моделі

Для тренування використаємо метод моделі model.fit. Він тренує модель протягом встановленої кількості епох (ітерацій на наборі даних) . Також він приймає набір функцій зворотнього виклику в які ми передаємо функцію для зберігання моделі ModelCheckpoint яка приймає аргументами шлях зберігання, save_weights_only: якщо True, то будуть збережені тільки ваги моделі або вся модель для використання їх у подальшому зі збереженого стану, наступна опція save_best_only обирає чи зберігати тільки ту модель, яка досягла найкращих результатів, або зберігати модель в кінці кожної епохи незалежно від результатів. Також передаємо функцію LearningRateSheduler для зміни швидкості навчання (learning rate) після проходження певної епохи, що приймає в аргумент функцію яка визначає поточну швидкість.

```
history = model.fit(
    train_set,
    validation_data=valid_set,
    epochs=18,
    callbacks=[cp_callback, callbacks]
)
```

Рис. 3.6. Навчання моделі

Було побудовано графік метрик точності моделі після проходження кожної епохи для даних навчання та перевірки. З графіка точності можна побачити, що модель, можна було б навчити ще трохи, оскільки тенденція точності на обох наборах даних все ще зростає протягом останніх кількох епох, показуючи порівнянну майстерність на обох наборах даних. Ви також можете бачити, що модель починає розходитися після 15 епохи на навчальному наборі даних, це є ознакою що потрібно збільшити кількість засобів проти перенавчання, змінювати швидкість навчання після певної епохи або припиняти навчання на більш ранній епосі.

З графіка втрат видно, що модель має порівнянну продуктивність як на навчальному, так і на валідаційному наборах даних . Якщо ці паралельні графіки почнуть послідовно розходитися, це може бути ознакою того, що потрібно припинити навчання на більш ранній епосі, однак доки різниця між значеннями втрат та валідаційних втрат не вище за 0.2 метрики відносно нормальні.

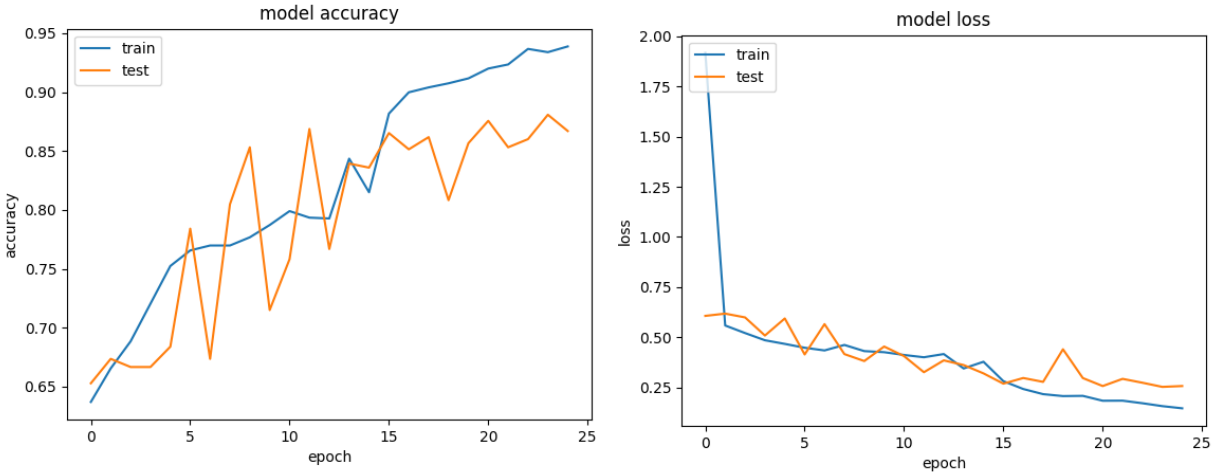

Рис. 3.7. Графік метрик

Для використання навченої нейронної мережі потрібно відтворити модель всі її шари, метрики такі як швидкість навчання, ширина і довжина зображення функції активації та кількість нейронів. Наступним кроком потрібно зкомпілювати модель встановивши швидкість навчання, обраний оптимайзер та метрику яку підраховує модель в точності як на навченій моделі. Тепер ми можемо завантажити збережену модель наступним чином.

```
model.load_weights(
    "./models/checkpoint_adamgrad_15_lr01_B/checkpoints.ckpt").expect_partial()
```

Рис. 3.8. Завантаження моделі

Наступним кроком потрібно завантажити зображення з тестового набору вказавши широту та висоту з якою навчалася модель для зображення. Перетворити зображення у масив методом img_to_array для приведення даних до виду в якому наша нейронна мережа зможе її зрозуміти. Наступним кроком використаємо np.expernd_dims Ця функція розширює масив, вставляючи нову вісь у вказаній позиції. Ця функція вимагає два параметри масив з яким працює та вісь. Та застосуємо метод моделі predict для прогнозування навченої мережі.

```
def check(img, label):
    img = image.load_img(img,
                         target_size=(img_width, img_height))
    y = image.img_to_array(img)
    y = np.expand_dims(y, axis=0)
    y_pred = model.predict(y)
    y_pred_g = [1 if y_pred >=
                0.5 else 0 for y_pred in y_pred]
    print("Predict: ", y_pred_g, "Label: ", label)
```

