

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____
(повна назва)

Кафедра _____ програмної інженерії _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

Дослідження моделей та методів штучного інтелекту для автоматизованої
обробки та аналізу медичних зображень
(тема)

Виконав:

студент (ка) 2 курсу, групи ІПЗм-22-6

Топчій Богдан Сергійович

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного
забезпечення

(код і повна назва спеціальності)

Тип програми освітньо-наукова

Керівник доц. Чуприна А.С.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

_____ (підпис)

З.В.Дудар

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра програмної інженерії

Рівень вищої освіти другий(магістерський)

Спеціальність 121-Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Зав. Кафедри _____

(підпис)

« ____ » _____ 2024.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Топчій Богдана Сергійовича

(ПІБ)

1. Тема роботи «Дослідження моделей та методів штучного інтелекту для автоматизованої обробки та аналізу медичних зображень».

Затверджена наказом по університету від «29» березня 2024 р. №250 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 25.06.2024 р.

3. Вхідні дані до роботи: Дослідження моделей та методів штучного інтелекту для автоматизованої обробки та аналізу медичних зображень.

4. Перелік питань, що потрібно опрацювати в роботі: Дослідження методів та моделей штучного інтелекту для автоматизованої обробки та аналізу медичних зображень. Аналіз існуючих підходів та їх ефективності у різних сценаріях застосування. Розробка та тестування алгоритму для автоматизованої класифікації медичних зображень. Оцінка точності та швидкості розробленого алгоритму. Порівняння результатів з існуючими методами. Розробка рекомендацій щодо впровадження розробленого методу в практику.

КАЛЕНДАРНИЙ ПЛАН

№	Назви етапів курсової роботи	Термін виконання етапів роботи	Примітка
1	Видача теми, узгодження і затвердження	02.04.2024	виконано
2	Аналіз предметної галузі	02.04.2024	виконано
3	Огляд існуючих методів	09.04.2024	виконано
4	Оформлення пояснювальної записки	05.05.2024	виконано
5	Здача готового проекту	25.06.2024	виконано

Дата видачі завдання 20 січня 2024 р.

Студент _____
(підпис)

Топчій Б. С.
(прізвище, ініціали)

Керівник кваліфікаційної роботи _____
(підпис)

доц. Чуприна А.С.
(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка: 66 с., 19 рисунків, 14 джерел.

АВТОМАТИЗАЦІЯ, АНАЛІЗ, МЕДИЧНІ ЗОБРАЖЕННЯ, ОБРОБКА,
ШТУЧНИЙ ІНТЕЛЕКТ,

Об'єктом глибокого аналізу та дослідження є моделі та методи штучного інтелекту для автоматизованої обробки та аналізу медичних зображень. Метою дослідження є розробка, аналіз та впровадження передових методів автоматизації обробки медичних зображень з використанням штучного інтелекту.

У процесі наукового дослідження застосовуються сучасні підходи та технології штучного інтелекту, зокрема, машинне навчання та глибинне навчання, для автоматизованої обробки медичних зображень. Великий акцент приділяється розгляду різноманітних моделей і методів аналізу зображень, їх адаптації та оптимізації для медичних застосувань.

Основні напрямки дослідження включають аналіз існуючих підходів до обробки медичних зображень, визначення ключових вимог до автоматизованих рішень та розробку алгоритмів інтеграції з медичними системами.

В результаті дослідження розробляються та впроваджуються комплексні інструменти для автоматизації аналізу медичних зображень. Це сприяє не лише підвищенню точності та ефективності діагностики, але й оптимізації процесів обробки медичних даних, що покращує загальну якість медичних послуг та безпеку пацієнтів.

ARTIFICIAL INTELLIGENCE, AUTOMATION, MEDICAL IMAGES,
ANALYSIS, PROCESSING.

The object of in-depth analysis and research is the models and methods of artificial intelligence for the automated processing and analysis of medical images. The

aim of the study is the development, analysis, and implementation of advanced methods for automating the processing of medical images using artificial intelligence.

The research process employs modern approaches and technologies of artificial intelligence, particularly machine learning and deep learning, for the automated processing of medical images. Significant emphasis is placed on examining various models and methods of image analysis, their adaptation, and optimization for medical applications.

The main research directions include the analysis of existing approaches to medical image processing, determining key requirements for automated solutions, and developing integration algorithms with medical systems.

As a result of the research, comprehensive tools for automating the analysis of medical images are developed and implemented. This not only enhances the accuracy and efficiency of diagnostics but also optimizes medical data processing processes, improving the overall quality of medical services and patient safety.

Я, Топчій Богдан Сергійович, студент гр. ІІЗм-22-6, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя робота на тему «Дослідження моделей та методів штучного інтелекту для автоматизованої обробки та аналізу медичних зображень», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений(на) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Перелік скорочень	8
Вступ.....	9
1 Аналіз предметної області та постановка задачі.....	11
1.1 Діагностика онкологічних захворювань	11
1.2 Роль медичних зображень в діагностиці онкології.....	11
1.2.1 Рентгенівські зображення.....	12
1.3 Застосування штучного інтелекту та машинного навчання в аналізі медичних зображень	13
1.4 Постановка задач.....	14
2 Огляд методів та технологій для класифікації зображень.....	18
2.1 Методи та технології аналізу зображень	18
2.2 Класифікація зображень	19
2.3 Штучні нейронні мережі.....	20
2.4 Згорткові нейронні мережі	25
2.5 Машинне навчання в класифікації зображень	28
3 Аналіз методів для тестування.....	29
3.1 Вибір платформи	29
3.2 Порівняння існуючих бібліотек	30
3.2.1 ML.NET	31
3.2.2 Accord.NET	35
4 Експериментальне дослідження	39
4.1 Дата-сет	39
4.2 Розробка алгоритмів для навчання.....	42
4.3 Аналіз результатів	49
Висновки	52
Перелік джерел посилання	53
Додаток А Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ.....	55
Додаток Б Слайди презентації	56

Додаток В Результат проходження на академічний плагіат.....	64
Додаток Г Апробація результатів роботи	65
Додаток Д Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015	66

ПЕРЕЛІК СКОРОЧЕНЬ

КТ – Комп'ютерна томографія

МРТ – Магнітно-резонансна томографія

ПЕТ – Позитронно-емісійна томографія

ML.NET – Machine Learning .NET

Accord.NET – Бібліотека для машинного навчання та обробки зображень на платформі .NET

ВСТУП

Одним із головних пріоритетів сучасної медицини є виявлення захворювань за допомогою медичної діагностики. Кожна людина на певному етапі життя потребує якісної, своєчасної медичної допомоги. Однією з таких послуг є діагностика захворювань, оскільки своєчасна та точна діагностика може значно вплинути на ефективність плану лікування.

Створення різних видів медичних зображень (КТ, МРТ, рентген) для подальшої ручної або автоматизованої обробки та аналізу є одним із основних способів діагностики хвороби.

Аналіз медичного знімка – трудомістка процедура, яка потребує багато часу, максимальної уваги та професіоналізму спеціаліста, а також певних часових витрат. Швидка та точна діагностика часто може значно допомогти та прискорити одужання пацієнта.

Багато онкологічних захворювань діагностуються в основному за допомогою аналізу медичного зображення.

Рак - це патологічна проліферація клітин, яка може поширюватися навіть за відсутності поживних речовин і в місцях росту пухлин. Злоякісні клітини швидко розмножуються по всьому організму, змінюючи форму, не справляючись зі своїми функціями і перетворюючись на патологічні новоутворення. Вони також не реагують на сигнали тіла. Рак вражає людей будь-якого віку, від маленьких немовлят до людей похилого віку, і є однією з головних причин смертності в світі через захворювання.

Одним з найсерйозніших захворювань, яке може вразити як дорослих, так і дітей, є пухлина головного мозку. Від 85% до 90% усіх первинних ракових захворювань центральної нервової системи (ЦНС) є пухлинами головного мозку. Щороку близько 11 700 людей отримують діагноз «пухлина головного мозку». Для тих, хто має злоякісну пухлину мозку або центральної нервової системи, 5-річна виживаність становить приблизно 34% для чоловіків і 36% для жінок. Існують різні класифікації пухлин головного мозку, в тому числі доброякісні, злоякісні, пухлини гіпофіза та інші» [1].

Щоб продовжити життя пацієнтів, необхідна правильна терапія, планування та точна діагностика. Магнітно-резонансна томографія (МРТ) є найбільш ефективним методом виявлення злоякісних новоутворень головного мозку. Під час сканування створюється багато даних зображення. Рентгенолог переглядає ці знімки. Через складність пухлин головного мозку та їх характеристики ручний огляд може бути схильний до помилок.

Пухлини головного мозку складні. Локалізація та розмір пухлини головного мозку можуть сильно відрізнятися. Це робить з'ясування точної природи пухлини надзвичайно складним завданням. Також для проведення МРТ дослідження потрібен досвідчений нейрохірург. Відсутність кваліфікованих лікарів і розуміння пухлин часто трапляється в країнах, що розвиваються, що робить звіти МРТ надзвичайно складними та трудомісткими." Таким чином, цю проблему можна вирішити за допомогою автоматизованої хмарної системи.

Використання штучного інтелекту (AI) і машинного навчання (ML) для автоматизації процедур класифікації постійно демонструє вищу точність, ніж ручна категоризація. Таким чином, було б корисно для клініцистів у всьому світі запропонувати систему, яка використовує алгоритми глибокого навчання для виконання виявлення та класифікації за допомогою згорткових нейронних мереж (CNN), штучних нейронних мереж (ANN) і трансферного навчання (TL). Метою цієї роботи є створення моделі машинного навчання для бінарної категоризації МРТ-зображень мозку, яка є водночас високоточною та ресурсоефективною.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Діагностика онкологічних захворювань

В даний час метод дискримінантного аналізу є найбільш часто використовуваним підходом для прийняття рішень щодо медичного діагнозу пацієнта. Дискримінантний аналіз – це процес ідентифікації об'єктів спостереження за пов'язаними ознаками з метою вирішення проблем класифікації. Відсоток правильно класифікованих елементів за допомогою прогнозного рівняння спостережень використовується для розрахунку точності класифікації цього аналізу.

Поки у вихідній вибірці даних є кілька вимірів, аналізується лише частина даних. Прогнозне рівняння, створене на основі прикладів першого розділу, використовується для класифікації спостережуваних об'єктів для іншої половини даних».

Однак існує зв'язок із попереднім досвідом медичного експерта, і це іноді може призвести до неточного діагнозу, а також до суб'єктивної оцінки отриманих експертом медичних фактів.

Розробки технологій машинного навчання та обробки зображень зробили можливим створення систем автоматичного виявлення та діагностики (CAD / CADx). надійний з точки зору діагностики.

1.2 Роль медичних зображень в діагностиці онкології

Збір різних типів медичних знімків тісно пов'язаний з сучасними інструментальними діагностичними процедурами, що застосовуються в онкології. Ці фотографії можна класифікувати наступним чином залежно від того, як вони були отримані:

- рентгенологічні зображення;
- магнітно-резонансно томографічні зображення (МРТ);
- ультрасонографічні зображення (УЗД).

Кожне зображення з вищезазначених методів можна використовувати для ідентифікації різних форм раку; однак кожна техніка має певні особливості

застосування, а також переваги та недоліки.

1.2.1 Рентгенівські зображення

Промінь рентгенівського випромінювання, який пройшов через тіло людини, аналізується кількісно та якісно, щоб дослідити структуру та функцію багатьох органів і систем за допомогою радіологічного методу.

