

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ  
КАФЕДРА БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Владислав ШЛИКОВ

«\_\_» \_\_\_\_\_ 20\_\_р.

**Дипломна робота  
на здобуття ступеня бакалавра  
за освітньо-професійною програмою «Медична інженерія»  
спеціальності 163 «Біомедична інженерія»  
на тему: «Комп'ютерний аналіз МРТ-зображень для оцінки структури  
головного мозку»**

Виконав:

студент IV курсу, групи БМ-92

Радченко Денис Олександрович \_\_\_\_\_

Керівник:

Старший викладач

Данілова Валентина Анатоліївна \_\_\_\_\_

Консультант з охорони праці:

Доцент каф. ОППЦБ, к.т.н., доцент

Демчук Гліб Вікторович \_\_\_\_\_

Рецензент:

Професор кафедри ББЗЛ, професор, доктор мед.наук

Глоба Олександр Петрович \_\_\_\_\_

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

Студент \_\_\_\_\_

Київ – 2023 року

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

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 163 «Біомедична інженерія»

Освітньо-професійна програма «Медична інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_Владислав ШЛИКОВ

«\_\_\_» \_\_\_\_\_2023 р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

**Радченко Денис Олександрович**

1. Тема роботи «Комп'ютерний аналіз МРТ-зображень для оцінки структури головного мозку», керівник роботи Данилова Валентина Анатоліївна, старший викладач затверджені наказом по університету від «31» травня 2023 р. №2106-с
2. Термін подання студентом роботи «9» червня 2023 р.
3. Вихідні дані до роботи: модель класифікації зображень, система розпізнавання хвороби Альцгеймера по МРТ-зображенням.
4. Зміст роботи: дослідження предметної області розпізнавання МРТ-зображень хвороби Альцгеймера
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація
6. Консультанти розділів роботи

| Розділ        | Прізвище, ініціали та посада консультанта      | Підпис, дата   |                  |
|---------------|------------------------------------------------|----------------|------------------|
|               |                                                | завдання видав | завдання прийняв |
| Охорона праці | Демчук Г.В., доцент каф. ОППЦБ, к.т.н., доцент |                |                  |

7. Дата видачі завдання 5 квітня 2023 р.

Календарний план

| № з/п | Назва етапів виконання дипломної роботи             | Термін виконання етапів роботи | Примітка |
|-------|-----------------------------------------------------|--------------------------------|----------|
| 1     | Отримання завдання на ДР                            | 12 квітня 2023 р.              |          |
| 2     | Ознайомлення зі структурою ДР                       | 13 квітня 2023 р.              |          |
| 3     | Огляд літератури за темою ДР                        | 17-28 квітня 2023 р.           |          |
| 4     | Написання теоретичної частини ДР                    | 8-19 травня 2023 р.            |          |
| 5     | Проведення роботи над експериментальною частиною ДР | 21-30 травня 2023 р.           |          |
| 6     | Реалізація нейронних мереж                          | 1-4 червня 2023 р.             |          |
| 7     | Написання розділу з охорони праці                   | 4-5 червня 2023 р.             |          |
| 8     | Оформлення ДР та аналіз отриманих результатів       | 6-8 червня 2023 р.             |          |

Студент

Денис РАДЧЕНКО

Керівник

Валентина ДАНІЛОВА

## РЕФЕРАТ

Тема дипломної роботи: «Комп'ютерний аналіз МРТ-зображень для оцінки структури головного мозку»

Дипломна робота: 78 сторінок, 12 таблиць, 35 рисунків, 3 додатки, 27 джерел.

Актуальність роботи: у останні роки класифікація медичних зображень стала важливою в клінічних задачах лікування хворих. Проте, традиційний метод досяг своїх меж у своїй ефективності. Застосування штучних нейронних мереж, зокрема згорткових нейронних мереж, стало новим методом машинного навчання, який проявив свій потенціал у рішенні різних класифікаційних завдань.

Мета роботи: розробити програмне забезпечення, яке використовує згорткові нейронні мережі для автоматизованого виявлення ранніх стадій хвороби Альцгеймера шляхом аналізу МРТ-зображень мозку пацієнтів.

Задачі дипломної роботи:

1. Провести аналіз методів машинного навчання.
2. Дослідити різні підходи до обробки медичних зображень.
3. Впровадити та навчити згорткову нейронну мережу, використовуючи спеціальні датасети МРТ-зображень головного мозку.
4. Провести серію експериментів для визначення найбільш точної оцінки класифікації зображення.

Використані моделі: для реалізації класифікатора було використано три базові моделі згорткової нейронної мережі– InceptionV3, VGG19 та ResNet50.

Отримані результати: готова система класифікації хвороби Альцгеймера по МРТ-знімкам до одного з трьох класів з точністю 88%.

Ключові слова: магнітно-резонансна томографія, хвороба Альцгеймера, штучні нейронні мережі, машинне навчання, класифікація, розпізнавання.

## ABSTRACT

Thesis topic: «Computer analysis of MRI images for evaluating the structure of the brain»

Diploma thesis: 78 pages, 12 tables, 35 figures, 3 appendices, 27 sources.

Relevance of the work: in recent years, medical image classification has become important in clinical tasks of patient treatment. However, the traditional method has reached its limits in terms of efficiency. The use of artificial neural networks, in particular convolutional neural networks, has become a new method of machine learning that has shown its potential in solving various classification problems.

Purpose: to develop software that uses convolutional neural networks for automated detection of early stages of Alzheimer's disease by analysing MRI images of patients' brains.

Tasks of the thesis:

1. To analyse machine learning methods.
2. Investigate different approaches to medical image processing.
3. Implement and train a convolutional neural network using special datasets of MRI brain images.
4. Conduct a series of experiments to determine the most accurate image classification score.

Models used: three basic models of convolutional neural networks were used to develop the software product: InceptionV3, VGG19, and ResNet50.

Results obtained: a ready-made system for classifying Alzheimer's disease from MRI images into one of three classes with an accuracy of 88%.

Keywords: magnetic resonance imaging, Alzheimer's disease, artificial neural networks, machine learning, classification, recognition.

## ЗМІСТ

|                                                             |    |
|-------------------------------------------------------------|----|
| ПЕРЕЛІК СКОРОЧЕНЬ .....                                     | 8  |
| ВСТУП.....                                                  | 9  |
| РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПОТОЧНОГО СТАНУ ПРОБЛЕМИ .....         | 11 |
| 1.1 Огляд предметної області.....                           | 11 |
| 1.2 Опис хвороби Альцгеймера .....                          | 12 |
| 1.3 Діагностика хвороби Альцгеймера .....                   | 15 |
| 1.4 Аналіз існуючих методів вирішення проблеми .....        | 18 |
| 1.5 Постановка задачі .....                                 | 22 |
| Висновки до розділу 1 .....                                 | 23 |
| РОЗДІЛ 2 МЕТОДИ АНАЛІЗУ МРТ-ЗОБРАЖЕНЬ.....                  | 24 |
| 2.1 Машинне навчання .....                                  | 24 |
| 2.2 Штучна нейронна мережа .....                            | 25 |
| 2.3 Згорткова нейронна мережа .....                         | 29 |
| Висновки до розділу 2 .....                                 | 35 |
| РОЗДІЛ 3 РЕАЛІЗАЦІЯ НЕЙРОННИХ МЕРЕЖ.....                    | 37 |
| 3.1 Програмне забезпечення та використання бібліотек .....  | 37 |
| 3.2 Набір даних .....                                       | 38 |
| 3.3 Проведення експериментів.....                           | 39 |
| 3.4 Аналіз результатів.....                                 | 50 |
| Висновки до розділу 3 .....                                 | 51 |
| РОЗДІЛ 4 ОХОРОНА ПРАЦІ.....                                 | 52 |
| 4.1 Вступ.....                                              | 52 |
| 4.2 Характеристика приміщення та робочого місця .....       | 52 |
| 4.3 Оцінка небезпечних і шкідливих виробничих факторів..... | 55 |
| Висновки до розділу 4 .....                                 | 58 |
| ВИСНОВКИ .....                                              | 59 |
| СПИСОК ЛІТЕРАТУРИ .....                                     | 60 |
| Додаток А .....                                             | 63 |

|           |      |               |        |      |                                                                        |  |  |                           |      |        |    |  |
|-----------|------|---------------|--------|------|------------------------------------------------------------------------|--|--|---------------------------|------|--------|----|--|
|           |      |               |        |      | БМ92.30.3105.2106                                                      |  |  |                           |      |        |    |  |
|           |      |               |        |      |                                                                        |  |  |                           |      |        |    |  |
| Вим       | Лист | №докум.       | Підпис | Дата |                                                                        |  |  |                           |      |        |    |  |
| Розробив  |      | Радченко Д.О. |        |      | Комп'ютерний аналіз МРТ-зображень для оцінки структури головного мозку |  |  | Лім.                      | Лист | Листів |    |  |
| Перевірив |      | Данілова В.А. |        |      |                                                                        |  |  |                           |      | 6      | 78 |  |
| Реценз.   |      | Глоба О.П.    |        |      |                                                                        |  |  | КПІ ім. Ігоря Сікорського |      |        |    |  |
| Затвердив |      | Шликов В.В.   |        |      |                                                                        |  |  | ФБМІ БМ-92                |      |        |    |  |
|           |      |               |        |      |                                                                        |  |  |                           |      |        |    |  |

|                 |    |
|-----------------|----|
| Додаток Б.....  | 71 |
| Додаток В ..... | 75 |

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 7    |

## ПЕРЕЛІК СКОРОЧЕНЬ

МРТ – Магнітно-резонансна томографія

ХА – Хвороба Альцгеймера

МОВ (англ. Support Vector Machines, SVM) – Метод опорних векторів

ЗНМ (англ. Convolutional Neural Networks, CNN) – Згортова нейронна мережа

ШНМ – Штучна нейронна мережа

GCA (англ. Global Cortical Atrophy Scale) – шкала глобальної коркової атрофії

ReLU (англ. Rectified Linear Unit) - Зрізаний лінійний вузол

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   | 8    |
| Изм. | Лист | № докум. | Підпис | Дата |                   |      |



## ВСТУП

Сучасна медицина постійно шукає нові підходи та інструменти для діагностики та оцінки різних захворювань. Одним із викликів, з яким стикаються лікарі та дослідники, є рання діагностика та моніторинг хвороби Альцгеймера, яка є основною причиною деменції. Хоча ця хвороба не є неминучим наслідком старіння, раннє виявлення та початок лікування можуть великою мірою полегшити життя пацієнтів і покращити їхнє майбутнє.