Рис. 3.9. Прогнозування навченою нейронною мережею

Викличемо створену функцію для прогнозування нашою мережею. Вона приймає два аргументи шлях до зображення та мітка до якої класифікації відноситься зображення має зображення пухлину чи ні (0 або 1). У результаті ми отримаємо шанс з яким це зображення має пухлину та виведену мітку поруч.

```
Predict:  [1] Label:  1
Predict:  [1] Label:  1
Predict:  [1] Label:  1
Predict:  [1] Label:  1
Predict:  [0] Label:  0
Predict:  [0] Label:  0
Predict:  [0] Label:  0
Predict:  [0] Label:  0
```

Рис. 3.10. Результат прогнозування на тестовому зображенні

3.3 Тестування для знаходження найкращої моделі

Для нашої моделі потрібно обрати найбільш підходящий для вирішення поставленної задачі оптимайзер. Тому проведемо тести існуючих оптимайзерів для відбору найкращого. Для тестування були обрані такі

оптимайзери: Adam, Nadam, Adadelta, Adagrad, Adamax, SGD. Всі моделі навчаються протягом 5 епох з одним з оптимайзерів з незмінною швидкістю навчання що становить 0.001. Розмір партій що поставляються в мережу (batch size) становить 32. Модель має шар відсіву після повнозв'язного шару з ймовірністю в 0.2 та шар випадкового обертання з відхилом у 20 градусів. Почнемо з тестування на малій виборці.

Таблица.3.1

Тестування оптимайзеру на малій виборці

Оптимайзер	Точність	Втрати
Adam	0.8204	0.5380
Nadam	0.8221	0.4025
Adadelta	0.8377	0.3382
Adagrad	0.8307	0.3813
Adamax	0.8378	0.3967
SGD	0.8221	0.3786

Найкращу точність на малій виборі мають оптимайзери Adamax та Adadelta. Найгіршу точність має оптимайзер SGD. Різниця між втратами та втратами валідації не перевищує 0.2 отже наші моделі не отримали перенавчання або недостатньої пристосованості. Виходячи з отриманих даних можна зробити висновок що точність всіх моделей з кожною епохою зростають, тому можна продовжувати їх навчати. Тепер відслідкуємо роботу оптимайзерів на виборці з 15 епох не змінюючи швидкість навчання та залишаючи ті ж самі шари які використовували для навчання моделі на малій виборці.

Таблица.3.2

Тестування оптимайзеру на виборці з 15 епох

Оптимайзер	Точність	Втрати
Adam	0.8411	0.5473
Nadam	0.8653	0.3947
Adadelta	0.8152	0.4453
Adagrad	0.8860	0.2693
Adamax	0.8687	0.3268
SGD	0.8532	0.3204

Найкращу точність на виборі з 15 епох мають оптимайзери Adamax та Adagrad. Найгіршу точність має оптимайзер adadelta. Різниця між втратами та втратами валідації найкращих модель не перевищує 0.2 отже моделі не отримали перенавчання або недостатньої пристосованості. Для подальшої вибірки використаємо найкращі оптимайзери та протестуємо їх на різній швидкості навчання протестувавши та порівнявши отримані моделі на 0.01, 0.001, та 0.0001 швидкостях. Навчатимемо 15 епох не змінюючи шарів моделі.

Таблица 3.3

Тестування на різній швидкості навчання

Оптимайзер	Швидкість навчання	Точність	Втрати
Adamax	0.01	0.8307	0.2698
Adagrad	0.01	0.9223	0.1926
Adamax	0.001	0.8687	0.3268
Adagrad	0.001	0.8860	0.2693
Adamax	0.0001	0.7789	0.3592
Adagrad	0.0001	0.8636	0.3310

Опираючись на всі попередні дослідження та рішення в рамках технічних можливостей комп’ютеру модель та для упередження перенавчання була навчена модель на 15 епохах. Оптимайзером було оброно Adagrad зі швидкістю навчання 0.01. Для зменшення перенавчання були використані дропаут з аргументом 0.2 та випадкове обертання з обертом у 20 градусів.