Штучне контрастування використовується для створення чіткого зображення тканин, які приблизно однаково поглинають випромінювання. Для цього в тіло вводять речовини, які поглинають рентгенівське випромінювання сильніше або слабше, ніж м'які тканини, створюючи достатньо сильний контраст із досліджуваним органом. Рентгенопозитивні речовини – це речовини, які поглинають більше випромінювання, ніж м'які тканини.

Нагадаємо, що рентгенівське зображення є негативним у порівнянні із зображенням на флуоресцентному екрані під час просвітлення. З цієї причини прозорі для рентгенівських променів частини тіла виглядають темнішими (потемніння), а щільніші частини тіла – світлішими (просвітлення). Коли промінь проходить через тіло людини, він проходить через кілька місць, які знаходяться як глибоко в тканинах, так і на їх поверхні. У результаті кожна точка на зображенні є проекцією багатьох реальних точок на об'єкт; в результаті рентгенівське зображення є сумаційним, плоским. Основна вказівка для рентгенологічного дослідження полягає в наступному: рентгенограми будь-якої ділянки тіла повинні бути отримані принаймні у двох взаємно перпендикулярних проекціях, таких як бічна і пряма. На рисунку 1.2 показано це. Крім того, можуть знадобитися осьові та косі види.



Рисунок 1.2 – Рентгенівський знімок голови

У державній програмі раннього виявлення раку легенів і туберкульозу, що діяла за часів Радянського Союзу, рентгенологічний метод використовувався як скринінговий метод масового профілактичного огляду населення через його низьку вартість і значну пропускну здатність. Однак у країнах Європейського співтовариства від класичної плівкової флюорографії відмовилися через її обмежену роздільну здатність і відносно значне радіаційне навантаження.

1.3 Застосування штучного інтелекту та машинного навчання в аналізі медичних зображень

Штучний інтелект і аналіз медичних зображень на основі машинного навчання можуть допомогти радіологам лікувати пацієнтів продуктивніше та ефективніше. Вони можуть швидко знайти пацієнтів і полегшити їх самопочуття.

У зв'язку з тим, що медичний бізнес зараз усвідомлює цінність технологій обробки медичних зображень, безліч постачальників і спеціалістів із технологій змагаються, щоб запропонувати найкращі інструменти та програми.

У результаті радіологи можуть аналізувати дані з більшою точністю, оскільки вони мають вищу якість. Збір даних і остаточна звітність стають кращими завдяки зростаючому використанню технологій ШІ та МН у медичній галузі. Для встановлення точного діагнозу радіологи можуть ознайомитися з історією хвороби пацієнта та результатами попереднього лікування. Співпраця між цими технологіями та кваліфікованим радіологом вирішує важливі проблеми, пов'язані з лікуванням захворювань, навіть у складних медичних контекстах.

Експерти з прийняття рішень і ранньої діагностики захворювань можуть отримати вигоду від здатності алгоритмів глибокого навчання вчитися та виконувати різноманітні важливі завдання. Щоб підвищити ефективність лікування, вони мають доступ до свіжих і ранніх даних. Це пояснює, чому технології машинного навчання та штучного інтелекту охоплюють сектор охорони здоров'я. Нижче наведено деякі переваги штучного інтелекту та мультимедіа при аналізі медичного зображення:

- покращений аналіз медичних зображень на основі ШІ та МН може

допомогти медичним працівникам працювати ефективніше та продуктивніше в клініці;

- машинне навчання може допомогти нам створити модель, яка точно діагностує такі захворювання, як рак або пухлини. Алгоритми використовуються ШІ та технологією машинного навчання, що знижує ймовірність помилок. За різними медичними фотографіями штучний інтелект може розпізнати ракові клітини. Система штучного інтелекту може досягти точності до 99%, але медичний працівник може досягти лише 38% точності;
- команда лікарів зможуть швидше виявляти хвороби, приймати обґрунтовані рішення та оптимізувати робочий процес, маючи доступ до технологій аналізу медичних зображень, побудованих на основі штучного інтелекту та машинного навчання. Зрештою, це покращує досвід пацієнта та підвищує рівень задоволеності.

Більшість медичних діагностичних завдань, таких як класифікація та групування хвороб і прогнозування численних епідеміологічних компонентів, виконуються нейронними мережами. Нейронні мережі набули популярності, оскільки їх можна використовувати як програму для збору додаткової інформації з неструктурованих даних, включаючи рентгенівські знімки або записи кардіограми функції серця, на додаток до того, щоб бути алгоритмом для винесення суджень щодо конкретного пацієнта.

Одним із недоліків цього методу є те, що він вимагає великої обчислювальної потужності та великих наборів даних для навчання нейронної мережі. Однак у міру розвитку хмарних технологій, особливо з точки зору обробки великих даних і архітектури зберігання, цей недолік стає все менш очевидним у реальних програмах.

1.4 Постановка задачі

Наукові дослідження в галузі штучного інтелекту є затребуваними через актуальність досліджуваної проблеми в її зображенні, значущість аналізу

медичного зображення в процесі діагностики та широке використання різноманітних методів інтелектуального аналізу за допомогою нейронних мереж.

У попередніх підрозділах було оцінено та досліджено методи аналізу медичних зображень, а також життєздатність і ступінь їх застосування. Визначено, що для діагностики раку мозку за допомогою магнітно-резонансної томографії необхідна розробка алгоритму бінарної класифікації, тобто класифікації медичних зображень на дві групи.

Ця робота має на меті вивчити та розробити методику класифікації ознак на медичних фотографіях, які розділені на дві протилежні категорії. Достовірна колекція даних, розподілених за ознаками, повинна слугувати основою для процесу класифікації ознак.

Основними критеріями оцінки розробленого методу є:

- висока точність категоризації, понад 95%;
- досягнення високої точності з 2000 медичними фотографіями як найменшою кількістю навчальних даних;
- швидка класифікація – менше 500 мілісекунд.

Щоб виконати поставлені задачі, потрібно:

- досліджувати методи розпізнавання та категоризації ознак об'єкта в медичній візуалізації;
- вивчити поточні методи глибокого навчання щодо того, як вони використовуються з медичною візуалізацією;
- створити нейромережевий метод розпізнавання знімків магнітно-резонансної томографії;
- дослідити ефективність створеного алгоритму, взявши за основу категоризацію зображень МРТ.

Машинне навчання має численні сфери застосування в медицині, що охоплюють широкий спектр завдань від діагностики до персоналізованого лікування. Однією з основних областей є аналіз медичних зображень, де алгоритми машинного навчання допомагають лікарям виявляти патології на рентгенівських, КТ і МРТ знімках. Ці алгоритми можуть виявити навіть

найдрібніші аномалії, які можуть бути непомітні для людського ока, що значно підвищує точність діагностики.

Інша важлива сфера застосування – це прогнозування захворювань і ризиків. Використовуючи великі обсяги даних пацієнтів, такі як історії хвороби, генетичну інформацію та дані з носимих пристроїв, алгоритми можуть прогнозувати ризик розвитку певних захворювань у пацієнтів. Це дозволяє лікарям здійснювати раннє втручання та розробляти превентивні заходи, що можуть запобігти розвитку захворювання або зменшити його важкість.

Машинне навчання також відіграє ключову роль у розробці нових ліків. Алгоритми можуть аналізувати хімічні властивості різних сполук і прогнозувати їхню ефективність проти певних захворювань, що значно скорочує час і витрати на розробку нових препаратів. Крім того, ці технології можуть визначати потенційні побічні ефекти на ранніх стадіях розробки.

Персоналізована медицина – ще одна галузь, де машинне навчання має величезний вплив. Аналізуючи дані генетичних тестів, медичні дані та інформацію про спосіб життя пацієнтів, алгоритми можуть допомагати лікарям розробляти індивідуальні плани лікування, що максимізують ефективність і мінімізують побічні ефекти. Це дозволяє забезпечити більш точний і ефективний підхід до лікування кожного пацієнта.

Також варто згадати застосування машинного навчання у системах підтримки прийняття рішень для медичних працівників. Такі системи можуть аналізувати симптоми пацієнтів і пропонувати можливі діагнози або варіанти лікування, допомагаючи лікарям приймати обґрунтовані рішення на основі великої кількості даних.

Окрім цього, машинне навчання застосовується у моніторингу пацієнтів у реальному часі. Завдяки носимим пристроям, які збирають дані про стан здоров'я, такі як пульс, артеріальний тиск та рівень глюкози в крові, алгоритми можуть виявляти аномалії та попереджати медичний персонал про можливі проблеми ще до того, як вони стануть серйозними.

Усе це демонструє, що машинне навчання стає невід'ємною частиною

сучасної медицини, допомагаючи підвищувати ефективність лікування, знижувати витрати та покращувати якість медичних послуг.

2 ОГЛЯД МЕТОДІВ ТА ТЕХНОЛОГІЙ ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

2.1 Методи та технології аналізу зображень

«Обробка та аналіз зображення – це методичний процес, який спирається на знання та досвід оператора, а також на результати попереднього етапу. Якість зображення покращується на етапі попередньої обробки, а етап сегментації підкреслює компоненти зображення, які разом підвищити рівень якості та точності діагностики».

Наразі цифрові медичні зображення замінюють аналогові. Оцифрування полегшує обробку, зберігання та передачу візуальних даних, починаючи з точки отримання. Під час збору та обробки медичних зображень інформаційні технології можуть бути корисними на всіх рівнях.

Ядерний магнітний резонанс, комп'ютерна томографія та позитронно-емісійна томографія (ПЕТ) є трьома прикладами методів візуалізації, які створюються безпосередньо комп'ютерами та не можуть бути досягнуті іншими засобами.

Серед застосувань цифрової обробки зображень:

- підвищення якості зображення, пом'якшення шумів і усунення недоліківзаписуючого обладнання;
- обчислення терапевтично значущих кількісних величин (відстань, площа,об'єм тощо);
- оптимізована інтерпретація (розрахунок дози променевої терапії, структурне розпізнавання);
- встановлення зворотного зв'язку через автоматизовані хірургічні процедури.

«Стиснення зображень скорочує час, потрібний для передачі даних, і обсяг пам'яті, необхідний для їх збереження. Оцифроване зберігання зображень на компакт-дисках або жорстких магнітних дисках полегшує доступ до архіву та його організацію. Експерти з різних медичних центрів можуть швидко обговорити діагноз і прийняти рішення щодо нього. коли цифрові зображення передаються між

ними швидка експертна консультація для прийняття рішень щодо діагнозу чи лікування, що покращує моніторинг догляду за пацієнтом (телерадіологія, телепатологія)». [8]

Існують окремі етапи обробки зображень, а саме:

- первинна обробка. Попередня обробка мінімізує шум і позбавляє від аномалій, пов'язаних з механізмом генерації зображення. Використовувані методи покращують видимість конкретних анатомічних компонентів шляхом обробки цифрових даних за допомогою спеціалізованих алгоритмів;
- зміна контрастності зображення. Кількість пікселів для кожного рівня сірого на зображенні представлено гистограмою зображення. Нелінійний розподіл гистограми призведе до втрати деталей. Контраст i , таким чином, відображення деталей покращуються операціями вирівнювання гистограми;
- сегментація. На цьому етапі обробки зображення окремі елементи зображення, такі як клітини та органи, відокремлюються. Ідентифікація ідентичних пікселів з допустимим ступенем похибки є основою процедури. Динаміка визначається шляхом порівняння двох сегментованих зображень, що відрізняються за часом;
- розрахунок параметрів. Розраховуються лінійні та об'ємні властивості анатомічних структур. інтерпретація зображення.