Завдяки значному розвитку області медичних зображень та комп'ютерних технологій з'явилися нові можливості для аналізу та інтерпретації зображень головного мозку. Одним з найбільш потужних інструментів у цій області є штучні нейронні мережі, зокрема згорткові нейронні мережі, які показали вражаючі результати в розпізнаванні образів та аналізі зображень.

Магнітно-резонансна томографія (МРТ) є одним з найбільш поширених методів зображення головного мозку. Вона надає детальну інформацію про структуру та функцію мозку, що робить її важливим засобом для діагностики різних неврологічних захворювань. Однак, інтерпретація та аналіз МРТ-зображень можуть бути складними завданнями для лікарів, оскільки вимагають великої експертизи та часу.

Тому, актуальним завданням є розробка програмного забезпечення, яке використовує згорткові нейронні мережі для автоматичного виявлення ранніх стадій хвороби Альцгеймера шляхом аналізу МРТ-зображень мозку пацієнтів. Такий підхід може допомогти лікарям у виявленні ознак хвороби на ранніх стадіях, коли втрати функцій ще не є суттєвими, та вчасно призначити відповідну терапію.

Ця дипломна робота присвячена розробці та реалізації програмного забезпечення, яке поєднує передові технології штучного інтелекту та медичного обладнання для аналізу МРТ-зображень головного мозку з метою ранньої діагностики хвороби Альцгеймера. Це дозволить покращити точність та

швидкість діагностики, а також зробить процес аналізу більш об'єктивним та доступним.

Метою дипломної роботи є розробка програмного забезпечення, яке використовує згорткові нейронні мережі для автоматизованого виявлення ранніх стадій хвороби Альцгеймера шляхом аналізу МРТ-зображень мозку пацієнтів.

Для досягнення вказаної мети роботи треба виконати основні завдання:

1. проаналізувати сучасний стан задачі аналізу МРТ-зображень мозку людей з захворюванням Альцгеймера;
2. дослідити існуючі технології для вирішення поставленої задачі;
3. реалізувати штучну нейронну мережу та її налаштування для аналізу МРТ-зображень;
4. навчити та протестувати нейронну мережу.

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 10   |

# РОЗДІЛ 1

## ДОСЛІДЖЕННЯ ПОТОЧНОГО СТАНУ ПРОБЛЕМИ

### 1.1 Огляд предметної області

На сьогоднішній день аналіз даних отриманих за допомогою комп'ютерної томографії (КТ), магнітно-резонансної томографії (МРТ) та позитрон-емісійної томографії (ПЕТ), в основному виконувався спеціалістами, такими як радіологи, рентгенологи та лікарі, в відповідній сфері діагностики. Проте, завдяки поступовому вдосконаленню медичних інструментів та приладів, значно зростала кількість накопичених знань про хвороби, діагностику та їх якість. Останнім часом у медичній практиці щоденно створюється значна кількість цифрових медичних зображень, завдяки цифровій епохі. Ці зображення отримуються під час різних обстежень, таких як рентгенологічні, ультразвукові, ендоскопічні, офтальмологічні тощо. Лікарі потребують цих зображень для діагностики, прогнозування та планування лікування різних хвороб. Тому класифікація медичних зображень відіграє важливу роль у сфері медицини. Проте, лікарі витрачають значні зусилля та час на виділення та вибір ознак для класифікації, що може бути виснажливим. Також важливо враховувати обмеженість людських ресурсів: не завжди є достатньо висококваліфікованих фахівців, а лікарі – це не боги, вони теж можуть припускатися помилок при постановці діагнозів хворим.

Сьогодні згорткові нейронні мережі (ЗНМ) стали одним з найефективніших інструментів для розуміння зображень. Вони перевершили людських експертів у багатьох завданнях, пов'язаних з класифікацією та розпізнаванням зображень, і медицина не є винятком. Застосування згорткових нейронних мереж (ЗНМ) у медицині вже було підтверджено численними дослідженнями. Наприклад, такі дослідження включають автоматичну класифікацію туберкульозу на рентгенограмах легенів [7], розрізнення мас печінки з динамічним контрастом на КТ-знімках [8] та інші. Деякі дослідження,

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 11   |

що використовують ЗНМ для класифікації медичних зображень, досягли показників, які можна порівняти з експертними лікарями у відповідній галузі медицини.

На сьогоднішній день, проблема боротьби з хворобою Альцгеймера набуває все більшого значення по всьому світу. Це прогресуюче невиліковне неврологічне захворювання головного мозку, для якого не існує ліків. Тому раннє діагностування є надзвичайно важливим для запобігання прогресу симптомів. Однак, діагноз ХА на ранніх стадіях часто ставиться з великою важкістю, а точність діагностики в значній мірі залежить від досвіду та кваліфікації рентгенолога. Для досягнення точної діагностики цього захворювання, дослідники розробили кілька комп'ютерних діагностичних систем, що базувались на експертних правилах у період з 1970-х до 1990-х років. У цих системах використовувалися вектори ознак з медичних зображень для навчання контрольованих систем. Однак, виникала потреба у використанні експертів, що вимагало значних зусиль, часу та коштів [9]. Тому сучасні дослідники активно працюють над розробкою моделей штучних нейронних мереж для точної діагностики ХА і вже досягли значних результатів. Наприклад, для діагностики хвороби Альцгеймера за допомогою 2D зрізів МРТ була розроблена згорткова нейронна мережа під назвою AlzNet, яка має точність 99,3% [10]. Однак, всі ці моделі ще потребують додаткової вдосконаленості.

Отже, діагностика хвороби Альцгеймера за допомогою штучних нейронних мереж широко розвивається та отримує великий розгляд. Знаходження нових підходів до класифікації та розпізнавання ХА на знімках МРТ за допомогою нейронних мереж є актуальним завданням у сучасній медицині.

## 1.2 Опис хвороби Альцгеймера

Хвороба Альцгеймера (ХА) – це невиліковна хронічна нейродегенеративна патологія, яка вперше була описана німецьким лікарем Алоїсом Альцгеймером

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 12   |

у 1906 році. Ця хвороба пов'язана з поступовою загибеллю нейронів головного мозку, зокрема в області гіпокамп та коркової речовини, що призводить до різних порушень когнітивних та психічних функцій. Основні причини ХА пов'язані з накопиченням білкових агрегатів (амілоїду та тау-білка) у мозку, що сприяє запальним процесам та дисфункції мітохондрій. На сьогоднішній день, деталізація патогенезу ХА ще потребує подальшого вивчення. Хвороба Альцгеймера характеризується трьома основними стадіями - ранньою (м'якою), середньою (помірною) і пізньою (важкою). Іноді також виокремлюють доклінічну стадію і стадію помірно-важкої деменції (рис. 1.1) [12].



Рисунок 1.1 – Зображення мозку людини в нормі та при ураженні ХА

Вчені розрізняють два клінічні типи хвороби Альцгеймера: ранній початок (тип 2) і пізній початок (тип 1). Хвороба раннього початку зазвичай розвивається в передстаречому віці і характеризується поступовим прогресуванням порушень пам'яті, інтелектуальної діяльності, мовних функцій, координації рухів та просторової діяльності. Хвороба пізнього початку найчастіше виникає в старечому віці і починається з незначних порушень пам'яті та зниження інтелектуальних здібностей, які в кінцевому підсумку призводять до зниження вищих кіркових функцій. Також існує атипова форма хвороби Альцгеймера, в якій спостерігається поєднання симптомів, характерних для хвороби Альцгеймера і судинної деменції. Хоча існує кілька теорій, які намагаються пояснити причину розвитку хвороби Альцгеймера, це питання залишається відкритим [11].

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 13   |

Можливі причини хвороби Альцгеймера включають спадкову схильність, атеросклероз, кисневу недостатність, відкладення бета-амілоїду в клітинах головного мозку, зниження рівня ацетилхоліну, хромосомні порушення, реакції імунітету, вплив токсичних речовин, травми (психологічні, черепно-мозкові) та віруси. Також існує припущення про заразність хвороби Альцгеймера на білковому рівні. Фактори ризику захворювання включають повільну депресію, високий артеріальний тиск, низький інтелект, цукровий діабет та інші[12].

Клінічні прояви хвороби Альцгеймера включають зниження пам'яті, уваги, розумової працездатності та здатності до формування і використання рухових навичок та інші симптоми. Серед симптомів хвороби можуть бути погіршення запам'ятовування нової та раніше засвоєної інформації, порушення інтелектуальної діяльності, афазії (порушення мовлення), апраксії (порушення цілеспрямованих рухів і дій) і агнозії (порушення сприйняття: зорового, слухового, тактильного).

Хвороба Альцгеймера розподіляється на три основні стадії: ранню (м'яку), середню (помірну) і пізню (важку), а також може існувати доклінічна стадія і стадія помірно-важкої деменції.

Признаки хвороби Альцгеймера відрізняються залежно від стадій розвитку захворювання. Нижче наведено переформульовані ознаки захворювання на кожній стадії:

1. Доклінічна стадія: Початкові ознаки включають зниження інтелектуальної діяльності, легку забудькуватість, труднощі з визначенням часу та неповне відтворення подій.
2. Стадія м'якої деменції: Симптоми включають порушення пам'яті на поточні події, складності в хронологічному та географічному орієнтуванні, труднощі в розумових процесах, порушення абстрактного мислення, зниження здатності узагальнювати й порівнювати, неможливість самостійно, без помилок здійснювати фінансові операції, подорожувати та вести кореспонденцію.

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 13   |

3. Стадія помірної деменції: Характеризується амнестичним синдромом, включаючи втрату пам'яті на події теперішнього часу й недавнього минулого. Зберігаються спогади про дитинство і молодість, з'являються вигадані спогади. Також спостерігаються часова і просторова дезорієнтація, підвищена тривожність, розгубленість, емоційна нестійкість, і хворі потребують постійної допомоги і підтримки.
4. Стадія важкої деменції: Проявляється глибоким розладом пам'яті до повної втрати, втратою уявлень про час і навколишнє оточення, відсутністю самоідентифікації, втратою здатності до умовиводів і суджень. Люди з такими порушеннями потребують постійного догляду.
5. Заключна стадія важкого недоумства: Характеризується повною руйнацією психічної діяльності, важкими неврологічними розладами, примітивними рефlekсами, автоматизмом у діях, насильницькими гримасами плачу і сміху, розвитком змушених "ембріональних" поз і епілептичних припадків.