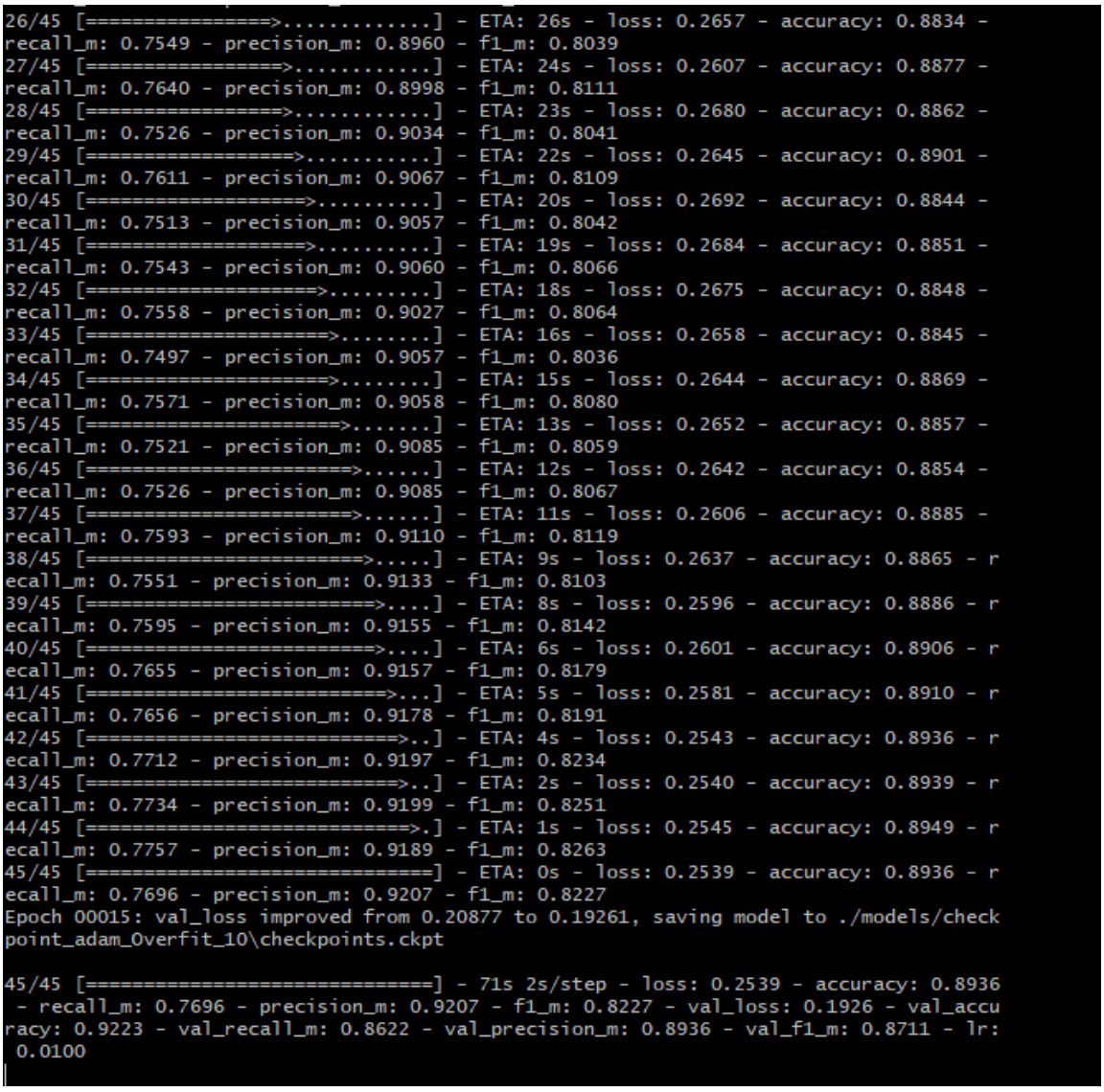

Рис. 3.11. Результуючі данні за 15 епох

Отже, бачимо, що після завершення навчання моделі на 15й епосі, досягнута точність складає 92.2% а різниця між втратою та валідаційною втратою складає 0.06 що менше за 0.2 які б означали перенавчання моделі.

Встановимо швидкість 0.01 для оптимізатора Adagrad та будемо змінювати швидкість навчання після 15 епохи на 0.001 спробуємо навчити кращу модель залишивши ті ж самі шари та засоби для уникнення перенавчання.