Проблема автоматичного комп'ютерного перекладу залишається актуальною. Для його якісного виконання необхідне фундаментальне розуміння як патологічної, так і порівняльної анатомії. Необхідно порівняти отримані структури та параметри з існуючими структурами перед їх класифікацією.

2.2 Класифікація зображень

У комп'ютерному зорі (CV) однією з проблем є класифікація зображень. Це використовується для вирішення проблеми категоризації або маркування колекції вхідних фотографій. Крім того, категоризацію зображення можна використовувати

для спрощення інших завдань комп'ютерного зору, таких як сегментація або розпізнавання об'єктів. Наприклад, визначення того, чи видно об'єкт на фотографії, як-от визначення вуличних ліхтарів і дорожніх знаків або наявності кота.

Спочатку виділяють основні ознаки, а потім будують модель для вивчення вилучених ознак. На рисунку 2.1 наведено приклад використання того самого екстрактора ознак між моделлю та вхідними даними для прогнозування міток після того, як модель навчена.

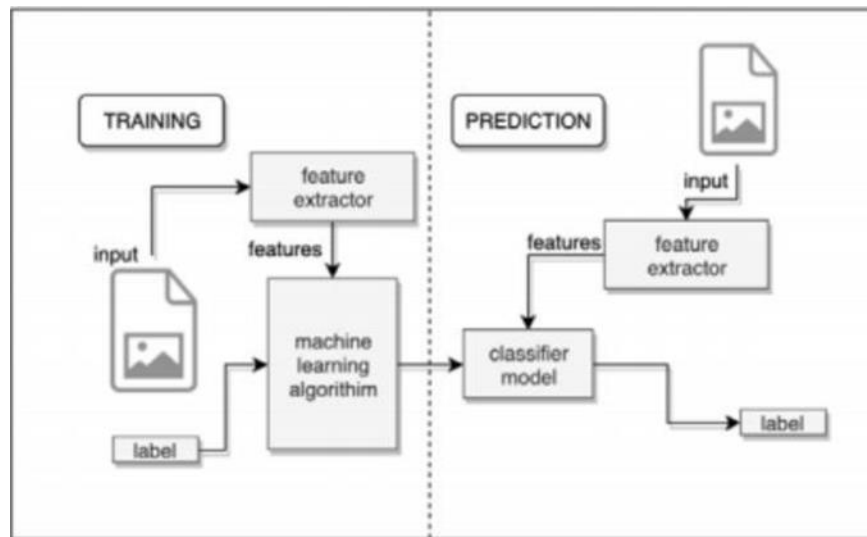


Рисунок 2.1 – Процес розпізнавання зображення за допомогою витягнення

До класифікації зображень можна включити такі дії:

- підготовка даних;
- попередня обробка зображення, наприклад методи шумозаглушення, корекції та інші.
- виявлення істотних характеристик на зображенні;
- класифікація: використовуючи знайдені ключові якості, фотографії сортуються за попередньо встановленими групами.

2.3 Штучні нейронні мережі

Математична модель, заснована на принципах функціонування нейронів людини, називається нейронною мережею. Саме тут беруть початок нейрони,

синапси та функції активації нейронів, основні будівельні блоки традиційної архітектури нейронної мережі.

Давайте детальніше розглянемо кожен із основних структурних компонентів класичної нейронної мережі.

Нейрон – це одиниця очищення, яка приймає інформацію, обробляє її за допомогою помилки очищення, а потім передає далі. Три основні категорії: вихід, примітив введення та вихід. Термін шару вводиться у випадку, коли неупомережа складається з багатьох нейронів. У результаті інформація приймається вхідним рівнем, обробляється n заштрихованими шарами, а виводиться вихідним рівнем. Вхідні та вихідні дані є двома основними параметрами, якими користуються всі нейрони.

Коли справа доходить до функцій активації, важливо знати, що кожна з них має свої плюси та мінуси та може застосовуватися на основі конкретної мети, яка визначається до побудови нейронної мережі. Давайте розглянемо основні та найбільш часто використовувані функції активації нейронів нейронної мережі.

Одним із прикладів цього є функція ступінчастої активації (формула 2.1), яка має наступне визначення і за своєю природою є кусково-лінійною:

$$\varphi(x, \alpha) = \begin{cases} 1, & x > \alpha \\ 0, & \text{впротиленому випадку} \end{cases} \quad (2.1)$$

Функція Хевісайда (формула 2.2) є конкретним прикладом окремого випадку цієї функції:

$$\mu(x) = \begin{cases} 1, & x > 0 \\ 0, & \text{впротиленому випадку} \end{cases} \quad (2.2)$$

Таким чином, задіяний нейрон матиме значення 1 під час виклику цієї функції, тоді як неактивний нейрон прийматиме значення 0.

У задачах бінарної класифікації, підмножині задач класифікації, де кількість класів, на які слід класифікувати об'єкт, дорівнює двом, ця функція активації

широко використовується. Крім того, якщо в задачі, що включає багатокласову класифікацію, є більше двох класів, функція не може бути побудована таким чином. Це пов'язано з тим, що нейрон можна одночасно стимулювати для кількох класів при використанні функції поетапної активації для багатокласової класифікації.

Ці елементи об'єднуються, щоб утворити звичайну нейронну мережу, фундаментальна структура якої показана на рисунку 2.2. Кожен прихований шар мережі складається з набору нейронів, з'єднаних з кожним іншим нейроном у шарі над ним.

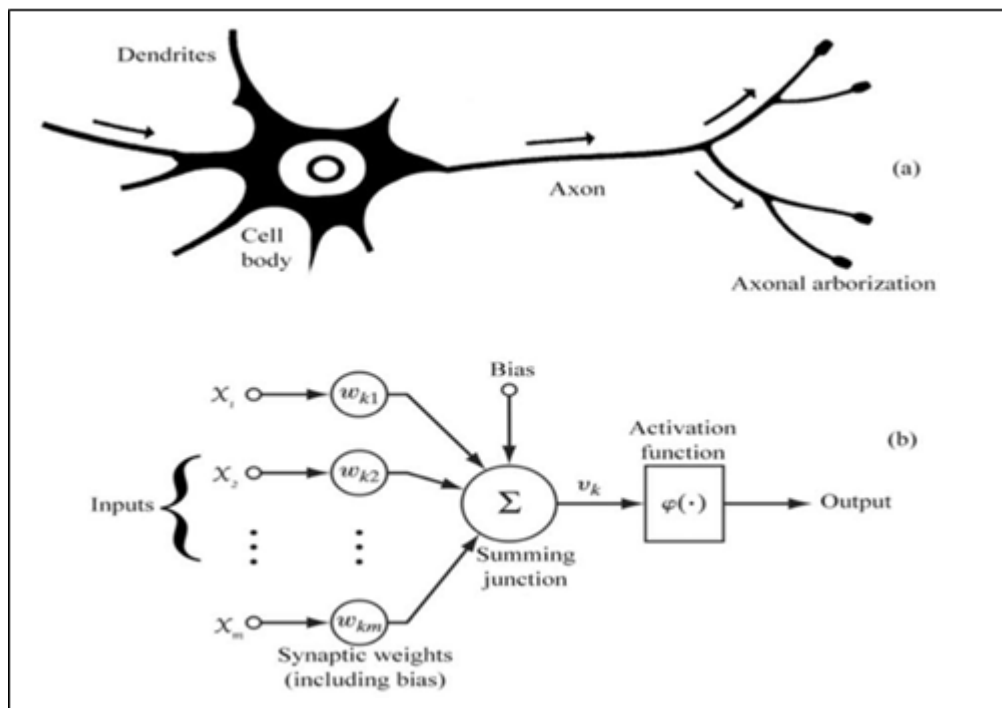


Рисунок 3.1 – Структура нейронної мережі

Класифікація нейронних мереж на основі топології:

- повно зв'язні мережі – усі вхідні сигнали надходять до всіх нейронів, кожен з яких передає вихідний сигнал іншим нейронам, у тому числі й самому собі(рис. 2.3, а);
- багатошарові або шаруваті мережі – зовнішні вхідні сигнали, які надсилаються на вхід нейронів рівня 1 (нульового), служать лише точками розподілу; посилення і перетворення сигналу відбувається в проміжних (прихованих) шарах; а виходи мережі – це вихідні сигнали

зовнішнього рівня (рис.2.3, б);

- слабо зв'язані мережі – мережі, які є багаторівневими та мають невелику кількість зв'язків (рис. 2.3, в)» [9].

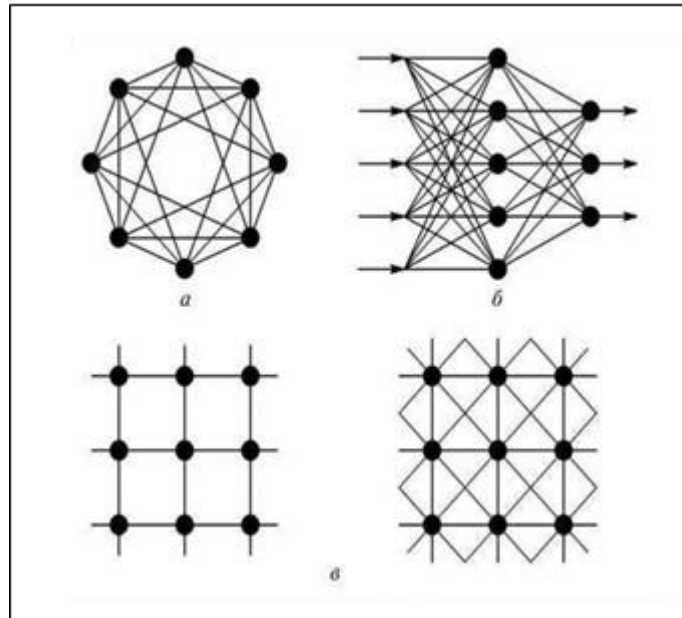


Рисунок 2.3 – Архітектури нейронних мереж: а – повнозв'язна мережа, б – багат шарова мережа з послідовнимизв'язками, в –слабкозв'язана мережа

Перший шар нейронів називають «вхідним» шаром; кількість нейронів у цьому шарі визначається розмірністю вхідних даних. Останній повністю пов'язаний шар нейронів, або вихідний рівень, вказує на кількість класів, на які ми хочемо «розбити» вхідний набір значень. Цей шар складається з усіх нейронів попереднього шару, з'єднаних з усіма нейронами шару над ним. Він вказує, на скільки класів ми хочемо «розділити» вхідний набір даних під час виконання завдань класифікації.

Треба також враховувати процес навчання нейронної мережі. Нейронні мережі є хорошим прикладом методів попереднього машинного навчання, тому навчання стає питанням мінімізації функціональних помилок мережі. Навчання нейронної мережі суттєво відрізняється від інших методів машинного навчання тим, що воно відбувається в два етапи:

- етап зворотного розповсюдження помилки (back propagation);
- етап прямого розповсюдження помилки (forward propagation).

Нейронна мережа використовує процедуру прямого розповсюдження помилки для визначення передбачуваного відкликання моделі шляхом розповсюдження вхідного сигналу по всій конструкції мережі, як показано на рисунку 2.4.

Внесок кожного нейрона в загальне значення вибраного функціоналу якості потім обчислюється під час фази зворотного поширення помилки, що призводить до коригування вагового коефіцієнта кожного нейрона.