На кожному етапі розвитку хвороби Альцгеймера можуть спостерігатися агресія, крики, гіперактивність, збудження, занепокоєння, спроби втечі з дому, сексуальна розторможеність, марення тощо. Захворювання також супроводжується соціальною та професійною неспроможністю. Тривалість хвороби від перших симптомів до смерті може становити від 2 до 20 років (в середньому 12 років). Найбільш серйозним наслідком хвороби Альцгеймера є аспіраційна пневмонія - запалення легенів, що виникає в результаті вдихання рідин, їжі, рвоти або інших твердих частинок в легені [11].

### 1.3 Діагностика хвороби Альцгеймера

На сьогоднішній день, зображення головного мозку використовуються переважно для виявлення видимих відхилень, що пов'язані з іншими станами, а не хворобою Альцгеймера. Однак, нові методи, які наразі перебувають у клінічних дослідженнях, дозволяють діагностувати саме хворобу Альцгеймера.

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 16   |

1. Магнітно-резонансна томографія (МРТ) є методом, який використовує принцип магнітного резонансу. Під час МРТ дослідження пацієнт розміщується в сильному магнітному полі. Цей метод є абсолютно безпечним, якщо виключені наявність будь-яких магнітних або металевих імплантатів.
2. Комп'ютерна томографія (КТ) використовує рентгенівську технологію для отримання багатьох 2D рентгенівських знімків об'єкта з різних ракурсів. За допомогою комп'ютерних програм ці знімки обробляються для створення 3D зображення. Варто зазначити, що КТ не є повністю безпечним методом, оскільки рентгеновське випромінювання має іонізуючі властивості. Тому кількість проведених КТ-процедур має бути обмеженою і контролюваною лікарем. В даний час головне застосування КТ полягає в виявленні пухлин, інсультів і травм голови.
3. Позитрон-емісійна томографія (ПЕТ) використовує радіоактивний індикатор низького рівня, щоб виявити особливу функцію в мозку під час сканування. Безпека цього методу залежить від індикатора, який використовується.

Магнітно-резонансна томографія (МРТ) використовує радіохвилі для орієнтації протонів у тілі пацієнта за допомогою прядильних полів. Ця техніка не використовує іонізуючого випромінювання і дозволяє отримувати тривимірні зображення на будь-якій глибині та будь-якому напрямку [13].

При використанні методу магнітно-резонансної томографії (МРТ), проводять сканування у трьох площинах: аксіальній, корональній та сагітальній. При візуалізації МРТ зображень при хворобі Альцгеймера найчастіше використовуються аксіальні та корональні площини сканування.

На рисунку 1.2 можна побачити зображення МРТ корональної та аксіальної площин мозку здорової людини та людини з ХА.



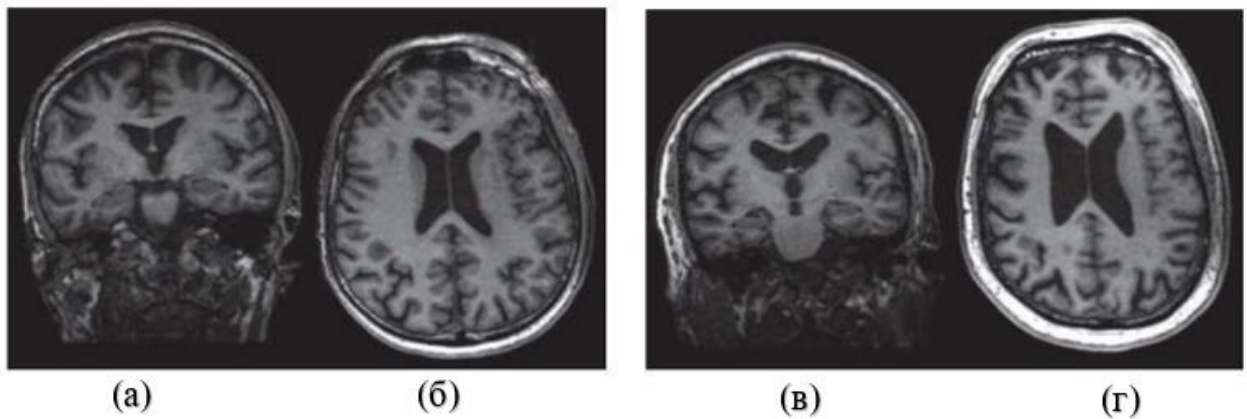


Рисунок 1.2 – Зображення мозку здорової людини (а – корональній площині, б – аксіальній площині) та з ХА (в, г відповідно)

Зі знімків добре видно структурні біомаркери змін мозку людини, а саме:

- атрофію лобкових та тем'яних часток мозку;
- атрофію мезіальної скроневої частки.

Вимірювання ступеня атрофії на основі зображень МРТ в даний час вважають достовірним показником стану і прогресування хвороби. Атрофія є необхідним і невідпинно прогресуючим проявом нейродегенерації. Втрати дендритів і нейронів вважаються основними причинами атрофії. Розташування втрати мозкової тканини має сильну кореляцію зі ступенем когнітивного дефіциту [14].

Атрофія лобкових та теменних ділянок мозку є одним з основних маркерів хвороби ХА, що можна спостерігати на МРТ-знімках у площині аксіального сканування. Оцінку атрофії лобкових та теменних ділянок можна здійснити, враховуючи ступінь розширення бічних шлуночків та глибину коркових борозен на цьому рівні. Для цього використовується шкала глобальної коркової атрофії (GCA), яка дозволяє оцінити ступінь атрофії неокортекса. На рисунку 1.3 зображено атрофію неокортекса та розширення бічних шлуночків на знімку МРТ у площині аксіального сканування.

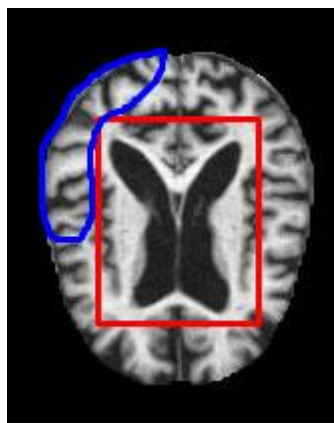


Рисунок 1.3 – Виділення на МРТ-зображенні людини з ХА бічних шлуночків (червоним) та неокортексу (синім)

Шкала GCA (GlobalCorticalAtrophyScale) є інструментом оцінки глобальної коркової атрофії мозку на основі зображень МРТ. Вона дозволяє кваліфікованим лікарям або дослідникам оцінити ступінь атрофії неокортекса (зовнішнього шару кори головного мозку) за певною шкалою. Типова шкала GCA може включати такі ступені атрофії:

1. Відсутність атрофії або мінімальна атрофія: немає помітної втрати коркової маси.
2. Легка атрофія: Помірна втрата коркової маси, але все ще в межах норми.
3. Легка атрофія: Помірна втрата коркової маси, але все ще в межах норми.
4. Помірна атрофія: Середня втрата коркової маси, спостерігаються помітні ознаки атрофії.
5. Важка атрофія: Значна втрата коркової маси, що викликає видимі дефекти та дегенерацію.

Так як дані біомаркери найбільш вагомі для діагностики хвороби Альцгеймера, інші маркери розглядатися не будуть.

#### 1.4 Аналіз існуючих методів вирішення проблеми

У контексті класифікації зображень на декілька класів, існують різні підходи, які можуть бути застосовані. Один з таких методів - класичний підхід машинного навчання. Цей метод базується на використанні статистичних

алгоритмів для класифікації зображень на основі вручну вибраних ознак. Перш за все, для цього методу необхідно визначити певні ознаки зображень, такі як кольорові характеристики, текстурні особливості або геометричні структури. Потім використовуються алгоритми, такі як метод опорних векторів (SVM) або найвний Баєсівський класифікатор, для побудови моделі, яка може розпізнавати та класифікувати зображення на основі цих ознак.

Метод опорних векторів (Support Vector Machines, SVM)[15] є одним з найпоширеніших методів машинного навчання для задач класифікації. Основна ідея полягає у знаходженні гіперплощини, яка найкращим чином розділяє два або більше класи даних (рис. 1.4)[16].

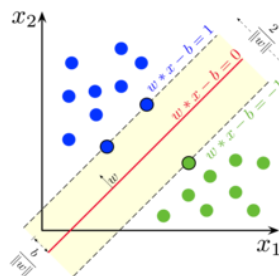


Рисунок 1.4 – Класифікація даних методом опорних векторів (SVM)

Процес роботи методу опорних векторів можна розбити на декілька кроків:

1. Підготовка даних: Спочатку необхідно підготувати навчальний набір даних, в якому кожен зразок має вектор ознак і відповідну мітку класу. Вектор ознак може представляти характеристики зображень, такі як яскравість, текстура або форма.
2. Визначення гіперплощини: Задача полягає в знаходженні гіперплощини, яка максимально розділяє класи даних. Гіперплощина визначається таким чином, щоб максимізувати відстань між найближчими зразками різних класів, відомих як опорні вектори.
3. Класифікація нових зразків: Після знаходження гіперплощини можна використовувати її для класифікації нових зразків. Новий зразок відноситься до одного з класів на основі його положення відносно гіперплощини. Наприклад, якщо зразок знаходиться на одній стороні

гіперплощини, то він буде класифікований до одного класу, а якщо на іншій стороні - до іншого класу.

Однією з переваг методу опорних векторів є його здатність до класифікації у високорозмірних просторах, коли кількість ознак досить велика. Він також ефективно працює навіть з обмеженим обсягом тренувальних даних. Крім того, SVM має математичні основи, що дозволяють уникнути локальних мінімумів і забезпечити оптимальне розділення класів.

Однак, метод опорних векторів має певні недоліки. Він вимагає великої кількості обчислень для знаходження оптимальної гіперплощини, особливо при великій кількості ознак. Також він може бути чутливим до нерівномірного розподілу класів або наявності шуму в даних.

Узагалі, метод опорних векторів є потужним і гнучким інструментом для класифікації зображень. Він може забезпечити високу точність та ефективно працювати з великими обсягами даних, якщо враховувати його обмеження та враховувати відповідні параметри і особливості проблеми.

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) є потужними моделями машинного навчання для аналізу зображень. Вони ефективно використовуються для класифікації, виявлення об'єктів, сегментації зображень та багатьох інших завдань обробки зображень. Основна ідея CNN полягає у використанні згорткових шарів та пулінгових шарів для поетапного виявлення різних рівнів абстракції в зображенні [15].

Основний процес роботи CNN можна розглянути наступним чином:

1. Згортковий шар: у першому шарі згорткової мережі використовуються фільтри, які скользять по вхідному зображенню. Кожен фільтр виконує операцію згортки, перемножаючи свої ваги з пікселями вхідного зображення та додаючи результати. Це дозволяє виявити локальні особливості, такі як границі, форми, текстури тощо. В результаті отримується карта активації (карта ознак).