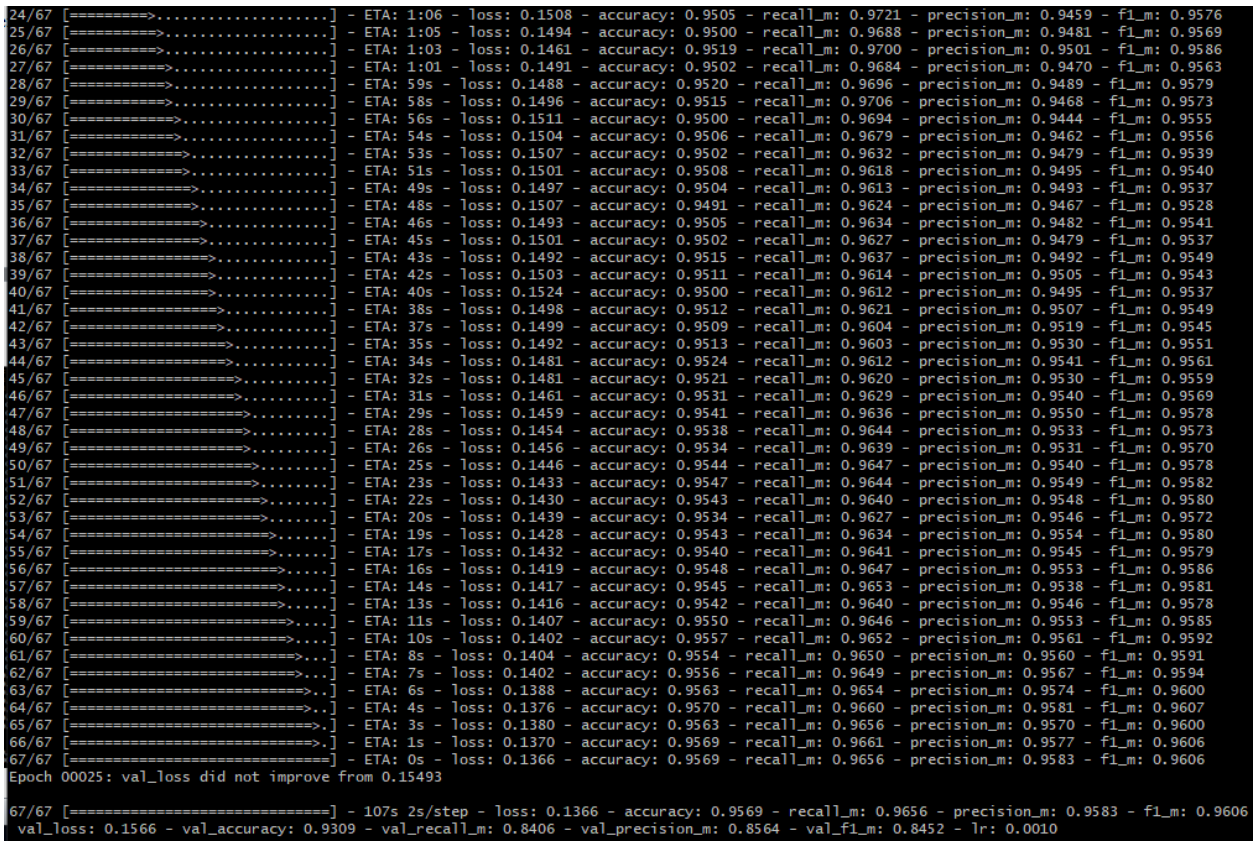

Рис. 3.12. Результуючі дані за 25 епох

У результаті, бачимо, що після завершення навчання моделі на 25й епосі, досягнута точність складає 93.09% а різниця між втратою та валідаційною втратою складає 0.02 що менше за 0.2 які б означали перенавчання моделі.

Модель на останніх епохах перебуває на приблизно одній шкалі тому слід збільшити швидкість навчання для моделі щоб досягти більшої точності.Встановимо швидкість 0.01 для оптимізатора Adagrad та збільшимо швидкість навчання після 15 епохи на 0.005 залишимо ті ж самі шари та засоби для уникнення перенавчання.