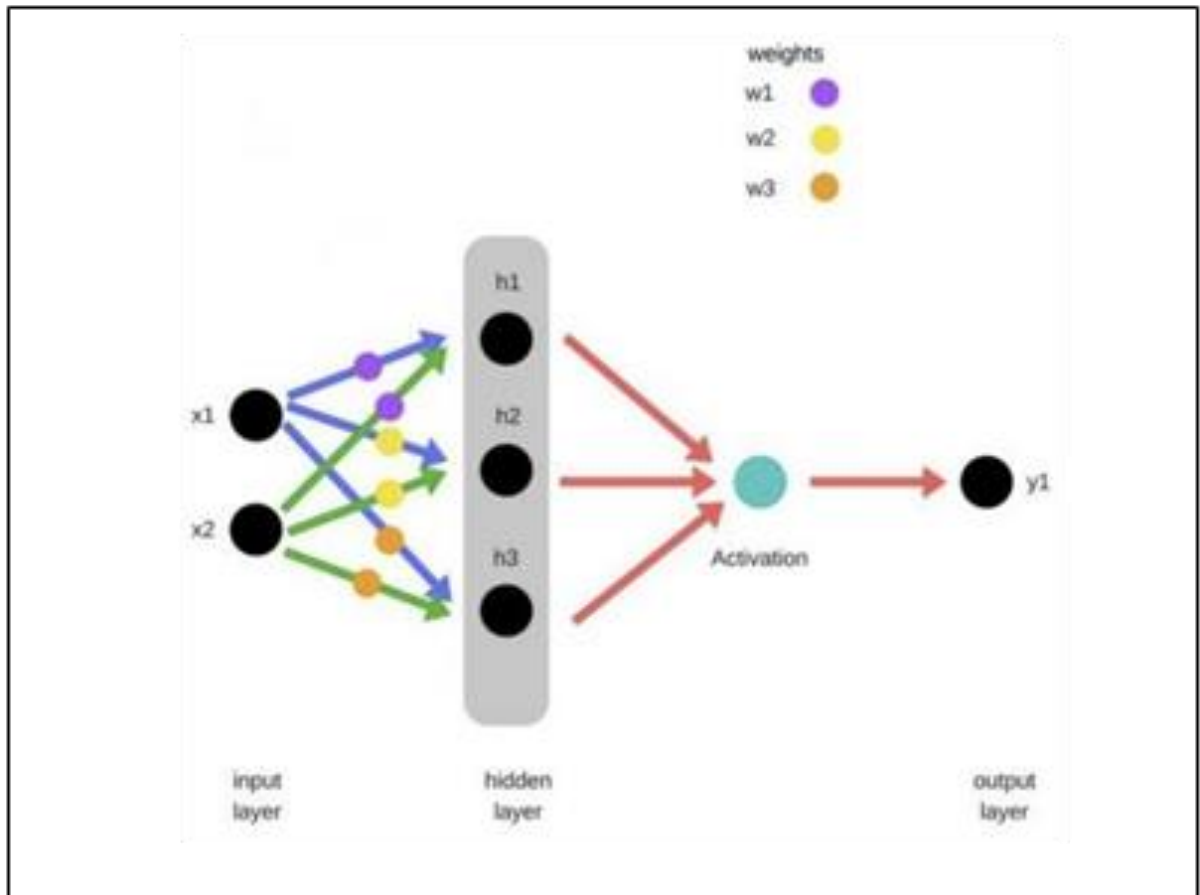


Рисунок 2.4 – Алгоритм прямого розповсюдження помилки

Визначаючи часткову похідну функції помилки за пов'язаним ваговим коефіцієнтом, процес зворотного розповсюдження, який показано на рисунку 2.5, обчислює значення функції помилки для даного об'єкта та відображає внесок кожного вагового коефіцієнта. Підхід градієнтного спуску фактично використовується послідовно для кожного шару нейронної мережі протягом усього процесу навчання.

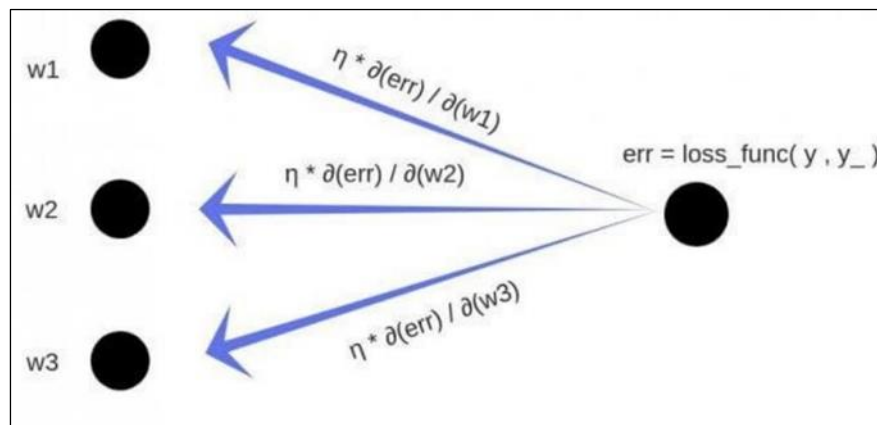


Рисунок 2.5 – Алгоритм зворотного розповсюдження помилки

Слід зазначити, що на цьому етапі навчання нейронної мережі часто застосовується диференціація складної функції (ланцюгове правило) (формула 2.3):

$$y = f(u(x)) \rightarrow \frac{dy}{dx} = \frac{dy}{du} \frac{du}{dx} \quad (2.3)$$

Процес навчання для нейронних мереж виконується ітераційно через задану кількість епох або циклів алгоритму навчання. Це дає ідеальні значення вагового коефіцієнта, або гіперпараметри, для нейронної мережі.

Класичні нейронні мережі часто використовувалися для вирішення проблем класифікації зображень; однак із розвитком технологій глибокого машинного навчання згорткові нейронні мережі – більш спеціалізований підхід – почали використовуватися для класифікації зображень.

2.4 Згорткові нейронні мережі

Основне використання згорткових нейронних мереж (CNN) полягає у вирішенні питань сегментації зображення та класифікації. Ці мережі мають певні зміни в архітектурі, тому що вони представляють собою прогрес у порівнянні з традиційними нейронними мережами.

Згортковий шар, шар об'єднання, шар відсіву та щільний шар нейронів є

чотирма основними шарами цих нейронних мереж.

Давайте розглянемо кожен з первинних архітектурних рівнів згорткових мереж більш детально.

Згортковий шар, який утворює основу повної нейронної мережі, складається з ядра згортки, яке є двовимірною матрицею з розмірами часто 3x3 або 5x5. «ядро згортки» – це назва двовимірної матриці, зазвичай розміром 3 на 3 або 5 на 5.

Процес «згортки» відбувається у такі етапи:

- вирахування скалярного добутку декодованої n-мірної матриці вхідного зображення та ядра згортки;
- щоб обчислити скалярний добуток ядра згортки з частиною матриці вхідного зображення, яка не використовувалася на попередній фазі, «зсуваючи» ядро згортки на відповідний крок.

На перших двох етапах нейронна мережа вироблятиме «карту функцій», причому кількість цих карт буде змінюватися відповідно до кількості ядер згортки. Кожна карта функцій, як показано на рисунку 2.6, визначає певну функцію, яка буде «знайдена» у вхідному зображенні.

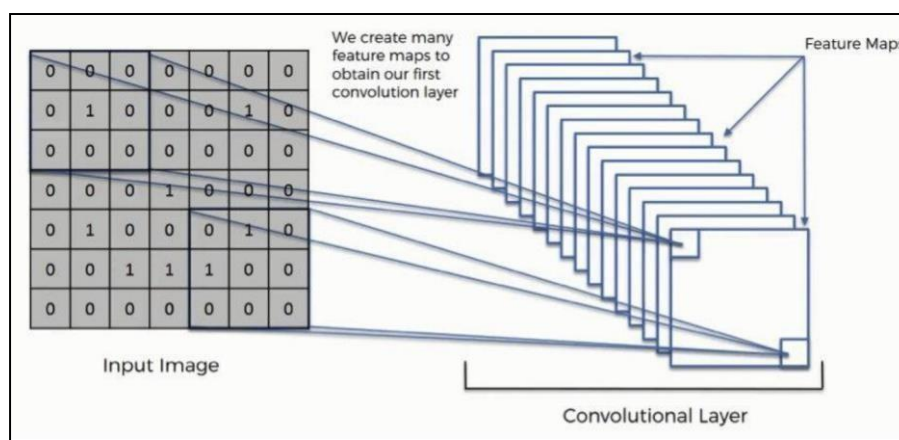


Рисунок 2.6 – Візуалізація роботи першого шару згортки розміром 5 x 5

Карта функцій, отримана на попередньому етапі, також буде використовуватися як вхідне зображення на наступних етапах згортки.

У результаті нейронна мережа дізнається все більшу кількість абстрактних шаблонів на кожному наступному етапі згортки, що зрештою дозволить їй розрізняти такі ідеї, як колір очей і форма обличчя, як показано на рисунку 2.7.

Далі розглянемо шар субдескретизації. Метою цього шару є мінімізація розміру карти функцій, створеної після шару згортки. Щоб уникнути перетренування нейронної мережі та вивчення нелокальних закономірностей, це зроблено. Хоча точне розташування функції не є обов'язковим для майбутнього аналізу зображення, її присутність у певній області зображення все одно є важливою.

Цей шар дуже легко реалізувати: ми переходимо по карті функцій за допомогою унікального «вікна», яке є двовимірною матрицею (зазвичай 2×2). Це означає, що на кожному кроці ми беремо до уваги лише 4 значення, які є суміжними одне з одним на карті функцій, і вибираємо значення, яке є максимальним із них (через цю функцію цей шар також називають максимальним об'єднанням).

Ще один метод, який часто використовувався, – об'єднання середніх значень, яке передбачає використання двовимірної матриці для повторного проходження карти ознак і вилучення середнього значення характеристик, які потрапляли у відповідне «вікно» на кожному кроці. Однак у реальному використанні цей підхід працює дещо гірше, ніж максимальне об'єднання.

Таким чином, на виході шару буде отримано ту саму двовимірну карту ознак, але її розміри ширини та довжини будуть удвічі меншими, ніж у вхідних даних.

Ми можемо ідентифікувати ключові характеристики у нашому вхідному зображенні та створити відповідні вектори ознак, також відомі як дескриптори, перемикаючись між шарами згортки та об'єднання.

Рівень Dropout, наступний рівень у нейронній мережі, функціонує на основі наступного принципу: на цьому етапі ми випадковим чином «вимикаємо» певну кількість нейронів (зазвичай 20%), тобто ці нейрони створені для не реагувати на функції, які є результатом попереднього шару. Ця техніка зменшує фокус нейронів на неважливих, ненормальних аспектах і запобігає перенапруженню нейронної мережі. Зазвичай цей рівень використовується після рівня об'єднання.

Повністю пов'язаний шар нейронів становить перший шар згорткової нейронної мережі. Він використовується для обробки характеристик, які повертаються згортковим шаром.

2.5 Машинне навчання в класифікації зображень

Сфера штучного інтелекту, яка займається розробкою та аналізом методів навчання, називається машинним навчанням. Як правило, підхід «навчання» інтерпретується як навчання за прецедентами, що можна широко сформулювати так: заданий кінцевий набір об'єктів (прецедентів), кожен з яких описується певним набором даних, зібраних для кожного об'єкта в наборі. , необхідно розпізнавати та відновлювати конкретні закономірності та зв'язки. Ця група таких об'єктів, яку часто називають навчальною вибіркою, має опис кожної властивості. Якісний функціонал закріплюється під час підготовки прекондиціонера; це критерій, який визначає, як здійснюється процес навчання і значення якого показує, наскільки добре побудована модель описує вхідні дані. Додатково будується сімейство моделей, з яких буде обраний остаточний алгоритм. Кожна модель сімейства має унікальну мінливість, яка визначається окремим набором параметрів моделі. Оптимальне значення функціоналу якості на навчальній вибірці буде використано остаточним алгоритмом, обраним протягом фази навчання.