2. Функція активації: після кожного згорткового шару використовується нелінійна функція активації, така як ReLU (Rectified Linear Unit), що дозволяє виявити нелінійні залежності в даних.
3. Пулінговий шар: після згорткових шарів може бути застосований пулінговий шар, який зменшує розмір карти ознак, виконуючи операцію пулінгу (наприклад, максимального пулінгу) в певному регіоні. Це допомагає знизити кількість параметрів і забезпечує інваріантність до маленьких зміщень об'єктів на зображенні.
4. Повторення згорткових та пулінгових шарів: Згорткові та пулінгові шари можуть бути повторені кілька разів для поступового виявлення більш високорівневих ознак.
5. Повнозв'язний шар: на останньому етапі, після декількох згорткових та пулінгових шарів, отримана карта ознак змінюється в одновимірний вектор та проходить через повнозв'язний шар. Повнозв'язний шар містить нейрони, які виконують класифікацію на основі отриманої інформації про ознаки.
6. Функція втрат: для навчання моделі використовується функція втрат, яка оцінює різницю між прогнозованими і справжніми значеннями. Попередньо навчена модель використовує цю функцію для підналагодження своїх внутрішніх ваг та параметрів.

Цей процес триває кілька епох навчання, де модель змінює свої ваги, щоб максимізувати точність класифікації на тренувальних даних. Після навчання модель може бути застосована для класифікації нових зображень, передбачаючи їх приналежність до одного з класів.

Цей опис дає загальне уявлення про те, як працює згорткова нейронна мережа, деталі реалізації будуть розглянуті в наступному розділі.

Отже, опорні вектори є потужними методами машинного навчання для багатьох задач, включаючи класифікацію зображень. Вони використовуються для знаходження оптимального розділяючого гіперплощини між класами. SVM прагне знайти границю прийняття рішень, яка максимально відділяє класи в

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 21   |

просторі ознак. Однак, для аналізу зображень, особливо коли враховуються просторові залежності та локальні ознаки, CNN є більш ефективним вибором. CNN використовує згорткові шари для локального аналізу зображення, виявлення різних рівнів абстракції та автоматичного вивчення ваг цих ознак. Вони дозволяють виявити границі, форми, текстури та інші важливі ознаки зображень, використовуючи внутрішню структуру зображення. Ще однією перевагою CNN є їх здатність до автоматичного вивчення ознак, тобто необхідні фільтри та ознаки для аналізу зображень не потрібно вказувати вручну, вони вивчаються під час навчання моделі. Це робить CNN більш гнучкими та здатними до адаптації до різноманітних завдань аналізу зображень.

Таким чином, ми обрали CNN для нашої задачі класифікації зображень, оскільки вона є потужним інструментом для обробки зображень, здатним автоматично виявляти важливі ознаки та використовувати просторову структуру зображення. Використання CNN дозволить нам досягти більш точних результатів класифікації та кращої здатності до узагальнення на нові зображення.

### 1.5 Постановка задачі

Згідно з актуальністю аналізу зображень головного мозку та широкого розповсюдження використання нейронних мереж в медицині, об'єктом нашого дослідження обрано процес аналізу МРТ-зображень головного мозку людини з хворобою Альцгеймера з використанням згорткових нейронних мереж. Зосереджуючись на цій темі, нашим основним завданням є автоматизація процесу виявлення хвороби Альцгеймера на різних її стадіях.

Метою дипломної роботи є розробка програмного забезпечення, яке використовує згорткові нейронні мережі для виявлення ранніх стадій хвороби Альцгеймера шляхом аналізу МРТ-зображень мозку пацієнтів.

Отже, для досягнення вказаної мети роботи треба виконати наступні завдання:

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 22   |

- проаналізувати сучасний стан задачі аналізу МРТ-зображень мозку людей з захворюванням Альцгеймера;
- дослідити існуючі технології для вирішення поставленої задачі;
- реалізувати штучну нейронну мережу та її налаштування для аналізу МРТ-зображень;
- навчити та протестувати нейронну мережу.

## Висновки до розділу 1

У даному розділі було розглянуто одну з основних проблем сучасної медицини – діагностування хвороби Альцгеймера. Було розглянуто що являє собою ХА, її стадії, причини, розвиток, клінічні типи, виділено найголовніші біомаркери структурних змін головного мозку при Альцгеймері, за допомогою яких на МРТ-зображеннях і визначають хворобу. Також було проведено аналіз існуючих методів до автоматизованого розпізнавання ХА на МРТ-зображеннях, і в результаті цього аналізу було зроблено висновок, що найкращим підходом до вирішення даної проблеми є використання згорткових нейронних мереж. Крім цього була формалізована постановка задачі – розпізнавання хвороби Альцгеймера по знімкам МРТ за допомогою згорткової нейронної мережі.

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 23   |

## РОЗДІЛ 2

### МЕТОДИ АНАЛІЗУ МРТ-ЗОБРАЖЕНЬ

Впровадження машинного навчання у медичну сферу відкриває безліч нових можливостей для точної та об'єктивної оцінки зображень, отриманих з магнітно-резонансної томографії (МРТ) сканування головного мозку.

#### 2.1 Машинне навчання

Машинне навчання (МН, англ. MachineLearning, ML) –широкий клас методів штучного інтелекту, який займається розробкою алгоритмів та моделей, що дозволяють комп'ютерам самостійно вчитися і покращувати свою продуктивність на основі даних, без явного програмування. Якщо говорити формально, то задача полягає в наступному: на основі сукупності прикладів вхідних і вихідних даних функції  $f$  отримати функцію  $h$  (гіпотезу), яка апроксимує  $f$  [15].

Замість того, щоб напряду кодувати правила для вирішення конкретних завдань, машинне навчання дозволяє комп'ютерам «навчатися» на прикладах і даних. Алгоритми МН використовують статистичні методи та математичні моделі, щоб знайти закономірності та патерни у великих обсягах даних і зробити прогнози або прийняти рішення на основі цих даних.

Машинне навчання використовується в багатьох сферах, включаючи розпізнавання образів, обробку мови, рекомендаційні системи, автономні автомобілі, медичну діагностику та багато іншого. За допомогою машинного навчання можна вирішувати складні завдання, які раніше вимагали людської експертизи і програмування.

Машинне навчання буває трьох видів:

1. Контрольване навчання (англ. SupervisedLearning) – також «навчання з вчителем», передбачає вивчення деякої функції на прикладах її вхідних та вихідних даних.



2. Неконтрольване навчання (англ. UnsupervisedLearning) – також «навчання без вчителя», стосується виявлення певних закономірностей у вхідних даних в тих умовах, коли не задаються конкретні вихідні значення.
3. Навчання з підкріпленням (англ. ReinforcementLearning) –агент, що проходить навчання з підкріпленням, не отримує від учителя вказівок про те, що робити, а повинен навчатися на основі винагород.

## 2.2 Штучна нейронна мережа

Штучна нейронна мережа (ШНМ) – це математична модель, яка намагається відтворити роботу нейронної мережі в головній системі. ШНМ складається зі штучних нейронів, які підтримують обчислення та обмін сигналами. Вона використовується в машинному навчанні для вирішення різних завдань, таких як класифікація, регресія, генерація, впізнавання образів та багато інших.

Нейрон – це основний будівельний блок штучної нейронної мережі. Він моделює нейрон в живому мозку і приймає вхідні сигнали, обчислює їх зважену суму та застосовує нелінійну функцію активації для вироблення вихідного сигналу. На рисунку 2.1 зображена модель нейрона [17].

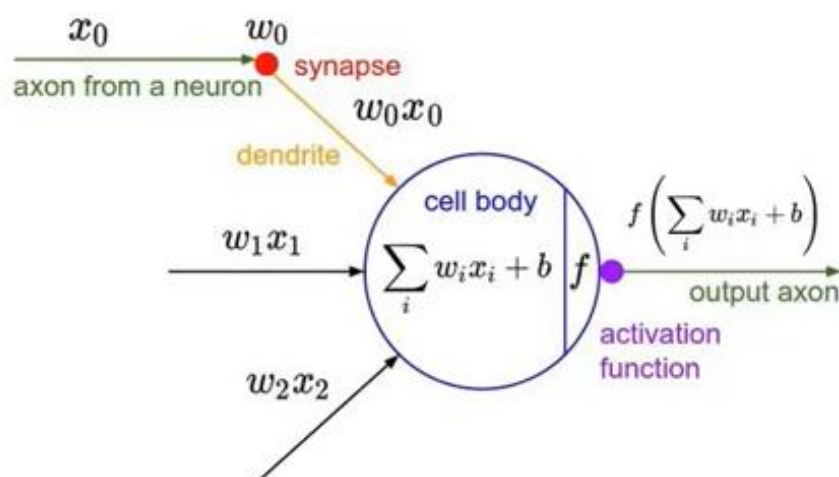


Рисунок 2.1 – Формальна модель нейрона

Відповідно до рисунка 2.1:

- $x_i$  – вхідні сигнали;
- $w_i$  – вагові коефіцієнти (синаптична вага);
- $b$  – зміщення (bias);
- $f$  – нелінійна функція активації.

Кожен нейрон має свої вхідні зв'язки, які мають вагу, що відображає силу зв'язку. Вхідні дані помножуються на ваги, і зважена сума передається функції активації, яка визначає, чи активується нейрон, чи ні. Описані вище дії можна записати у вигляді наступної формули:

$$y = f(\sum w_i x_i + b), \quad (2.1)$$

де  $y$  – вихід нейрона.

Функція активації може бути сигмоїдальною, ReLU (Rectified Linear Unit), тангенсом гіперболічним або іншою. На рисунку 2.2 [17] наведено декілька функцій активації.