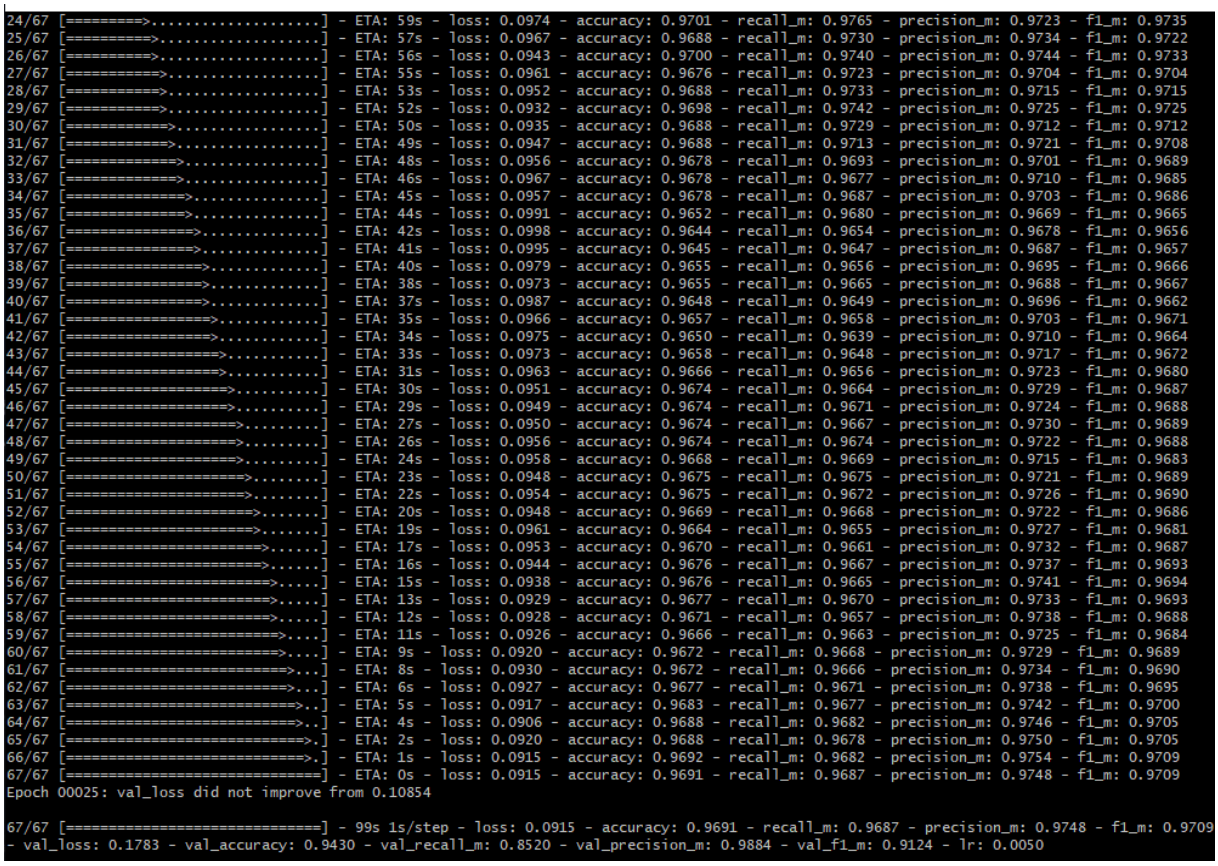

Рис. 3.13. Результуючі дані за 25 епох

У результаті, бачимо, що після завершення навчання моделі на 25й епосі, досягнута точність складає 94.3% а різниця між втратою та валідаційною втратою складає 0.08, що менше за 0.2 які б означали перенавчання моделі.

Перевіримо модель на тестових данних:

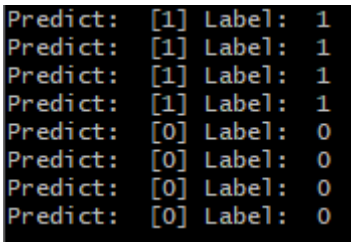

Рис. 3.14. Перевірка на тестових данних.

Перевірка пройдена успішно. Модель пройшла всі тести.

ВИСНОВОК ДО РОЗДІЛУ 3

В третьому розділі була розглянута створення моделі та реалізація нейронної мережі. Розглянуто створення структури каталогу для дата сету, розбиття даних по відповідним папкам у ефективному для навчання відношенні та приведення даних до нормованого виду в якому їх зручно використовувати у навчанні нейронної мережі.

Розглянуто метод для побудови вхідного шару, прихованих шарів, вихідного шару у моделі та метод максимального об`єднання (Maxpooling). Показано використання методів для уникнення перенавчання моделі а саме відсіву (Dropout) та випадкового обертання наданого зображення. Також розглянуто методи для налаштування, зберігання та тренування моделі з колбєк функцією яка буде змінювати швидкість навчання утвореної моделі після проходження певної епохи. Розглянуто яким чином використовувати навчену модель.

Обрано найефективніший оптимайзер для вирішення поставленної задачі шляхом тестування - Adagrad.

На основі всіх обраних архітектурних рішень було створенно згорткову нейронну мережу точність якої складає 94.3% без ознак перенавчання.

РОЗДІЛ 4.

РОЗРОБКА СТАРТАП ПРОЕКТУ

4.1 Опис проекту

Обравши за основу дипломний проект, взявши створенну згорткову мережу за основу можна реалізувати додаток та АПІ для виявлення ракових пухлин, який може бути використаний як розробниками так і користувачами при отриманні ключа.

Даний розділ буде присвячено для описання розробки стартап-проекту на основі згорткової мережі.

Таблица 4.1.

Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Додаток та АПІ для виявлення ракових пухлин	Пошук пухлин для користувача	Можливість виявити злоякісну пухлину паціенту
	Співправця з закладами освіти	Можливість перевірки поставленого діагнозу при навчанні абітурієнтів

Обрана технологія реалізації ідеї проекту: розробка програми на мові програмування Python, фреймворк для моделі TensorFlow із використанням бібліотеки keras.

4.2. Технологічний аудит ідеї проекту

Таблиця 4.2.

Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технологій
1	Додаток та АПІ для виявлення ракових пухлин	Мова програмування Python	Є в наявності	Доступні безкоштовно
2		Фреймворк TensorFlow	Є в наявності	Доступні безкоштовно
3		Бібліотека keras	Є в наявності	Доступні безкоштовно

Обрані технології для утілення в життя ідеї проекту: Мова програмування Python, фреймворк TensorFlow, бібліотека keras. Усі технології в наявності і доступні для реалізації проекту.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів. В таблиці 4.3 наведена попередня характеристика потенційного ринку для розроблюваного стартап-проекту.