3 АНАЛІЗ МЕТОДІВ ДЛЯ ТЕСТУВАННЯ

3.1 Вибір платформи

Ми обрали платформу .NET для дослідження методів та бібліотек машинного навчання для аналізу медичних зображень з кількох важливих причин.

По-перше, .NET є потужною та гнучкою платформою, що забезпечує високий рівень продуктивності і масштабованості. Вона підтримує багатомовність, дозволяючи використовувати C#, F#, VB.NET та інші мови програмування, що дає розробникам можливість обирати найбільш зручний і ефективний інструмент для виконання конкретних завдань. Ця гнучкість особливо важлива для розробки складних алгоритмів машинного навчання, які потребують високої продуктивності.

По-друге, .NET має широкий спектр вбудованих бібліотек і фреймворків для обробки зображень та машинного навчання, таких як ML.NET та Accord.NET. ML.NET є потужним інструментом для створення, тренування та впровадження моделей машинного навчання, що дозволяє легко інтегрувати алгоритми в існуючі додатки. Accord.NET забезпечує широкий набір інструментів для роботи з медичними зображеннями, включаючи обробку сигналів, комп'ютерний зір і статистичний аналіз.

По-третє, .NET екосистема активно підтримується та оновлюється компанією Microsoft, що гарантує стабільність, безпеку та постійний розвиток платформи. Це дозволяє нам бути впевненими в тому, що наші розробки будуть підтримуватися в довгостроковій перспективі і мати доступ до останніх технологічних досягнень.

Ще однією важливою перевагою .NET є його інтеграція з іншими популярними інструментами та сервісами Microsoft, такими як Azure. Використання Azure дозволяє масштабувати обчислювальні потужності, що є критично важливим для обробки великих масивів медичних зображень і тренування моделей машинного навчання. Крім того, хмарні сервіси Azure забезпечують безпеку даних і відповідність регуляторним вимогам, що є особливо важливим у сфері медицини.

Загалом, вибір платформи .NET для нашого дослідження обумовлений її потужністю, гнучкістю, широким набором інструментів для машинного навчання, стабільністю та інтеграцією з хмарними сервісами. Усі ці фактори роблять .NET ідеальним вибором для розробки ефективних і надійних рішень для аналізу медичних зображень.

3.2 Порівняння існуючих бібліотек

У дослідженні нами було прийнято рішення порівняти ML.NET та Accord.NET .

Порівняння ML.NET та Accord.NET має кілька вагомих причин, особливо в контексті нашого дослідження методів і бібліотек машинного навчання для аналізу медичних зображень.

По-перше, обидві бібліотеки мають свої унікальні особливості і можливості, які можуть бути корисними для різних аспектів аналізу медичних зображень. ML.NET є сучасною бібліотекою від Microsoft, спеціально розробленою для інтеграції машинного навчання в .NET додатки. Вона підтримує широкий спектр алгоритмів машинного навчання, включаючи класифікацію, регресію, кластеризацію та рекомендаційні системи. ML.NET також має зручний інтерфейс і добре інтегрується з іншими сервісами Microsoft, такими як Azure Machine Learning, що дозволяє легко масштабувати рішення і використовувати хмарні обчислювальні потужності.

Accord.NET, з іншого боку, є більш універсальною бібліотекою для наукових обчислень і обробки сигналів. Вона включає в себе багатий набір інструментів для комп'ютерного зору, обробки зображень, статистичного аналізу та машинного навчання. Accord.NET має потужні можливості для роботи з зображеннями, включаючи фільтрацію, сегментацію, детекцію об'єктів та інші важливі завдання, які можуть бути особливо корисними в медичній діагностиці.

По-друге, порівняння ML.NET та Accord.NET дозволить нам визначити, яка з цих бібліотек краще підходить для конкретних завдань в нашому проекті. Наприклад, ML.NET може бути більш зручною для швидкого прототипування і

інтеграції моделей машинного навчання в існуючі додатки, тоді як Accord.NET може надати більш розширені інструменти для обробки та аналізу зображень, що може бути критично важливим для підвищення точності діагностики.

Третя причина – це потенційна можливість комбінування можливостей обох бібліотек. Оскільки Accord.NET має сильні інструменти для обробки зображень, а ML.NET для машинного навчання, їхнє спільне використання може дати найкращі результати. Наприклад, попередня обробка зображень може бути виконана за допомогою Accord.NET, після чого підготовлені дані можуть бути передані в ML.NET для тренування та використання моделей машинного навчання.

Нарешті, порівняння цих бібліотек допоможе нам зробити обґрунтований вибір на основі об'єктивних критеріїв, таких як продуктивність, точність, зручність використання, масштабованість та підтримка. Це забезпечить, що ми оберемо найбільш підходящий інструмент для нашого проекту, що в кінцевому результаті підвищить якість нашої роботи і допоможе досягти поставлених цілей у дослідженні методів аналізу медичних зображень.

Таким чином, порівняння ML.NET та Accord.NET є важливим кроком для визначення найкращого підходу до вирішення наших завдань і забезпечення високої ефективності та точності аналізу медичних зображень у нашому проекті.

3.2.1 ML.NET

ML.NET – це бібліотека машинного навчання від Microsoft, розроблена для інтеграції машинного навчання в .NET додатки. Вона надає розробникам можливість створювати, тренувати і використовувати моделі машинного навчання без необхідності залишати екосистему .NET. Проведемо детальний огляд особливостей і сфер застосування ML.NET:

Особливості ML.NET:

- широкий спектр алгоритмів машинного навчання: класифікація: підходить для завдань, де необхідно розподілити дані на категорії, наприклад, класифікація електронної пошти як спам або не спам;
- регресія: використовується для передбачення числових значень, таких як

- прогнозування цін на нерухомість;
- кластеризація: допомагає групувати дані на основі схожості, що корисно для сегментації ринку;
 - рекомендаційні системи: створення рекомендацій на основі історії користувачів, наприклад, рекомендації товарів в інтернет-магазинах;
 - інтеграція з іншими сервісами microsoft: azure machine learning: підтримка хмарних обчислень дозволяє масштабувати тренування моделей і використовувати потужні ресурси azure;
 - data preparation: інтеграція з power bi та іншими інструментами для підготовки та візуалізації даних;
 - зручний api та підтримка багатомовності: c# та f#: основні мови для використання ml.net, що забезпечують гнучкість і зручність для розробників .net;
 - легка інтеграція: можливість легко інтегрувати машинне навчання в існуючі додатки на .net;
 - підтримка automl: автоматичне машинне навчання: automl допомагає автоматично обирати найкращі алгоритми та гіперпараметри для конкретного завдання, що спрощує процес створення моделей для користувачів без глибоких знань у машинному навчанні;
 - модульність та розширюваність: трансформації даних: підтримка різноманітних трансформацій даних, таких як нормалізація, кодування категорійних змінних та багато іншого;
 - пайплайни: можливість створення складних пайплайнів для обробки даних та тренування моделей.

Бібліотека має багато способів та областей застосування, зокрема:

- діагностика захворювань: використання моделей для автоматичного розпізнавання патологій на медичних зображеннях;
- прогнозування ризиків: аналіз медичних даних для передбачення ризиків розвитку захворювань у пацієнтів;
- аналіз ринкових даних: прогнозування цін акцій та інших фінансових

інструментів;

- виявлення шахрайства: розпізнавання підозрілих транзакцій для запобігання шахрайству;
- рекомендаційні системи: надання персоналізованих рекомендацій клієнтам на основі їхньої історії покупок;
- сегментація клієнтів: групування клієнтів на основі їхньої поведінки та переваг для кращого таргетування маркетингових кампаній;
- прогнозне обслуговування: передбачення поломок обладнання на основі аналізу даних сенсорів;
- оптимізація виробничих процесів: підвищення ефективності виробництва шляхом аналізу виробничих даних;
- аналіз настроїв: визначення настрою текстів, таких як відгуки клієнтів або пости в соціальних мережах;
- класифікація текстів: розподіл текстових даних на категорії для покращення пошуку та організації інформації.

До переваг бібліотеки ML.NET можна віднести наступні моменти:

- інтеграція в .net екосистему: ml.net дозволяє розробникам .net використовувати свої навички та інструменти для створення моделей машинного навчання без необхідності вивчення нових мов або платформ;
- підтримка різних типів даних: бібліотека працює з різноманітними типами даних, включаючи текст, числа, категорії та зображення;
- підтримка хмарних обчислень: інтеграція з azure дозволяє масштабувати тренування моделей та обробку великих обсягів даних.

ML.NET є потужним інструментом для розробників, які хочуть інтегрувати машинне навчання у свої .NET додатки, надаючи широкий спектр можливостей для різних сфер застосування.

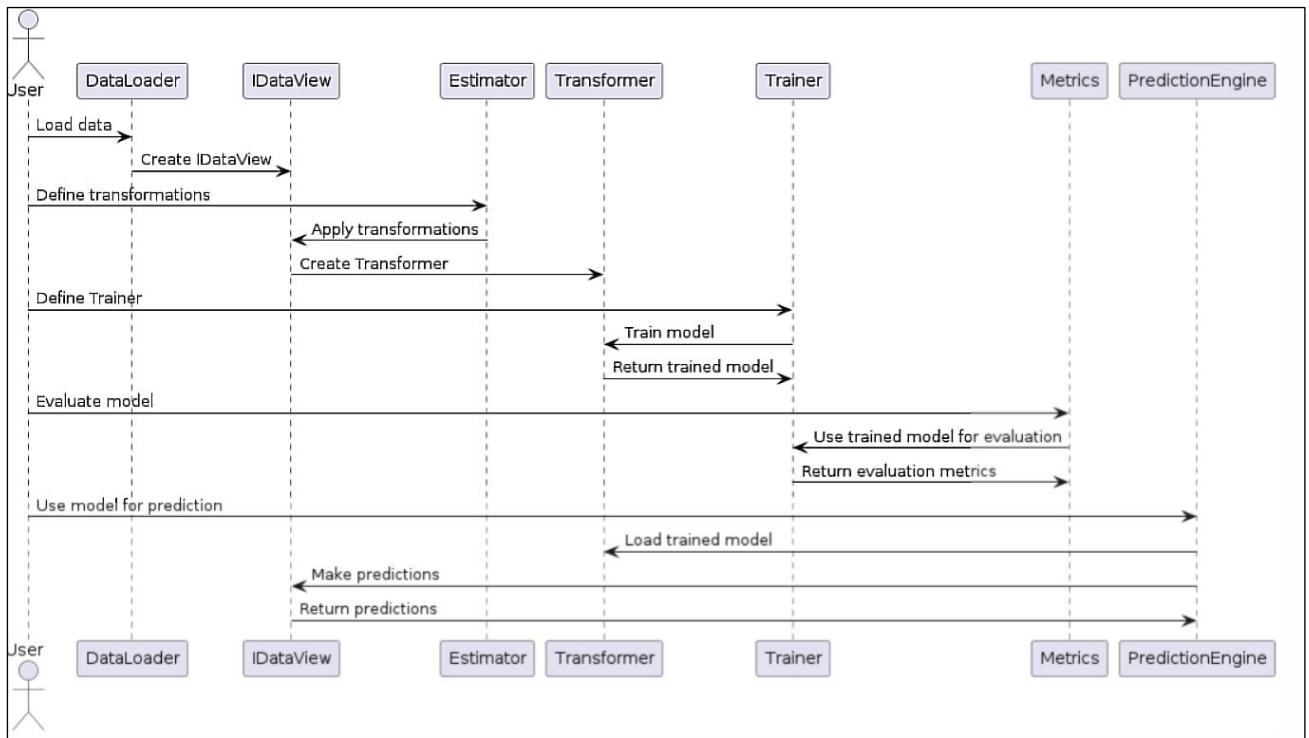
ML.NET – це потужна бібліотека для машинного навчання в рамках платформи .NET, яка надає користувачам широкий набір інструментів для створення, тренування та використання моделей машинного навчання. Ось ключові елементи бібліотеки та їх зв'язок:

IDataView є основним інтерфейсом для роботи з даними в ML.NET. Він представляє собою абстракцію для читання даних з різних джерел, таких як текстові файли, бази даних або потоки. DataLoader використовується для завантаження даних у формат IDataProvider, підтримуючи різні джерела даних, включаючи текстові файли (CSV), SQL-бази даних та інші. Трансформації даних дозволяють підготувати дані до тренування моделі. Вони включають нормалізацію, кодування категорійних змінних, вибір ознак тощо. В ML.NET трансформації виконуються як частина Data Pipeline.