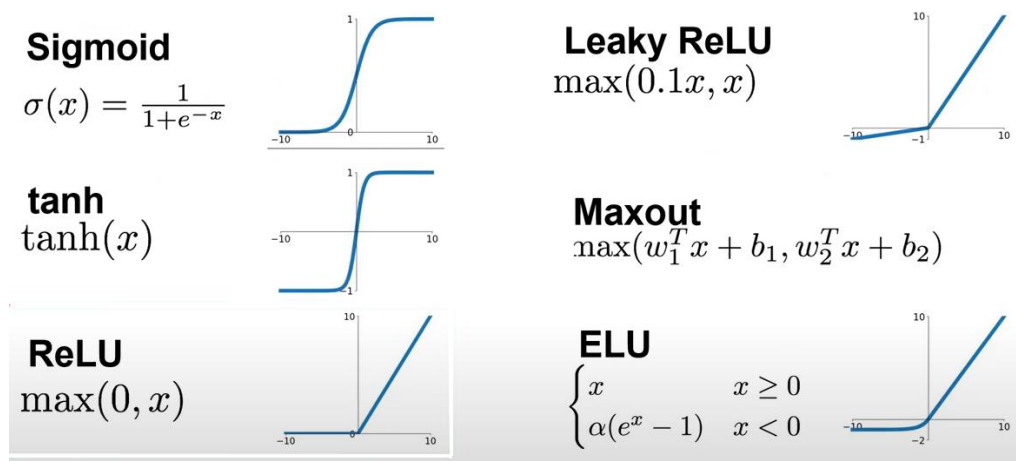


Рисунок 2.2 – Активаційні функції

Функції активації в штучних нейронних мережах відповідають за введення нелінійності в модель та вирішення нелінійних завдань. Вони приймають на вході ваговану суму вхідних сигналів і генерують вихідний сигнал або активацію, який передається наступним шарам мережі.

Функція сигмоїди перетворює будь-яке дійсне число на діапазон від 0 до 1. Вона використовується в задачах бінарної класифікації та в тих випадках, коли потрібно вивести ймовірність належності до одного з двох класів.

Функція ReLU повертає нуль для від'ємних значень і саме число для невід'ємних значень. Функція ReLU широко використовується в глибоких нейронних мережах, оскільки вона ефективно вирішує проблему зниклих градієнтів і підвищує швидкість навчання [18].

Функція тангенса гіперболічного є симетричною відносно початку координат і перетворює дійсні числа на діапазон від -1 до 1. Вона використовується в глибоких нейронних мережах та може бути корисною для прогнозування від'ємних і позитивних значень.

Також, слід виділити ще одну функцію активації – функція SoftMax. Функція активації SoftMax є одним з важливих елементів у штучних нейронних мережах, особливо в задачах багатокласової класифікації. SoftMax перетворює вектор реальних чисел на вектор ймовірностей, де кожне число відображає ймовірність належності до конкретного класу.

Функція активації SoftMax використовується в останньому шарі нейронної мережі, коли потрібно вирішити задачу класифікації. Вхідом для SoftMax є вектор реальних чисел, які називаються "логітами" або "оцінками". Логіти представляють вихідні значення з попереднього шару мережі, що вказують на впевненість моделі щодо належності до кожного класу [19].

Формула для обчислення SoftMax виглядає так:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^K e^{z_k}} \quad (2.2)$$

В цій формулі  $z_i$  – елементи вхідного вектору функції SoftMax, може приймати будь-які дійсні значення;  $e^{z_i}$  – стандартна експоненціальна функція, яка застосовується до кожного елементу вхідного вектору;  $\sum_{k=1}^K e^{z_k}$  – сума за для нормалізації, щоб гарантувати, що сума всіх вхідних значень функції буде

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 27   |

дорівнювати 1, і кожне значення буде належати діапазону (0,1);  $K$  – кількість класів в мультикласифікаторі.

Штучна нейронна мережа складається з багатьох таких нейронів, організованих у шари. Вхідний шар приймає вхідні дані, внутрішні шари обчислюють інтермедіатні репрезентації, а вихідний шар генерує вихідні результати або передає їх наступному шару ШНМ. Процес передачі сигналів та ваг відбувається через мережу, а ваги нейронів оновлюються під час тренування, щоб відповідати певним шаблонам або даним[20].

Розглянемо більш детально шари нейронної мережі. Шар нейронів – це група нейронів в штучній нейронній мережі, які знаходяться на одному рівні або одному горизонталі в архітектурі мережі. Кожен нейрон у шарі приймає вхідні сигнали, обчислює їх зважену суму та застосовує функцію активації для вироблення вихідного сигналу.

Штучні нейронні мережі зазвичай складаються з трьох основних типів шарів: вхідного шару, прихованих шарів і вихідного шару (рис. 2.3)[17].

1. Вхідний шар: це перший шар нейронів, який приймає вхідні дані, такі як зображення, текст або числові значення. Кількість нейронів у вхідному шарі відповідає кількості вхідних функцій або ознак.
2. Приховані шари: це шари, які знаходяться між вхідним та вихідним шарами. Кількість та розміщення прихованих шарів можуть різнитися залежно від архітектури мережі. Кожен шар нейронів у прихованому шарі отримує сигнали від попереднього шару та передає їх наступному шару.
3. Вихідний шар: це останній шар нейронів, який генерує вихідні сигнали або прогнози на основі обробки інформації з попередніх шарів. Кількість нейронів у вихідному шарі відповідає кількості класів у задачах класифікації або кількості вихідних значень у задачах регресії.

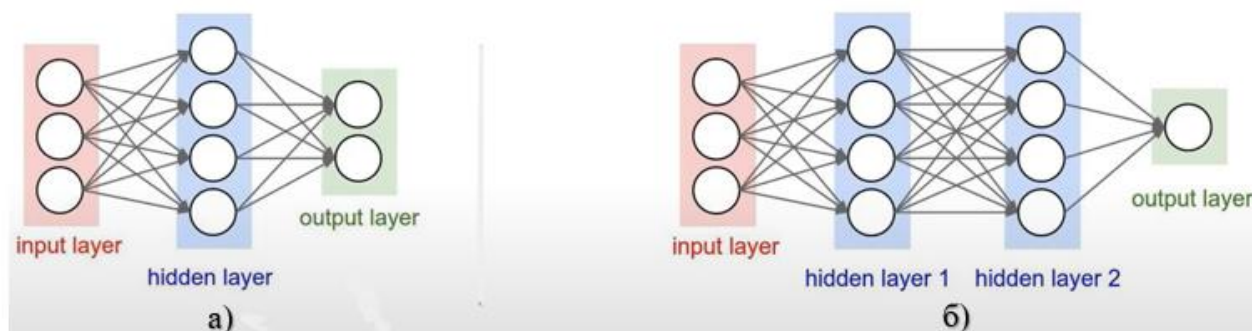


Рисунок 2.3 – 2-шарова нейронна мережа, або мережа з одним прихованим шаром (а); 3-шарова нейронна мережа, або мережа з двома прихованими шарами (б)

Кожен шар нейронів у мережі має свої ваги, які використовуються для зваженої суми вхідних сигналів. Ці ваги навчаються під час процесу навчання, де модель виробляє оптимальні значення ваг для досягнення бажаного вихідного результату. Шари нейронів допомагають моделі вирішувати складні завдання, виконувати обчислення та генерувати відповіді на основі вхідних даних.

### 2.3 Згорткова нейронна мережа

Згорткова нейронна мережа (Convolutional Neural Network, CNN) - це тип штучної нейронної мережі, яка особливо ефективна в обробці та аналізі зображень. Основна ідея згорткових нейронних мереж полягає в тому, щоб автоматично виявляти та витягувати різні локальні ознаки зображень за допомогою фільтрів або ядер згортки [21].

Розглянемо більш детально як працює згорткова нейронна мережа на прикладі простої задачі – класифікації зображень хрестиків і нуликів. Наприклад є зображення хрестика та нулика (рис. 2.4) [17].

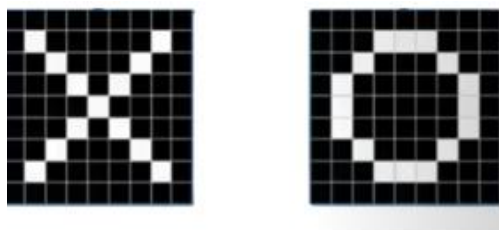
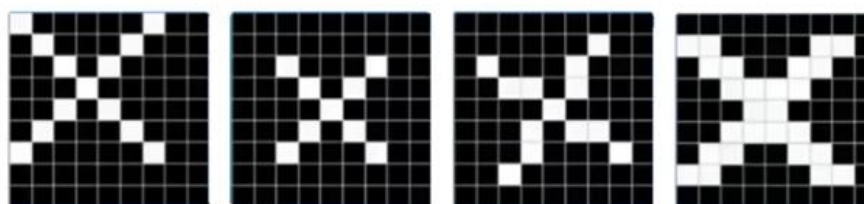


Рисунок 2.4 – Зображення вхідних даних

Використання багат шарового персептрона, що з вхідного зображення зробить один вектор з пікселів, який потім буде оброблятися покроково (попіксельно), призведе до того що кожен піксель безпосередньо оброблюватиметься не залежно від своїх сусідів, і при виникненні більш складніших випадків (рис. 2.5) [17], навчати таку мережу стає набагато складніше.



Зміщення Масштабування Поворот Жирність

Рисунок 2.5 – Складні випадки

Тому для вирішення задачі краще застосовувати ЗНМ. Ідея полягає в тому, що ми будемо аналізувати не конкретні пікселі, а саме частинки зображень (рис. 2.6) [17], це буде набагато ефективніше і дасть кращі результати, так як не буде відбуватися втрата інформації.

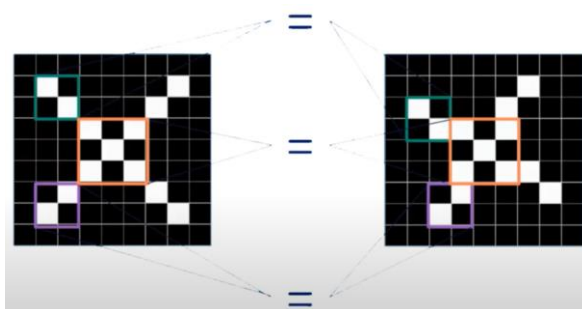


Рисунок 2.6 – Співставлення частин зображення

Але тепер постає питання – як розділити вхідне зображення на частинки? Відповідь – фільтри (ядра). Кожен фільтр представляє собою матрицю чисел, яка проходить через вхідні дані шару за допомогою складання покомпонентного

добутку. В процесі згортки фільтр ковзає по всьому вхідному зображенні (або картці ознак попереднього шару) і обчислює скалярний добуток з відповідною областю даних (рис. 2.7) [17].

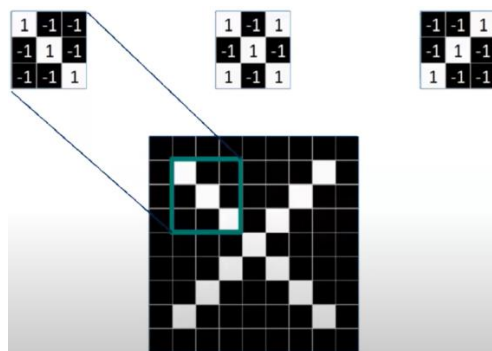


Рисунок 2.7 – Виділення ознак за допомогою фільтрів

Як зазначалося раніше, архітектура згорткової нейронної мережі складається з різних блоків, таких як шари згортки, шари об'єднання та повністю пов'язані шари. Зазвичай, вона має стек повторюваних шарів згортки та шарів об'єднання, за якими йде один або кілька повністю пов'язаних шарів. Цей процес, коли вхідні дані проходять через ці шари та перетворюються на вихідні дані, називається прямим поширенням.