Таблиця 4.3

Попередня характеристика потенційного ринку стартап-проекту

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	Більше 5
2	Динаміка ринку (якісна оцінка)	Проблема з раковими утвореннями по всьому світу постійно зростає і з цим зростає потреба у швидкому аналізі наявності пухлин
3	Наявність обмежень для входу (вказати характер обмежень)	Значна кількість конкурентів серед розробників рішень з використанням штучного інтелекту для сфери зберігання здоров’я
4	Специфічні вимоги до стандартизації та сертифікації	Специфічних вимог до стандартизації та сертифікації немає

Таблиця 4.4

Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару

Таблица 4.4(продовження)

Характеристика потенційних клієнтів стартап-проекту

1	Упереджений пошук пухлини	Заклади надання медичної допомоги	Навантаження системи	Точність надійність мережі та легкість у використанні
2	Навчання абітурієнтів навчальних закладів	Заклади освіти, абітуріенти	Кількість абітурієнтів та частота використання	Зручне користування та швидкість

Таблиця 4.5

Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Складність виходу моделі на міжнародний ринок	Всі потенційні конкуренти розташовані закордоном. Важкий вихід	Поступово розвиватися на міжнародні арені починаючи розповсюджуватися з найближчих країн.
2.	Довіра у використанні моделі	Упереджене відношення до використання моделі для додаткових кроків перед поставленням діагнозу	Зменшення попиту, необхідні додаткові стимули для довіри у використанні моделі.

Таблиця 4.6

Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Залучення інвестицій	Склавши чіткий бізнес-план, можна залучити інвесторів, що дозволить розширити штат та закупити потрібну техніку	Скорочення часу для виходу на міжнародний ринок та покращення системи
2	Розширення клієнтської бази	Реалізувати масштабну рекламну компанію щоб залучити якомога більше клієнтів	Спланувати рекламну кампанію, проводити презентації на зборах і виставках та семінари на тему згорткових мереж, в яких для демонстрації буде використано створений продукт
3	Збільшення клієнтів-підприємців	Складання договорів для подальшого співробітництво з закладами освіти	Розширення бази даних лікарів

Таблица 4.7

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкуренто- спроможною)
Складність старту	На ринку уже є досить успішні конкуренти	Якісна рекламна кампанія та покращення функціоналу
Складність входу на міжнародний ринок	Немає досвіду такого товару на ринку, значні фінансові затрати для створення початкових умов	Проаналізувати необхідну складову базу та створити найбільш сприятливих умов для клієнта

Таблиця 4.8

Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти в галузі	Постачальники	Клієнти	Товари-замінники
	фактично прямих конкурентів нема	Ai-derm	-	Заклади освіти та охорони здоров’я	Нові стартап-ідеї

Таблиці 4.8 (продовження)

Аналіз конкуренції в галузі за М. Портером

Висновки:	Можлива велика кількість різноманітних рішень у конкурентів	-	Постачальники орієнтується на потреби клієнтів.	Заклади здоров'я вже можуть співпрацювати з клієнтами	Ентузіасти можуть вийти на ринок с кращим функціоналом

Таблиця 4.9

Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Досить висока точність	Оскільки проект знаходиться на етапі розробки, можна спиратися на точність товарів-замінників, щоб при виході товару на ринок мати перевагу у точності передбачування
2	Унікальність надання послуг	Користувач має можливість отримання функцій таких послуг, які на сьогодні отримуються в основному за фізичної наявності

Таблиці 4.9 (продовження)

Обґрунтування факторів конкурентоспроможності

3	Швидке реагування на відгуки	На етапі розробки неможливо передбачити всі нюанси застосування. Завдяки забезпеченню швидкого відгуку, можна використовувати користувачів у якості бета-тестувальників, що дозволить максимально швидко адаптувати додаток під потреби реальних клієнтів

Таблиця 4.10

Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з … (наша підприємства)						
			–3	–2	–1	0	+1	+2	+3
1.	Точність класифікації	17					+		
2.	Швидкість обробки	10			+				
3.	Підтримка продукту	10				+			
4.	Вдосконалення продукту через реакцію користувачів на бету версію.	9					+		

Таблиця 4.11

SWOT- аналіз стартап-проекту

Сильні сторони: Надійність Швидке реагування на побажання клієнтів Легкість експлуатування Кращі характеристики	Слабкі сторони: Потрібні інвестиції для старту Новий метод, ще не визнаний у світі.
Можливості: Вихід на міжнародний ринок Збільшення функціоналу та покращення моделі	Загрози: Важкість виходу на міжнародний ринок Нестабільна ситуація в країні Поява нового продукту Складність знайти інвестора

Таблица 4.12

Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Активна реклама у всіх ресурсах	Висока. Потужна реклама допоможе розповсюдитися серед значної кількості користувачів	Після старту

Таблица 4.12 (продовження)

Альтернативи ринкового впровадження стартап-проекту

2	Вихід на міжнародній рівень	Середній, оскільки більша кількість конкурентів та потреба у відкриті офісів для потреб на міжнародній арені	Через півтора року
3	Участь в тендерах і укладення договорів з державними установами	Середня, складно передбачити потребу у співпраці	Через приблизно пів року, в залежності від зацікавлення продуктом державної установи

4.4. Розроблення ринкової стратегії проекту

Таблиця 4.13

Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачі в сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Державні установи	Не всі користувачі готові прийняти	Продукт необхідний уже	Конкуренція середня	Складний вхід

Таблиці 4.13(продовження)

Вибір цільових груп потенційних споживачів

2	Приватні компанії	Не всі користувачі готові прийняти	Продукт необхідний не всім	Висока конкуренція	Складний вхід
3	Навчальні та дослідницькі програми	Користувачі готові прийняти продукт	Не всі зацікавлені в продукті	Конкуренція середня	Вхід помірно-складний
Які цільові групи обрано: Потрібно працювати зі всіма цільовими групами, перевагу надати послугам, що найбільш популярні для клієнтів.					