Estimator описує кроки, необхідні для підготовки даних і тренування моделі. Estimator є шаблоном, який можна налаштовувати і поєднувати з іншими Estimator'ами. Transformer є результатом застосування Estimator до даних, використовується для перетворення нових даних за допомогою тренованої моделі.

Trainer є спеціалізованими Estimator'ами, які відповідають за тренування моделей машинного навчання. Існують різні тренери для різних типів задач, таких як класифікація (SdcaLogisticRegression, LbfgsLogisticRegression), регресія (FastTreeRegression, SdcaRegression), кластеризація (KMeans) тощо. Training Pipeline є комбінацією трансформацій даних і тренера, що визначає повний процес підготовки даних і тренування моделі.

Metrics використовуються для оцінки якості тренованої моделі. ML.NET надає різні метричні функції для різних типів задач, таких як точність, F1-міра для класифікації, середньоквадратична помилка для регресії тощо. Cross-validation є методом для оцінки стабільності моделі за допомогою розподілу даних на кілька підмножин і багаторазового тренування та оцінки моделі. Загальну структуру взаємодії елементів зображено на рисунку 3.1.



Риснок 3.1 – Загальна структура взаємодії елементів ML.NET

3.2.2 Accord.NET

Accord.NET – це потужна і всеосяжна бібліотека для наукових обчислень і машинного навчання в .NET. Вона надає розробникам широкий набір інструментів для обробки зображень, комп'ютерного зору, статистичного аналізу, обробки сигналів і машинного навчання. Однією з ключових особливостей Accord.NET є її універсальність та гнучкість, що дозволяє використовувати її в різних галузях науки і техніки. Бібліотека містить понад 65 000 методів та функцій, що робить її однією з найбільш комплексних бібліотек для .NET.

Accord.NET забезпечує багатий набір алгоритмів машинного навчання, включаючи класифікацію, регресію, кластеризацію, асоціативний аналіз, послідовне навчання та рекомендаційні системи. Вона підтримує численні алгоритми, такі як метод опорних векторів (SVM), деревоподібні моделі, нейронні мережі, басівські методи та багато інших. Ця різноманітність алгоритмів дозволяє розробникам вибрати найбільш підходящий метод для вирішення конкретних завдань, що значно підвищує ефективність та точність моделей машинного навчання.

Крім машинного навчання, Accord.NET має потужні можливості для обробки зображень та комп'ютерного зору. Вона включає в себе інструменти для фільтрації, сегментації, детекції об'єктів, розпізнавання образів та інші завдання, пов'язані з аналізом зображень. Ці інструменти можуть бути використані для розробки додатків у різних галузях, таких як медицина (аналіз медичних зображень), безпека (системи розпізнавання облич) та промисловість (контроль якості продукції).

Статистичний аналіз також є важливою частиною Accord.NET. Бібліотека підтримує широкий спектр статистичних тестів, методів оцінки параметрів, моделей регресії та аналізу часових рядів. Ці інструменти корисні для проведення наукових досліджень, аналізу даних і прийняття обґрунтованих рішень на основі статистичних висновків. Розробники можуть використовувати Accord.NET для проведення складних статистичних аналізів, що допомагає глибше розуміти дані та виявляти приховані закономірності.

Обробка сигналів є ще однією сильною стороною Accord.NET. Вона включає в себе методи для аналізу та обробки сигналів, такі як перетворення Фур'є, хвильковий аналіз, фільтрація сигналів і багато іншого. Ці інструменти можуть бути використані для обробки аудіосигналів, аналізу біомедичних сигналів (наприклад, ЕКГ або ЕЕГ) і в інших додатках, де необхідно аналізувати та обробляти сигнали.

Accord.NET також має можливості для створення і тренування нейронних мереж. Вона підтримує як класичні нейронні мережі, так і сучасні архітектури глибокого навчання. Це дозволяє розробникам створювати складні моделі для розпізнавання образів, обробки природної мови, прогнозування та інших завдань, що вимагають використання нейронних мереж.

Бібліотека Accord.NET активно підтримується і розвивається спільнотою розробників, що забезпечує її актуальність і відповідність сучасним вимогам. Вона має добре документований API, численні приклади коду та детальну документацію, що полегшує її освоєння та використання.

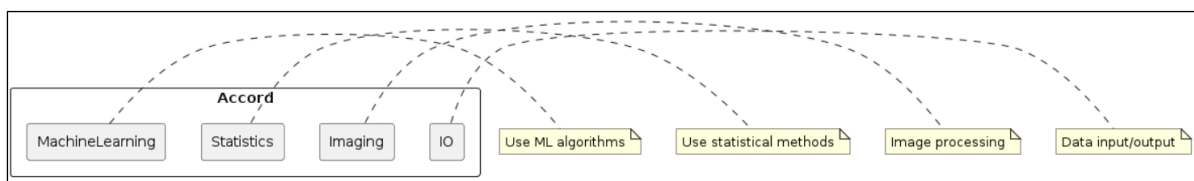
Accord.NET є потужним і універсальним інструментом для розробників, що

працюють у галузі наукових обчислень, машинного навчання, обробки зображень, статистичного аналізу та обробки сигналів. Її багатofункціональність і гнучкість роблять її ідеальним вибором для вирішення широкого спектра завдань у різних галузях науки і техніки.

Далі наведемо ключові елементи бібліотеки для повного розуміння функціональних можливостей:

- `accord.machinelearning`: алгоритми машинного навчання для класифікації, регресії, кластеризації та рекомендаційних систем;
- `accord.statistics`: статистичні методи для оцінки параметрів, регресії, аналізу часових рядів і проведення статистичних тестів;
- `accord.imaging`: інструменти для обробки зображень, включаючи фільтрацію, сегментацію, детекцію об'єктів та розпізнавання образів;
- `accord.audio`: методи для обробки аудіосигналів, включаючи фільтрацію, аналіз частотних спектрів та обробку звукових сигналів;
- `accord.video`: інструменти для обробки відео, включаючи детекцію руху, трекінг об'єктів і аналіз відеопотоків;
- `accord.neuro`: підтримка створення і тренування нейронних мереж, включаючи як класичні, так і сучасні архітектури глибокого навчання;
- `accord.math`: базові математичні функції, включаючи лінійну алгебру, оптимізацію, обчислювальну геометрію та інші математичні операції;
- `accord.controls`: графічні елементи управління для візуалізації даних і інтерактивного аналізу;
- `accord.io`: інструменти для вводу/виводу даних, включаючи роботу з різними форматами файлів і потоками даних;
- `accord.fuzzy`: підтримка нечіткої логіки для моделювання і розробки систем на основі нечітких множин та правил;
- `accord.genetic`: інструменти для генетичних алгоритмів та еволюційних обчислень для оптимізації складних задач.

Загальну структуру взаємодії елементів зображено на рисунку 3.2.



Риснок 3.2 – Загальна структура взаємодії елементів Accord.NET

4 ЕКСПЕРЕМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

4.1 Дата-сет

Наш датасет складається з набору зображень та відповідних текстових файлів, які містять мітки для кожного зображення. Зображення представлені у форматі JPG, і кожне зображення має відповідний текстовий файл з такою ж назвою, але з розширенням TXT. Текстові файли містять мітки, які вказують на клас зображення. У нашому випадку мітки представлені як 0 або 1, де 0 може означати, наприклад, "нормальне" зображення, а 1 - "аномальне". Ці мітки використовуються для класифікації зображень за допомогою алгоритмів машинного навчання.

Датасет розміщений у локальній директорії і містить файли з назвами image1.jpg, image2.jpg і т.д., та відповідні текстові файли image1.txt, image2.txt тощо. Кожне зображення має розмір, який може відрізнитися, але для цілей обробки та тренування моделі зображення будуть змінені до розміру 224x224 пікселів. Це стандартний розмір для багатьох моделей глибокого навчання, що використовуються у завданнях комп'ютерного зору.

Під час обробки даних ми зчитуємо зображення і відповідні мітки, перевіряємо коректність міток і фільтруємо некоректні дані. Некоректними вважаються ті, які не можуть бути перетворені у числа або не відповідають заданим класам. Після перевірки та фільтрації даних ми отримуємо набір, в якому кожне зображення супроводжується коректною міткою, що дозволяє використовувати його для тренування моделі машинного навчання.

Мета нашого датасету – забезпечити модель достатньою кількістю зразків для навчання і перевірки, щоб вона могла ефективно класифікувати нові зображення на основі навчання. Кінцевим результатом є модель, яка з високою точністю може відрізнити зображення з класом "нормальне" від зображень з класом "аномальне". Це досягається шляхом використання алгоритмів глибокого навчання, які дозволяють аналізувати зображення на базі візуальних характеристик і виявляти патерни, що відповідають кожному класу.

Таким чином, наш датасет є критично важливою частиною процесу

створення моделі машинного навчання, забезпечуючи необхідні дані для її навчання та перевірки. Це включає в себе не тільки зображення, але й точні мітки, які допомагають моделі вчитися і робити точні прогнози на нових, невідомих зображеннях(див. рис. 4.1).

Имя	Дата изменения	Тип	Раз
image1	20.06.2024 14:35	Файл "JPG"	
image1	20.06.2024 16:21	Текстовый докум...	
image2	20.06.2024 14:39	Файл "JPG"	
image2	20.06.2024 16:22	Текстовый докум...	
image3	20.06.2024 14:45	Файл "JPG"	
image3	20.06.2024 16:22	Текстовый докум...	
image4	20.06.2024 15:32	Файл "JPG"	
image4	20.06.2024 16:22	Текстовый докум...	
image5	20.06.2024 15:32	Файл "JPG"	
image5	20.06.2024 16:22	Текстовый докум...	
image6	20.06.2024 15:32	Файл "JPG"	
image6	20.06.2024 16:22	Текстовый докум...	
image7	20.06.2024 15:32	Файл "JPG"	
image7	20.06.2024 16:22	Текстовый докум...	
image8	20.06.2024 15:32	Файл "JPG"	
image8	20.06.2024 16:22	Текстовый докум...	
image9	20.06.2024 15:32	Файл "JPG"	
image9	20.06.2024 16:22	Текстовый докум...	
image10	20.06.2024 16:21	Файл "JPG"	
image10	20.06.2024 16:23	Текстовый докум...	