Ознайомимося більш детально з шаром згортки. Він відповідає за вилучення ознак, і виконує комбінацію лінійних і нелінійних операцій, включаючи операції згортки та функції активації.

Операція згортки є спеціалізованим типом лінійної операції, що використовується для вилучення ознак. Вона застосовується до вхідних даних, які представлені у вигляді тензорів, за допомогою невеликого масиву чисел, який називається ядром (рис.2.8) [17].

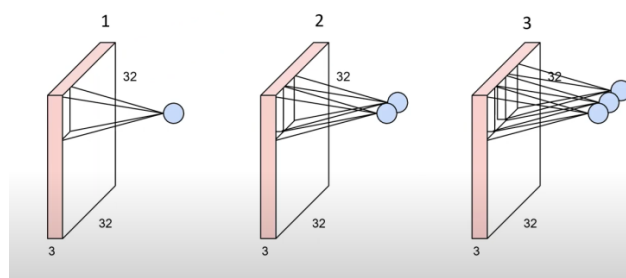


Рисунок 2.8 – Принцип роботи згортки

У кожній позиції тензора обчислюється поелементний добуток між елементами ядра та вхідним тензором, а потім сумується. Це дає вихідне значення у відповідній позиції вихідного тензора, яке називається карткою ознак (карта активації) (рис. 2.9)[17].

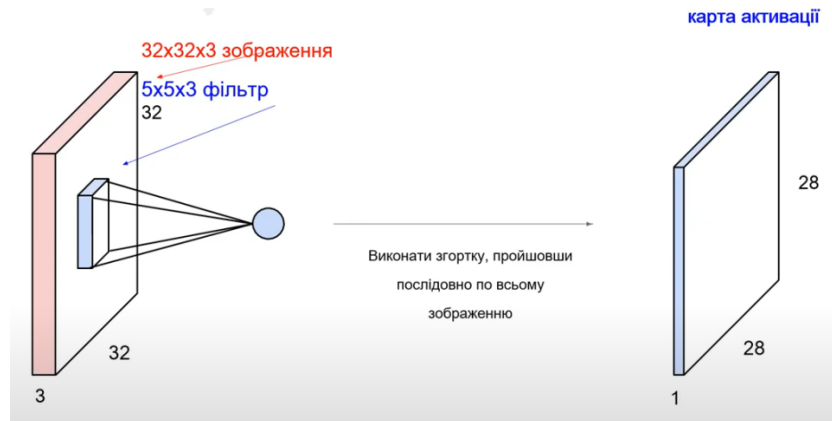


Рисунок 2.9 – Отримання карти активації

Так як один фільтр дорівнює одній властивості певного об'єкту, то зазвичай використовується певна кількість фільтрів з іншими ваговими коефіцієнтами, що дозволяє формувати довільні кількості карт ознак (рис. 2.10). Це дозволяє розглядати різні ядра як екстрактори різних ознак. Операція згортки має чотири головні гіперпараметри: розмір ядра (зазвичай 3x3, іноді 5x5 або 7x7), кількість ядер, крок (зсув) та нульове заповнення. Параметр кількості ядер є довільним і визначає глибину вихідних карт ознак об'єктів.



Рисунок 2.10 – Результат кількох операцій згортки

З зображень видно, що спостерігається деяка математична функція для визначення розмірності карти ознак:



$$\frac{N-F}{S} + 1 \quad (2.3)$$

де,  $N$  – розмірність входу;  $F$  – розмірність фільтра;  $S$  – розмір кроку (зсуву).

Операція згортки, описана вище, обмежує центр кожного ядра вхідного тензора та зменшує розміри вихідної карти об'єктів. Щоб зберегти розміри та дозволити більше шарів у моделі, використовується метод нульового заповнення. Це означає, що до вхідного тензора додаються рядки та стовпці нулів, щоб центр ядра міг знаходитись на крайньому зовнішньому елементі. Завдяки нульовому заповненню розміри карти об'єктів залишаються незмінними після кожного шару згортки (рис. 2.11) [17]. Без нульового заповнення кожна наступна карта об'єктів зменшується після операції згортки.

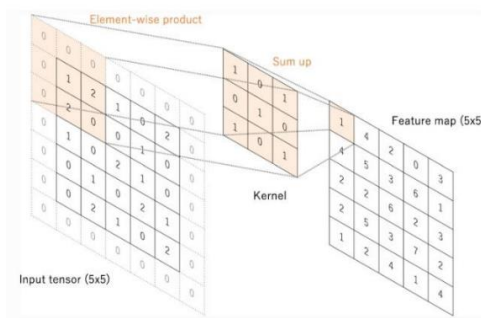


Рисунок 2.11 – Нульове доповнення дає на виході таку саму розмірність як на вході

Вихідні дані після лінійної операції, наприклад згортки, проходять через нелінійну функцію активації, описані раніше.

Шар об'єднання виконує зниження дискретизації, зменшуючи розмірність карт ознак для досягнення трансляційної інваріантності та зменшення кількості параметрів, що підлягають навчанню.

Найпоширенішою формою операції об'єднання є максимальний пулінг. Вона витягує патчі з вхідних карт об'єктів, вибирає максимальне значення у кожному патчі і відкидає решту значень. Зазвичай використовується максимальний пулінг з фільтром розміром  $2 \times 2$  і кроком 2, що зменшує розмірність у площині карт об'єктів у два рази (рис.2.12) [17]. Глибина карт об'єктів залишається незмінною, відмінно від ширини та висоти.



Рисунок 2.12 – Максимальний пулінг

Вихідні карти об'єктів остаточного шару згортки або об'єднання спочатку згладжуються в одновимірний масив чисел (вектор) і потім передаються в один або кілька повністю пов'язаних шарів, також відомих як щільні шари. У щільних шарах кожен вхід пов'язаний з кожним виходом за допомогою ваг, які підлягають навчанню. Після створення об'єктів, виділених згортковими та згортково-пулінговими шарами, вони відображаються підмножиною повністю пов'язаних шарів, що приводить до кінцевих вихідних даних мережі, наприклад, ймовірностей для класів в задачах класифікації. Останній повністю пов'язаний шар зазвичай має кількість вихідних вузлів, рівну кількості класів. Після кожного повністю пов'язаного шару використовується нелінійна функція активації, в нашому випадку, згідно попереднього опису – це функція SoftMax.

Навчання (ЗНМ) є процесом, під час якого модель "вчиться" розпізнавати патерни та виконувати завдання на основі вхідних даних. Розглянемо детально кожен крок навчання ЗНМ:

1. Ініціалізація ваг: На початку навчання ваги ЗНМ ініціалізуються випадковими значеннями. Ваги відображаються у згорткових та повністю пов'язаних шарах мережі.
2. Пряме поширення: Вхідні дані подаються на вхідний шар ЗНМ. Поширюючись вперед через згорткові шари, карти ознак формуються шляхом виконання згортки та активації. Потім вони можуть проходити через шари пулінгу, де розмір карти зменшується, зберігаючи важливі особливості.

3. Обчислення втрат: Вихідні дані з Остаточного шару порівнюються з бажаними вихідними значеннями, що відповідають вхідним даним. Це дозволяє обчислити втрату або помилку моделі, яка вказує, наскільки добре вона прогнозує вихідні дані.
4. Зворотне поширення помилки: Після обчислення втрати використовується алгоритм зворотного поширення помилки, щоб розповсюдити помилку назад через мережу. Це дозволяє обчислити градієнти ваг, які показують напрям, в якому потрібно змінити ваги, щоб зменшити втрату.
5. Оновлення ваг: Загальний градієнт ваг обчислюється, використовуючи градієнти зворотного поширення помилки. Потім ваги оновлюються з використанням певного алгоритму оптимізації, такого як стохастичний градієнтний спуск (SGD) або його варіації. Оновлення ваг допомагають збільшити продуктивність мережі та зменшити помилку.
6. Повторення процесу: Ці кроки повторюються для кожного пакету тренувальних даних (батчу), доки мережа не пройде всі дані тренувального набору. Кількість ітерацій і розмір пакетів визначаються гіперпараметрами навчання, які можуть бути налаштовані.

Цей процес навчання повторюється декілька епох, де кожна епоха представляє собою повний цикл прямого поширення, обчислення втрат, зворотного поширення та оновлення ваг. Модель навчається адаптуватися до даних, покращуючи свої ваги і здатність до розпізнавання патернів.

## Висновки до розділу 2

Отже, у розділі 2 було визначено, що являє собою МН, власне це галузь штучного інтелекту, яка вивчає алгоритми та моделі, що можуть навчатися на основі даних і виконувати завдання без явного програмування. Ключовими компонентами машинного навчання є штучні нейронні мережі, які моделюють поведінку людського мозку. Ці мережі складаються зі шарів нейронів, де вхідні

|      |      |          |        |      |                   |            |
|------|------|----------|--------|------|-------------------|------------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист<br>35 |
|      |      |          |        |      |                   |            |
| Изм. | Лист | № докум. | Підпис | Дата |                   |            |

дані проходять через лінійні операції та функції активації. Згорткові нейронні мережі є підтипом штучних нейронних мереж і використовуються для обробки зображень. Вони складаються зі згорткових шарів, які використовують фільтри для виявлення особливостей, та шарів пулінгу, які зменшують розмірність. Навчання згорткової нейронної мережі включає ініціалізацію ваг, пряме поширення сигналу, зворотне поширення помилки та оновлення ваг на основі алгоритму оптимізації, такого як стохастичний градієнтний спуск. Застосування згорткових нейронних мереж включає розпізнавання об'єктів, класифікацію даних та багато інших завдань, що вимагають обробки та аналізу зображень.