Таблиця 4.14

Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Розвиток шляхом потужної маркетинговій компанії та демонстрації продукту на відкритих конференціях	Реклама та охоплення цільової аудиторії на конференціях допоможе знайти клієнтів	Виступ на конференціях, вільне розповсюдження для навчально-дослідницьких цілей	Прислухатися до побажань клієнтів та оперативно оновлювати продукт під їх стандарти

Таблиця 4.15

Визначення базової стратегії конкурентної поведінки

№	Чи є проект «пер-шопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні, існують аналоги	Компанії необхідно буде шукати клієнтів які найбільш зацікавленні у продукті	Через схожість функціоналу, компанія буде копіювати та намагатися покращити основні характеристики у конкурентів	Пропонувати схожий функціонал на кращих умовах, оперативно оновлювати функціонал відповідно до нових стандартів

Таблиця 4.16

Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап- проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)

Таблиця 4.16 (продовження)

Визначення стратегії позиціонування

1	Модель повинна мати високу точність та бути надійною	Залучати все більше нових клієнтів для розширення бази даних та розширятися на міжнародній арені	Оперативне виправлення недоліків та впровадження побажань	Зручна для користувачів, точний у прогнозах

4.5. Розроблення маркетингової програми стартап-проекту

Таблиця 4.17

Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Моделювання мережі високої точності архітектурі CNN	Високоточність передбачування	Можливість отримувати чітке точне передбачення
2	Часте оновлення програми	Врахування побажань клієнтів та виправлення помилок в найкоротші терміни	Створення відділу підтримки для аналізування реагування на відгуки клієнтів

Таблиця 4.17 (продовження)

Визначення ключових переваг концепції потенційного товару

3	Доступ до АПІ створенної моделі	Можливість використання програми для власних розробок	Відкритий АПІ для використання розробниками

Таблиця 4.18

Опис трьох рівнів моделі товару

<table>
<tr><td>Рівні товару</td><td colspan="2">Сутність та складові</td></tr>
<tr><td>І. Товар за задумом</td><td colspan="2">Програма для передбачення наявності пухлини у пацієнта</td></tr>
<tr><td rowspan="5">ІІ. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>Розмір</td></tr>
<tr><td>Модуль- CNN моделі</td><td>40МБ</td></tr>
<tr><td>Веб-сторінка для передбачення пухлини</td><td>50МБ</td></tr>
<tr><td>Модуль обробки запросів</td><td>30МБ</td></tr>
<tr><td colspan="2">Якість: логування збоїв та система зворотнього зв’язку для повідомлення про неналежну роботу програми</td></tr>
</table>

Таблиця 4.18 (продовження)

Опис трьох рівнів моделі товару

II. Товар у реальному виконанні	Пакування: продаж електронних ключів-ліцензій
	Марка: назва організації-розробника + назва товару
III. Товар із підкріпленням	До продажу: програмний код
	Після продажу: АПІ/електронна версія додатку з ключем
За рахунок чого потенційний товар буде захищено від копіювання: прив'язка копії програмного продукту до конкретного ПК та активація програми шляхом введення ліцензійного ключа, або шляхом надання послуг без передачі програмного продукту замовнику	

Таблиця 4.19

Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі становлення ціни на товар/послугу
1	0грн - 100грн	0грн - 100грн	>10000грн/місяць	0грн - 1000грн

Таблиця 4.20

Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Придбання ліцензії на продукт або договір про надання послуг без передачі ПО до замовника	Розміщення Ключа для доступу до АПІ на веб-сервісі	Канал збуту одноріневий (через розміщення ключів на веб-сервісах)	Вертикальна (право власності залишається у розробника ПЗ)

Таблиця 4.21

Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення

Таблиця 4.21 (продовження)

Концепція маркетингових комунікацій

1	Купують товар на вимогу	Тематичні зустрічі, конференції, презентації, служба підтримки.	Надання можливості використання мережі для перевірки на наявність пухлини	Демонстрація основних функцій розробленого продукту	Охоплення аудиторії, пояснення функцій та можливостей розробленого продукту та його переваг

ВИСНОВОК ДО РОЗДІЛУ 4

У четвертому розділі ми переглянули розробку стартап проекту для передбачування наявності ракової пухлини мозку у пацієнта згортковою мережею. Було розглянуто ідею проекту також провели аналіз конкурентів які можуть завадити впровадженню на ринок та можливостей через які продукт має шанси зайняти свою нишу на міжнародній арені. У ході роботи буду утворено загальний опис ідеї утворюваного проекту для виходу на ринок також описані доступні можливості стартапу, виділено основних конкурентів та порівняно найкращі якості конкурентів для визначення переваг розроблюваного стартапу над обраними конкурентами. Проведено технічний аудит проекту і визначено можливості реалізації програмного продукту.