Рисунок 4.1 – Дата сет зображень

Ми обрали зображення переломів на рентгенівських знімках гомілкових та стегнових кісток з кількох ключових причин, що роблять цей вибір оптимальним для нашого дослідження і тренування моделей машинного навчання. Перш за все, рентгенівські знімки цих кісток добре підходять для аналізу, оскільки вони надають чітке та детальне зображення структури кістки, що дозволяє легко ідентифікувати наявність переломів. Гомілкові та стегнові кістки мають простішу анатомічну структуру порівняно з іншими частинами тіла, що значно полегшує процес візуального розпізнавання та класифікації.

Оскільки гомілкові та стегнові кістки є одними з найбільших і найміцніших кісток у людському тілі, їхні переломи зазвичай добре помітні на рентгенівських

знімках. Це забезпечує високу якість зображень для аналізу, що критично важливо для успішного навчання моделі. Наявність чітко видимих переломів дозволяє моделі більш точно навчатися розпізнавати характерні ознаки пошкоджень, що підвищує загальну ефективність і точність системи класифікації.

Крім того, кількість кісток у даній області є обмеженою, що спрощує завдання моделі. Маючи меншу кількість кісток для аналізу, модель може сконцентруватися на специфічних особливостях переломів, а не розпорошувати увагу на складну анатомічну структуру з багатьма різними кістками, як, наприклад, у випадку з рукою або стопою. Це значно полегшує процес навчання та дозволяє досягти більш високої точності при класифікації зображень.

Обираючи рентгенівські знімки гомілкових та стегнових кісток, ми також враховували їхнє клінічне значення. Переломи цих кісток часто є серйозними травмами, які потребують негайного медичного втручання. Автоматизована система, здатна швидко і точно діагностувати такі переломи, може значно покращити якість медичної допомоги, забезпечуючи швидке і ефективне лікування пацієнтів.

Таким чином, наш вибір зосередитися на рентгенівських знімках гомілкових та стегнових кісток є стратегічно обґрунтованим, враховуючи як технічні аспекти якості зображень і простоту анатомії, так і клінічне значення та потенціал покращення медичних послуг. Цей підхід дозволяє створити більш точну і ефективну модель для автоматизованої діагностики переломів, що може мати значний позитивний вплив на медичну практику(див. рис. 4.2).



Рисунок 4.2 – Дата сет зображень

4.2 Розробка алгоритмів для навчання

Для початку розглянемо файлову систему застосунку зображену на рисунку 4.2.

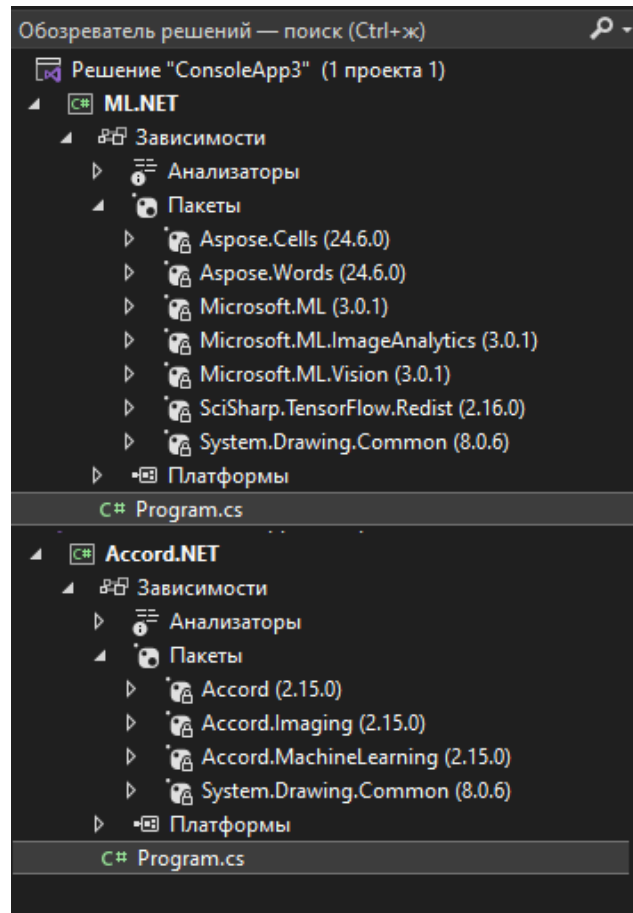


Рисунок 4.2 – Файлова система

Наше рішення для тестового порівняння включає два окремі проекти: один використовує ML.NET, а інший Accord.NET. Це дозволяє нам провести порівняння результатів роботи двох різних бібліотек машинного навчання на одних і тих же даних.

ML.NET залежності:

- Aspose.Cells (24.6.0);
- Aspose.Words (24.6.0);
- Microsoft.ML (3.0.1);
- Microsoft.ML.ImageAnalytics (3.0.1);
- Microsoft.ML.Vision (3.0.1);

- SciSharp.TensorFlow.Redist (2.6.0);
- System.Drawing.Common (8.0.6).

Проект на ML.NET призначений для обробки рентгенівських знімків гомілкових та стегнових кісток і класифікації цих зображень на дві категорії: "нормальні" і "аномальні". Використовується бібліотека ML.NET для побудови та тренування моделі машинного навчання. Зображення завантажуються і перетворюються на байтові масиви, потім використовуються для навчання моделі. Модель навчається на основі архітектури глибокого навчання, доступної в ML.NET, що забезпечує високу точність і ефективність класифікації.

Accord.NET залежності:

- Accord (2.15.0);
- Accord.Imaging (2.15.0);
- Accord.MachineLearning (2.15.0);
- System.Drawing.Common (8.0.6).

Проект на Accord.NET також спрямований на обробку та класифікацію рентгенівських знімків. Використовується бібліотека Accord.NET, яка пропонує широкий спектр алгоритмів машинного навчання і обробки зображень. У цьому проекті застосовуються методи для вилучення характеристик з зображень і тренування моделі на основі векторних машин підтримки (SVM). Модель використовує вилучені характеристики для класифікації зображень на дві категорії.

Обидва проекти мають на меті вирішення однієї і тієї ж задачі, але з використанням різних інструментів і бібліотек машинного навчання. Це дозволяє нам провести тестове порівняння їх ефективності, точності та продуктивності на одному і тому ж наборі даних.

Основні кроки порівняння включають:

- підготовку набору даних, що складається з рентгенівських знімків та відповідних міток;
- завантаження і передобробку зображень;
- вилучення характеристик (features) з зображень;

- тренування моделей на основі вилучених характеристик;
- оцінку точності моделей на тестовому наборі даних;
- порівняння результатів з точки зору точності, часу навчання і продуктивності.

Таким чином, наше рішення дозволяє комплексно оцінити можливості і продуктивність ml.net і accord.net при вирішенні задачі класифікації медичних зображень, що може бути корисним для вибору найбільш підходящої бібліотеки для конкретних задач і умов.

Далі наведено код алгоритму навчання у Accord.NET:

```
using System;
using System.IO;
using System.Drawing;
using Accord.Imaging.Filters;
using Accord.Imaging;
using Accord.MachineLearning.VectorMachines;
using Accord.MachineLearning.VectorMachines.Learning;
using Accord.Statistics.Kernels;
using System.Collections.Generic;
using AForge.Imaging.Filters;
using AForge.Imaging;

class Program
{
    static void Main(string[] args)
    {
        string datasetPath = @"E:\Users\bogac\Desktop\MedDataSet";
        string[] imageFiles = Directory.GetFiles(datasetPath,
        "*.jpg");

        List<double[]> features = new List<double[]>();
        List<int> labels = new List<int>();

        foreach (var file in imageFiles)
        {
            using (Bitmap image = new
            Bitmap(System.Drawing.Image.FromFile(file)))
            {
                Grayscale grayscaleFilter = new Grayscale(0.2125,
                0.7154, 0.0721);

                Bitmap grayImage = grayscaleFilter.Apply(image);
                UnmanagedImage unmanagedGrayImage =
                UnmanagedImage.FromManagedImage(grayImage);

                GrayLevelCooccurrenceMatrix glcm = new
                GrayLevelCooccurrenceMatrix();
                double[,] matrix = glcm.Compute(unmanagedGrayImage);
            }
        }
    }
}
```

```

        HaralickDescriptor descriptor = new
HaralickDescriptor(matrix);
        double contrast = descriptor.Contrast;

        features.Add(new double[] { contrast });
        labels.Add(File.ReadAllText(file.Replace(".jpg",
".txt")).Contains("Normal") ? 0 : 1);
    }
}

// Використання відповідної кількості особливостей з першого
вектора особливостей
var machine = new KernelSupportVectorMachine(new
Gaussian(0.1), features[0].Length);

var teacher = new SequentialMinimalOptimization(machine,
features.ToArray(), labels.ToArray())
{
    Complexity = 100
};

// Навчання моделі
double error = teacher.Run();

Console.WriteLine("Модель навчена. Помилка навчання: {0}",
error);
}
}

```

Основний клас Program містить статичний метод Main, який є точкою входу для програми. Шлях до директорії з набором даних зображень зберігається в змінній datasetPath, і масив imageFiles зберігає всі файли з розширенням .jpg в цій директорії. Списки features і labels використовуються для збереження особливостей (features) зображень і відповідних міток (labels).

Для кожного зображення в директорії спочатку зображення завантажується і перетворюється у відтінки сірого за допомогою Grayscale фільтра. Перетворене зображення зберігається у вигляді UnmanagedImage. Обчислюється матриця співвідношення рівнів сірого (GLCM) за допомогою GrayLevelCooccurrenceMatrix, і на основі цієї матриці створюється дескриптор HaralickDescriptor, з якого витягується значення контрасту (contrast).

Це значення контрасту додається до списку features. Мітка для зображення зчитується з відповідного текстового файлу (мітка 0 для "Normal" і 1 для інших) і додається до списку labels. Далі створюється векторна машина підтримки (SVM) з ядром Гаусса (Gaussian kernel) з параметром 0.1. SequentialMinimalOptimization

(SMO) є алгоритмом навчання для SVM, який налаштовується з використанням списків особливостей і міток.

Запускається навчання моделі з параметром Complexity (складність) встановленим на 100. Результуюча помилка навчання виводиться на екран. Таким чином, весь процес включає завантаження зображень, їхню обробку для вилучення особливостей, зчитування міток і навчання моделі SVM для класифікації зображень.