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 36   |

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ НЕЙРОННИХ МЕРЕЖ

#### 3.1 Програмне забезпечення та використання бібліотек

Згорткові нейронні мережі можуть бути реалізовані з використанням різних програмних мов, проте Python[22] є однією з найпопулярніших мов для розробки машинного навчання і має кілька переваг, які роблять його відмінним вибором для написання згорткових нейронних мереж:

1. Широкий вибір бібліотек: Python має багато популярних бібліотек, таких як TensorFlow, Keras, PyTorch та Theano, які надають потужні інструменти для роботи з нейронними мережами. Ці бібліотеки мають готові реалізації згорткових шарів, функцій активації та інших компонентів, що значно спрощує розробку та експериментування з моделями.
2. Легкість використання: Python має простий і зрозумілий синтаксис, який полегшує розробку і зрозуміння коду. Він також має велику спільноту розробників, що означає, що ви можете знайти багато прикладів, документації та підтримки в разі потреби.
3. Інтеграція з іншими інструментами: Python є широко використовуваною мовою в області наукових обчислень і аналізу даних. Він має розширення і бібліотеки для роботи з різними типами даних, включаючи зображення. Це означає, що ви можете легко інтегрувати згорткові нейронні мережі з іншими інструментами для попередньої обробки даних, візуалізації та аналізу результатів.
4. Висока продуктивність: Python має багато оптимізованих бібліотек, які дозволяють прискорити виконання обчислень. Наприклад, TensorFlow та PyTorch підтримують використання GPU для прискорення обчислень в згорткових нейронних мережах.

Загалом, вибір Python для написання згорткових нейронних мереж є розумним рішенням, оскільки він надає зручний інтерфейс, широкий вибір бібліотек, зрозумілий синтаксис та високу продуктивність, що сприяє швидкій розробці та ефективному впровадженню моделей.

### 3.2 Набір даних

Для даної роботи було взято набір даних під назвою Alzheimer'sDataset (4 class of Images) з платформи Kaggle. Kaggle – це веб-сервіс, який сприяє співпраці між науковцями з області даних та ентузіастами машинного навчання. Ця платформа дозволяє користувачам знаходити, публікувати та спільно працювати над наборами даних, використовувати графічні процесори у вбудованих ноутбуках та змагатися з іншими учасниками у вирішенні завдань з науки про дані. Головна мета Kaggle – допомогти професіоналам і студентам досягти своїх цілей у вивченні науки про дані, надаючи потужні інструменти та ресурси. На сьогоднішній день на Kaggle зареєстровано понад 8 мільйонів користувачів [23].

Набір містить 6400 зображень і поділений на 4 класи:

- NonDemented – клас МРТ-зображень мозку здорових людей;
- Very Mild Demented – дуже легке ураження мозку;
- MildDemented – легке ураження мозку;
- Moderate Demented – помірне ураження мозку.

Перевагою даного датасету є те, що із зображень вже видалені «непотрібні» фрагменти, які не впливають на діагностування ХА, в результаті наша ЗНМ буде навчатись більш коректно. Також датасет вже поділений на тренувальні та тестувальні дані, у відношенні 80:20.

З недоліків лише те, що розмірність для навчання ЗНМ має бути однаковою для ширини та довжини. Розмір зображення з даного датасету дорівнює 176x208. Але це легко можна виправити додавши пару рядків коду до моделі.

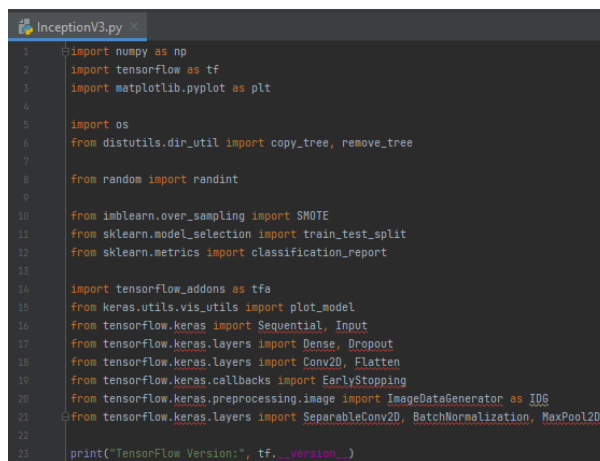
### 3.3 Проведення експериментів

В даній роботі було виконано низку експериментів, для визначення підходящої (найбільш точної) моделі – InceptionV3, VGG19, ResNet50. Нижче буде детально описано переваги та недоліки кожної з них.

Перша модель, яка була побудована – InceptionV3 [124]. InceptionV3 - це глибока згорткова нейронна мережа, яка була розроблена командою Google Research для розпізнавання об'єктів на зображеннях. Вона є покращенням попередньої моделі InceptionV1 і має вражаючі результати в області комп'ютерного зору.

Основна ідея InceptionV3 полягає в використанні модулів Inception, які дозволяють мережі вирішувати проблему вибору оптимального розміру фільтра в згорткових шарах. Замість використання фільтрів одного фіксованого розміру, InceptionV3 використовує фільтри різних розмірів (1x1, 3x3, 5x5) паралельно і об'єднує їх в один вихід шляхом конкатенації. Це дозволяє мережі ефективно виявляти ознаки різного рівня деталізації. Окрім цього, InceptionV3 використовує модуль зменшення розмірності перед складними блоками Inception. Це допомагає знизити обчислювальні витрати і сприяє збільшенню глибини мережі.

Перейдемо до написання коду. Перш за все імпортуємо необхідні бібліотеки, щоб виконувати потрібні операції, такі як робота з даними, навчання моделі, візуалізація та оцінка результатів (рис.3.1).



```
InceptionV3.py
1 import numpy as np
2 import tensorflow as tf
3 import matplotlib.pyplot as plt
4
5 import os
6 from distutils.dir_util import copy_tree, remove_tree
7
8 from random import randint
9
10 from imblearn.over_sampling import SMOTE
11 from sklearn.model_selection import train_test_split
12 from sklearn.metrics import classification_report
13
14 import tensorflow_addons as tfa
15 from keras.utils.vis_utils import plot_model
16 from tensorflow.keras import Sequential, Input
17 from tensorflow.keras.layers import Dense, Dropout
18 from tensorflow.keras.layers import Conv2D, Flatten
19 from tensorflow.keras.callbacks import EarlyStopping
20 from tensorflow.keras.preprocessing.image import ImageDataGenerator as IDG
21 from tensorflow.keras.layers import SeparableConv2D, BatchNormalization, MaxPool2D
22
23 print("TensorFlow Version:", tf.__version__)
```

Рисунок 3.1 – Імпортування необхідних бібліотек

|      |      |          |        |      |                   |            |
|------|------|----------|--------|------|-------------------|------------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист<br>39 |
|      |      |          |        |      |                   |            |
| Изм. | Лист | № докум. | Підпис | Дата |                   |            |

У цих рядках імпортується NumPy для роботи з масивами та математичними операціями, TensorFlow для побудови та навчання моделей машинного навчання, та Matplotlib для візуалізації графіків та зображень. Далі імпортуємо модуль os для роботи з операційною системою, а саме для роботи з файловою системою. distutils.dir\_util містить функції для копіювання та видалення каталогів. Імпортуємо різні функції та метрики з бібліотеки imbalanced-learn (imblearn) та scikit-learn. SMOTE є алгоритмом для вирішення проблеми незбалансованості класів у наборі даних. train\_test\_split використовується для розділення набору даних на тренувальну та тестову вибірки. MCC та BAS є метриками для оцінки моделей. classification\_report та confusion\_matrix використовуються для аналізу результатів класифікації. І останній блок імпорту – це різні класи та модулі з TensorFlow та Keras, що використовуються для побудови та навчання моделей глибокого навчання. Вони містять такі компоненти, як послідовність шарів (Sequential), вхід (Input), підсумовування (Dense), випадкове відключення (Dropout), згорткові шари (Conv2D), плоский шар (Flatten), раннє зупинення (EarlyStopping), InceptionV3 (популярна архітектура глибокої мережі), ImageDataGenerator (для аугментації зображень) та інші.

Після імпортування виведемо на екран версію TensorFlow, результат на рисунку 3.2.

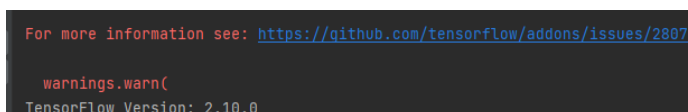


Рисунок 3.2 – Вивід на екран версії TensorFlow

Далі необхідно підготувати вхідні дані для нашої моделі перед тренуванням – визначення шляхів до потрібних папок у файловій системі, таких як папки базового каталогу, кореневої директорії, тестової та тренувальної папок, перевірка наявності робочої директорії, якщо вона вже існує, вона видаляється, створення нової робочої директорії, в яку потім копіюються файли

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 40   |



з тренувальної та тестової папок (рис. 3.3). Це дозволяє зібрати всі дані в одному місці для подальшої обробки.

```

27 base_dir = "Alzheimer_s Dataset/"
28 root_dir = "./"
29 test_dir = base_dir + "test/"
30 train_dir = base_dir + "train/"
31 work_dir = root_dir + "dataset/"
32
33 if os.path.exists(work_dir):
34     remove_tree(work_dir)
35
36 os.mkdir(work_dir)
37 copy_tree(train_dir, work_dir)
38 copy_tree(test_dir, work_dir)
39 print("Working Directory Contents:", os.listdir(work_dir))

```

Рисунок 3.3 – Створення структури даних

На завершення виводиться список файлів та папок, що знаходяться в робочій директорії, що допомагає переконатися в правильності структури даних (рис 3.4).

```
Working Directory Contents: ['MildDemented', 'ModerateDemented', 'NonDemented', 'VeryMildDemented']
```

Рисунок 3.4 – Робоча директорія

Визначемо класифікацію на класи та задамо значення розміру для вхідних зображень, як зазначалося раніше воно повинне бути png:

```

41 WORK_DIR = './dataset/'
42
43 CLASSES = ['NonDemented',
44            'VeryMildDemented',
45            'MildDemented',
46            'ModerateDemented']
47
48 IMG_SIZE = 176
49 IMAGE_SIZE = [176, 176]
50 DIM = (IMG_SIZE, IMG_SIZE)

```

Рисунок 3.5 – Задання класів та визначення розмірів зображення

Далі ми виконуємо аугментацію зображень для отримання більшої кількості прикладів даних. Змінні «ZOOM», «BRIGHT\_RANGE», «HORZ\_FLIP», «FILL\_MODE» та «DATA\_FORMAT» визначають

параметри аугментації, такі як збільшення масштабу, діапазон яскравості, горизонтальне відображення та інші.