Проаналізовано ринок збуту на успішну можливість для запуску стартапу також було визначено перспективи запуску та розвитку проекту в порівнянні з конкурентами та встановлено план і стратегію розвитку продукту, визначено цільові групи клієнтів на які орієнтований наш продукт та створено способи просування проекту. Приорітетною цілью при виході проекту на міжнародну арену є масштабна рекламна компанія для залучення якомога більшої кількості клієнтів.

Підсумовуючи, було створено стартап-проект для запуску розробленого програмного продукту на ринок, отримано навички створення стартап-проектів, побудови маркетингової стратегії та аналізу обраного ринку.

ВИСНОВОК

Результатом магістерської роботи є нейронна мережа для визначення чи є пухлина мозку на зображенні з точністю у 94.3%. Створена програма легка у використанні, не потребує додаткових роз’яснень в використанні та швидко оброблює дані користувача.

Проведено аналіз існуючих аналогів у використанні нейронних мереж для виконання схожих задач.

Визначено основні завдання, які повиння буде виконувати нейронна мережа. Описано дата сет, що був обранний для подальшого навчання згортвокої мережі. Дані поділено у відношені 70%:20%:10% для навчання, валідації та тестування відповідно. Класифікаційні розподілення розділене на наявність та відсутність пухлини на знімку.

Розроблено згорткову мережу для виявлення пухлини мозку. Розглянуто метод для побудови вхідного шару, прихованих шарів, вихідного шару у моделі та метод максимального об`єднання. Для уникнення перенавчання моделі було обрано відсів (dropout) та метод випадкового обертання зображення. Розглянуті методи для налаштування, тренування та зберігання моделі.

Було створенно концепт стартап ідеї на основі дипломного проекту для впровадження його в подальшому на ринок. Було проаналізовано ймовірні проблеми на шляху, переваги та недоліки ідеї та стан сучасного ринку і можливості для впровадження ідеї.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Everything you need to know about deep learning the technology that mimics the human brain Режим доступу: https://www.algotive.ai/blog/everything-you-need-to-know-about-deep-learning-the-technology-that-mimics-the-human-brain Дата доступу: грудень 2022

2.Types of Activation Functions in neural network Режим доступу: https://www.analyticssteps.com/blogs/7-types-activation-functions-neural-network Дата доступу: грудень 2022

3. Activation Functions in neural network Режим доступу: https://www.v7labs.com/blog/neural-networks-activation-functions Дата доступу: грудень 2022

4.Deep learning fundamentals Режим доступу: https://deeplizard.com/learn/video/DEMmkFC6IGM Дата доступу: грудень 2022

5.What is overfitting in deep learning Режим доступу: https://www.v7labs.com/blog/overfitting#h1 Дата доступу: грудень 2022

6. Max pooling Режим доступу: https://deepai.org/machine-learning-glossary-and-terms/max-pooling Дата доступу: грудень 2022

7. What is a keras model Режим доступу: https://www.activestate.com/resources/quick-reads/what-is-a-keras-model/ Дата доступу: грудень 2022

8. How neural network learn Режим доступу: https://towardsdatascience.com/learn-how-neural-networks-learn-d920fab3f72e Дата доступу: грудень 2022

9. CNN Explainer Режим доступу: https://poloclub.github.io/cnn-explainer/ - Дата доступу: грудень 2022

10. Activation functions Режим доступу: https://www.geeksforgeeks.org/activation-functions/ Дата доступу: грудень 2022

[11] 5 Types of Classification Algorithms in Machine Learning Режим доступу: https://monkeylearn.com/blog/classification-algorithms/ Дата доступу: грудень 2022

[12] Cancer diagnosis in histopathological image: CNN based approach Режим доступу: https://www.sciencedirect.com/science/article/pii/S2352914819301133 Дата доступу: грудень 2022

[13] Omdena Режим доступу: https://omdena.com/blog/machine-learning-classification-algorithms/ Дата доступу: грудень 2022

[14] Usage of KNN – IBN Documantation Режим доступу: https://www.ibm.com/docs/en/ias?topic=knn-usage Дата доступу: грудень 2022

[15] Two Ensemble-CNN Approaches for Colorectal Cancer Tissue Type Classification. Режим доступу: https://www.mdpi.com/2313-433X/7/3/51 Дата доступу: грудень 2022

[16] 6 Types of Artificial Neural Networks Currently Being Used in Machine Learning Режим доступу: https://analyticsindiamag.com/6-types-of-artificial-neural-networks-currently-being-used-in-todays-technology/ Дата доступу: грудень 2022