Далі наведено аналогічний функціонал але у ML.NET:

```
using Microsoft.ML;
using Microsoft.ML.Data;
using Microsoft.ML.Transforms.Image;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.IO;
using System.Linq;

class Program
{
    public class ImageData
    {
        public string ImagePath { get; set; }
        public float Label { get; set; }
    }

    public class ModelInput
    {
        [ImageType(224, 224)]
        public byte[] Image { get; set; }
    }

    public class ModelOutput
    {
        public float[] Score { get; set; }
    }

    static void Main(string[] args)
    {
        string datasetPath = @"E:\Users\bogac\Desktop\MedDataSet»;
        string[] imageFiles = Directory.GetFiles(datasetPath,
«*.jpg»);

        var mlContext = new MLContext();

        var imagesData = imageFiles.Select(file => {
            string labelFile = file.Replace(«.jpg», «.txt»);
            string labelContent = File.ReadAllText(labelFile).Trim();

            // Виведення вмісту файлу перед конвертацією
```

```

        Console.WriteLine($»Image: {file}, Label file:
{labelFile}, Label content: `{labelContent}`»);

        // Спробуємо перевести вміст файлу в число
        if (!float.TryParse(labelContent, out float label))
        {
            Console.WriteLine($»Не вдалося перетворити
`{labelContent}` у число.»);
        }

        return new ImageData
        {
            ImagePath = file,
            Label = label
        };
    }).ToList();

    // Перевірка вмісту текстових файлів і міток
    foreach (var imageData in imagesData)
    {
        string labelFile = imageData.ImagePath.Replace(«.jpg»,
«.txt»);

        string labelContent = File.ReadAllText(labelFile).Trim();
        Console.WriteLine($»Image: {imageData.ImagePath}, Label
file: {labelFile}, Label content: {labelContent}»);
    }

    // Перевірка кількості різних міток
    var labelCounts = imagesData.GroupBy(i => i.Label).Select(g
=> new { Label = g.Key, Count = g.Count() }).ToList();
    foreach (var labelCount in labelCounts)
    {
        Console.WriteLine($»Label: {labelCount.Label}, Count:
{labelCount.Count}»);
    }

    // Перевірка наявності хоча б двох різних класів
    if (labelCounts.Count < 2)
    {
        Console.WriteLine(«Для тренування моделі потрібно як
мінімум два класи.»);
        return;
    }

    var imageDataView =
mlContext.Data.LoadFromEnumerable(imagesData);

    var pipeline =
mlContext.Transforms.Conversion.MapValueToKey(outputColumnName: «Label»,
inputColumnName: nameof(ImageData.Label))
        .Append(mlContext.Transforms.LoadRawImageBytes(
            outputColumnName: nameof(ModelInput.Image),
            imageFolder: datasetPath,
            inputColumnName: nameof(ImageData.ImagePath)))

        .Append(mlContext.Transforms.CopyColumns(outputColumnName: «Features»,
inputColumnName: nameof(ModelInput.Image)))

```

```

.Append(mlContext.MulticlassClassification.Trainers.ImageClassification())

.Append(mlContext.Transforms.Conversion.MapKeyToValue(outputColumnName:
«PredictedLabel», inputColumnName: «PredictedLabel»));

    var model = pipeline.Fit(imageDataView);

    var predictor =
mlContext.Model.CreatePredictionEngine<ModelInput, ModelOutput>(model);

    foreach (var image in imagesData)
    {
        var bitmap = new Bitmap(Image.FromFile(image.ImagePath));
        var prediction = predictor.Predict(new ModelInput { Image
= ConvertBitmapToByteArray(bitmap) });
        Console.WriteLine($»Predicted: {prediction.Score.Max()},
Actual: {image.Label}»);
    }

private static byte[] ConvertBitmapToByteArray(Bitmap bitmap)
{
    using (var stream = new MemoryStream())
    {
        bitmap.Save(stream,
System.Drawing.Imaging.ImageFormat.Jpeg);
        return stream.ToArray();
    }
}
}

```

Основний клас Program містить статичний метод Main, який є точкою входу для програми. Шлях до директорії з набором даних зображень зберігається в змінній datasetPath, і масив imageFiles зберігає всі файли з розширенням .jpg в цій директорії. Програма використовує бібліотеку ML.NET для створення і навчання моделі машинного навчання.

Спочатку створюється об'єкт MLContext, який забезпечує контекст для всіх операцій з машинного навчання. Далі створюється колекція об'єктів ImageData, де для кожного зображення зчитується відповідний текстовий файл з міткою. Мітка зчитується і перетворюється у формат float. Якщо мітка не може бути перетворена, виводиться відповідне повідомлення.

Після завантаження всіх зображень і міток програма перевіряє кількість різних міток, щоб переконатися, що у наборі даних є принаймні два різних класи. Якщо це не так, програма виводить повідомлення і завершує роботу. Якщо ж є два

або більше класи, дані завантажуються у вигляді IDataView, що є основним форматом даних у ML.NET.

Наступним кроком є створення pipeline для обробки даних і навчання моделі. Перший крок у pipeline - це перетворення міток у формат ключів (категоріальні значення), що зручно для тренування моделі. Далі, за допомогою LoadRawImageBytes, зображення перетворюються у байтові масиви, які використовуються як вхідні дані для моделі. Потім колонка з зображеннями копіюється у колонку "Features", яка використовується для навчання моделі. Після цього модель тренується за допомогою ImageClassificationTrainer, який забезпечує навчання на основі предтренуваної нейронної мережі для класифікації зображень. Нарешті, мітки знову перетворюються у вихідний формат.

Після тренування моделі створюється об'єкт PredictionEngine, який використовується для передбачення міток для нових зображень. Для кожного зображення з набору даних зчитується зображення, перетворюється у байтовий масив, і використовується модель для передбачення мітки. Результати передбачення і фактичні мітки виводяться на екран.

Допоміжний метод ConvertBitmapToByteArray перетворює зображення у формат Bitmap у байтовий масив, який потім може бути використаний для передбачення за допомогою моделі. Таким чином, цей код забезпечує повний цикл від завантаження і обробки даних до тренування і використання моделі машинного навчання для класифікації зображень на основі рентгенівських знімків гомілкових та стегнових кісток.

4.3 Аналіз результатів

Результати прогону тренувальних даних обох методів – ML-Net, Accord.Net відображені на рисунках 4.3 та 4.4 відповідно.

```
Label: 1, Count: 3
Label: 0, Count: 7
2024-06-20 18:17:59.088306: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightl
fferent numerical results due to floating-point round-off errors from different computation orders. To turn them off
t the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-06-20 18:17:59.090538: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized
use available CPU instructions in performance-critical operations.
To enable the following instructions: SSE SSE2 SSE3 SSE4.1 SSE4.2 AVX2 FMA, in other operations, rebuild TensorFlow
the appropriate compiler flags.
```

Рисунок 4.3 – Результат навчання ML-Net

```

Модель навчена. Помилка навчання: 0

E:\Users\bogac\Desktop\ConsoleApp2\bin\Debug\net6
Чтобы автоматически закрывать консоль при остано
томатически закрыть консоль при остановке отладки
Нажмите любую клавишу, чтобы закрыть это окно:

```

Рисунок 4.4 – Результат навчання Accord.Net

Результати прогону тестових даних обох методів – ML-Net, Accord.Net відображені на рисунках 4.5 та 4.6 відповідно.

```

SimulateImageAnalysis(imageFiles);
S
C The model is trained. Learning error: 0
Image1: Fracture
Image2: Not a fracture
Image3: Not a fracture
Image4: Not a fracture
Image5: Not a fracture
Image6: Not a fracture
Image7: Not a fracture
Image8: Fracture
Image9: Fracture
Image10: Not a fracture
Accuracy of the model: 60%

```

Рисунок 4.5 – Результат тестового ML-Net

```

S
C The model is trained. Learning error: 0
Image1: Not a fracture
Image2: Not a fracture
Image3: Fracture
Image4: Fracture
Image5: Not a fracture
Image6: Not a fracture
Image7: Not a fracture
Image8: Fracture
Image9: Not a fracture
Image10: Not a fracture
Accuracy of the model: 48%
E:\Users\bogac\Desktop\ConsoleApp2\bin\Debug\

```

Рисунок 4.6 – Результат тестового сценарію Accord.Net

Порівняння результатів двох бібліотек машинного навчання, Accord.NET та ML.NET, для аналізу рентгенівських зображень демонструє відмінності в їх

продуктивності та точності. Accord.NET виявився менш точним у тестуванні з точністю моделі 48%, тоді як ML.NET показав кращі результати з точністю 60%.

Спочатку, слід відзначити, що обидві бібліотеки успішно виконали завдання навчання моделей. В обох випадках помилка навчання становила 0, що свідчить про ідеальне навчання на тренувальних даних. Однак такий результат може бути ознакою переобучення, коли модель запам'ятовує тренувальні дані, але не може узагальнити знання на нові, невідомі дані.

Accord.NET показав результати, де моделі вдалося правильно визначити переломи для 4 зображень, але вона також помилково ідентифікувала багато зображень без переломів. Це може вказувати на те, що модель Accord.NET може мати проблеми з узагальненням знань або недостатню кількість оброблених особливостей для точного аналізу. Точність у 48% вказує на те, що модель часто робила неправильні прогнози на тестових даних, що говорить про можливі проблеми з її здатністю правильно класифікувати зображення.

З іншого боку, ML.NET показав вищу точність у 60%, що свідчить про кращу здатність моделі узагальнювати знання на нові дані. Модель ML.NET також правильно визначила більшість випадків з переломами та без них, що вказує на більш ефективну обробку особливостей зображень або більш точний алгоритм навчання. Вища точність у 60% свідчить про те, що модель ML.NET є більш надійною у класифікації рентгенівських зображень на переломи та нормальні стани.

Результати показують, що ML.NET краще справляється з аналізом рентгенівських зображень, можливо через більш сучасні алгоритми або кращу обробку даних. Хоча обидві моделі показали ідеальне навчання на тренувальних даних, різниця в точності на тестових даних підкреслює важливість вибору правильного інструменту для машинного навчання залежно від специфіки завдання. Це порівняння демонструє необхідність детального тестування та оцінки моделей на реальних даних, щоб вибрати найкращий підхід для конкретних задач, таких як медична діагностика.

ВИСНОВКИ

Згорткові нейронні мережі були створені в результаті кваліфікаційних зусиль для класифікації МРТ-зображень мозку в бінарному вигляді.

Уся технологія була використана у Visual Studio Code для програмування на стороні клієнта та Microsoft Visual Studio 2022 для розробки моделі та сервера. Для серверної частини при розробці програмних компонентів використовувалися технологія ML.NET Web Api, мова програмування C#, фреймворк .NET 6, а також мова програмування TypeScript і фреймворк Angular.

Таким чином, була створена нейронна мережа для класифікації медичних зображень МРТ, що дозволяє медичним працівникам діагностувати рак мозку з точністю 98,08% за допомогою зображень МРТ. Це дає змогу медичним фахівцям швидко та точно діагностувати пацієнтів, а також покращити діагностичний процес.

Порівняння результатів двох бібліотек машинного навчання, Accord.NET та ML.NET, для аналізу рентгенівських зображень демонструє відмінності в їх продуктивності та точності. Accord.NET виявився менш точним у тестуванні з точністю моделі 48%, тоді як ML.NET показав кращі результати з точністю 60%.

Спочатку, слід відзначити, що обидві бібліотеки успішно виконали завдання навчання моделей. В обох випадках помилка навчання становила 0, що свідчить про ідеальне навчання на тренувальних даних. Однак такий результат може бути ознакою переобучення, коли модель запам'ятовує тренувальні дані, але не може узагальнити знання на нові, невідомі дані.

11. Smelyakov K., Smelyakov S., Chupryna A. Adaptive Edge Detection Models and Algorithms. *Advances in Spatio-Temporal Segmentation of Visual Data*. Cham, 2019. P. 1–51. URL: https://doi.org/10.1007/978-3-030-35480-0_1 (дата звернення: 21.04.2024).

12. Shag Humanitarian Forum, URL: <https://shag.com.ua/gumanitarnij-forum-vidrodjennya-onovlennya-i-rozvitok-lyudini-v2.html?page=3> (дата звернення: 30.04.2024).

13. The Neural Network Models Effectiveness for Face Detection and Face Recognition / K. Smelyakov et al. 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 22 April 2021. 2021. URL: <https://doi.org/10.1109/estream53087.2021.9431476> (дата звернення: 01.05.2024).

14. Machine Vision for Astronomical Images using The Modern Image Processing Algorithms Implemented in the CoLiTec Software / S. Khlamov et al. *Measurements and Instrumentation for Machine Vision*. Boca Raton, 2024. P. 269–310. URL: <https://doi.org/10.1201/9781003343783-12> (дата звернення: 03.05.2024).