Об'єкт «work\_dr» є екземпляром класу «ImageDataGenerator» з бібліотеки TensorFlow, який використовується для аугментації зображень. Ми налаштовуємо цей об'єкт з використанням визначених параметрів, таких як масштабування зображень до діапазону від 0 до 1, зміна яскравості, збільшення масштабу та інші. «train\_data\_gen» є генератором даних, який буде створювати пакети зображень для навчання моделі. Ми вказуємо директорію «WORK\_DIR», в якій знаходяться дані, «target\_size» – розмір зображень, «batch\_size» – розмір пакету зображень і «shuffle=False», щоб зберігати порядок зображень таким, як в директорії (рис.3.6).

```
54 ZOOM = [.99, 1.01]
55 BRIGHT_RANGE = [0.8, 1.2]
56 HORZ_FLIP = True
57 FILL_MODE = "constant"
58 DATA_FORMAT = "channels_last"
59
60 work_dr = IDG(rescale=1. / 255, brightness_range=BRIGHT_RANGE, zoom_range=ZOOM, data_format=DATA_FORMAT,
61              fill_mode=FILL_MODE, horizontal_flip=HORZ_FLIP)
62
63 train_data_gen = work_dr.flow_from_directory(directory=WORK_DIR, target_size=DIM, batch_size=6500, shuffle=False)
```

Рисунок 3.6 – Аугментація зображень

Після того як дані були завантажені та відформатовані, треба вивести на екран зображення, що допоможе зрозуміти, що використовується як вхідні дані для моделі (рис.3.7).

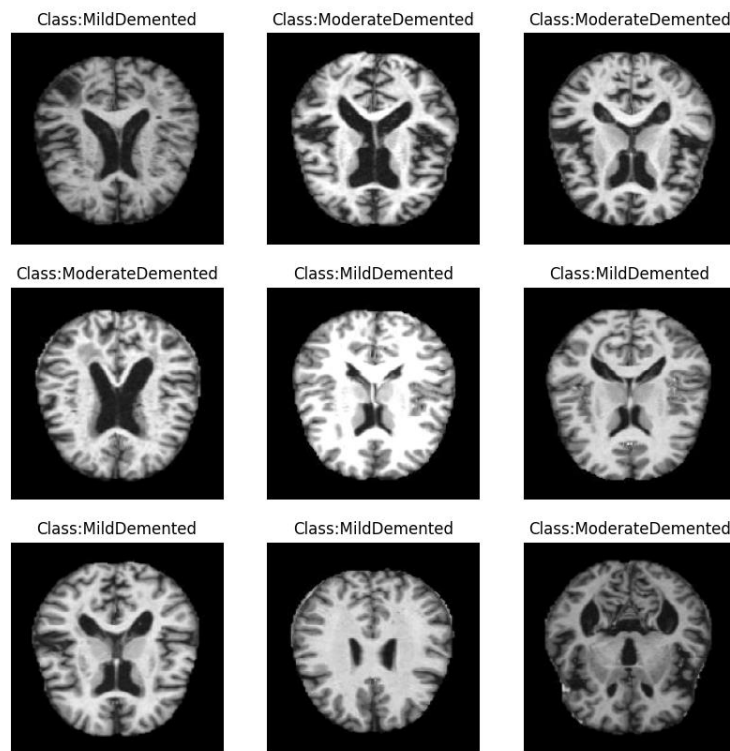


Рисунок 3.7 – Вивід вхідних даних

```

100 #Retrieving the data from the ImageDataGenerator iterator
101 train_data, train_labels = train_data_gen.next()
102
103 #Getting to know the dimensions of our dataset
104 print(train_data.shape, train_labels.shape)
105
106 #Performing over-sampling of the data, since the classes are imbalanced
107 sm = SMOTE(random_state=42)
108 train_data, train_labels = sm.fit_resample(train_data.reshape(-1, IMG_SIZE * IMG_SIZE * 3), train_labels)
109 train_data = train_data.reshape(-1, IMG_SIZE, IMG_SIZE, 3)
110
111 print(train_data.shape, train_labels.shape)
112
113 |
114 #Splitting the data into train, test, and validation sets
115
116 train_data, test_data, train_labels, test_labels = train_test_split(train_data, train_labels, test_size=0.2, random_state=42)
117 train_data, val_data, train_labels, val_labels = train_test_split(train_data, train_labels, test_size=0.2, random_state=42)

```

Рисунок 3.8 – Перебалансування даних

У цьому фрагменті (рис. 3.8) ми виконуємо наступні дії:

1. Здійснюємо аугментацію зображень, щоб отримати більше прикладів даних. Ми використовуємо ImageDataGenerator для отримання даних з генератора train\_data\_gen.
2. Виводимо розмірність набору даних (train\_data) та міток (train\_labels), щоб переконатися, що дані були отримані правильно.

3. Виконуємо oversampling (перебалансовування) даних, оскільки класи набору даних не збалансовані. Використовуємо метод SMOTE для збільшення кількості зразків шляхом створення штучних прикладів меншості класів. Ми перетворюємо дані (train\_data) та мітки (train\_labels) у формат, який може бути використаний SMOTE, а потім повертаємо їх у вихідний формат.
4. Розділяємо дані на тренувальний набір (train\_data, train\_labels), тестовий набір (test\_data, test\_labels) та валідаційний набір (val\_data, val\_labels). Використовуємо функцію train\_test\_split для цього. Розмір тестового та валідаційного набору встановлюється на 20% від загального набору даних, і ми використовуємо випадковий стан (random\_state=42) для повторюваності розділення даних.

Результати виводу на екран, рисунок 3.9:

```
Found 6400 images belonging to 4 classes.
(6400, 176, 176, 3) (6400, 4)
(12800, 176, 176, 3) (12800, 4)
```

Рисунок 3.9 – Балансування даних

Перейдемо до створення архітектури моделі, рисунок 3.10, 3.11:

```
121 def conv_block(filters, act='relu'):
122     """Defining a Convolutional NN block for a Sequential CNN model. """
123
124     block = Sequential()
125     block.add(Conv2D(filters, 3, activation=act, padding='same'))
126     block.add(Conv2D(filters, 3, activation=act, padding='same'))
127     block.add(BatchNormalization())
128     block.add(MaxPool2D())
129
130     return block
131
132
133 def dense_block(units, dropout_rate, act='relu'):
134     """Defining a Dense NN block for a Sequential CNN model. """
135
136     block = Sequential()
137     block.add(Dense(units, activation=act))
138     block.add(BatchNormalization())
139     block.add(Dropout(dropout_rate))
140
141     return block
142
```

Рисунок 3.10 – Створення архітектури моделі (перша частина)

```

def construct_model(act='relu'):
    """Constructing a Sequential CNN architecture for performing the classification task. """

    model = Sequential([
        Input(shape=(IMAGE_SIZE, 3)),
        Conv2D(16, 3, activation=act, padding='same'),
        Conv2D(16, 3, activation=act, padding='same'),
        MaxPool2D(),
        conv_block(32),
        conv_block(64),
        conv_block(128),
        Dropout(0.2),
        conv_block(256),
        Dropout(0.2),
        Flatten(),
        dense_block(512, 0.7),
        dense_block(128, 0.5),
        dense_block(64, 0.3),
        Dense(4, activation='softmax')
    ], name="cnn_model")

    return model

```

Рисунок 3.11 – Створення архітектури моделі (друга частина)

| Model: "cnn_model"           |                      |         |
|------------------------------|----------------------|---------|
| Layer (type)                 | Output Shape         | Param # |
| conv2d (Conv2D)              | (None, 176, 176, 16) | 448     |
| conv2d_1 (Conv2D)            | (None, 176, 176, 16) | 2320    |
| max_pooling2d (MaxPooling2D) | (None, 88, 88, 16)   | 0       |
| sequential (Sequential)      | (None, 44, 44, 32)   | 14016   |
| sequential_1 (Sequential)    | (None, 22, 22, 64)   | 55680   |
| sequential_2 (Sequential)    | (None, 11, 11, 128)  | 221952  |
| dropout (Dropout)            | (None, 11, 11, 128)  | 0       |
| sequential_3 (Sequential)    | (None, 5, 5, 256)    | 886272  |
| dropout_1 (Dropout)          | (None, 5, 5, 256)    | 0       |
| flatten (Flatten)            | (None, 6400)         | 0       |
| sequential_4 (Sequential)    | (None, 512)          | 3279360 |
| sequential_5 (Sequential)    | (None, 128)          | 66176   |
| sequential_6 (Sequential)    | (None, 64)           | 8512    |
| dense_3 (Dense)              | (None, 4)            | 260     |
| Total params: 4,534,996      |                      |         |
| Trainable params: 4,532,628  |                      |         |
| Non-trainable params: 2,368  |                      |         |

Рисунок 3.12 – Вивід шарів створенної архітектури

Визначаємо деякі інші параметри для нашої згорткової нейронної мережі (рис. 3.13). Спочатку ми викликаємо функцію `construct_model()`, яка буде будувати архітектуру нашої ЗНМ. Потім ми визначаємо список метрик, які хочемо використовувати для оцінки продуктивності моделі. У цьому випадку, ми використовуємо метрики категоріальної точності, площі під кривою AUC і F1-оцінки для 4 класів.

Далі ми компілюємо модель за допомогою оптимізатора "adam", функції втрати "CategoricalCrossentropy" і визначених раніше метрик.

Нарешті, ми виводимо зведену інформацію про модель, використовуючи метод `summary()`, який виводить детальну інформацію про шари та параметри моделі (рис. 3.14).

```

183     model = construct_model()
184
185     METRICS = [tf.keras.metrics.CategoricalAccuracy(name='acc'),
186                tf.keras.metrics.AUC(name='auc'),
187                tfa.metrics.F1Score(num_classes=4)]
188
189     CALLBACKS = [my_callback]
190
191     model.compile(optimizer='adam',
192                  loss=tf.losses.CategoricalCrossentropy(),
193                  metrics=METRICS)
194
195     model.summary()

```

Рисунок 3.13 – Встановлення додаткових параметрів для моделі

```

=====
Total params: 4,534,996
Trainable params: 4,532,628
Non-trainable params: 2,368
=====

```

Рисунок 3.14 – Інформація про параметри моделі

Процес навчання моделі складатиметься з 20 епох (в разі необхідності модель можна довчити), цього більш ніж достатньо для навчання, адже зі

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   | 46   |
| Изм. | Лист | № докум. | Підпис | Дата |                   |      |

збільшенням кількості епох може зрости ймовірність перенавчання моделі на тренувальних даних. Це означає, що модель може надмірно вивчати тренувальні дані, що призводить до погіршення її здатності до узагальнення на нові дані. Процес навчання зображено на рисунку 3.15:

```
Epoch 16/20
256/256 [=====] - 254s 991ms/step - loss: 0.2989 - acc: 0.8860 - auc: 0.9839 - f1_score: 0.8860 - val_loss: 0.4673 - val_acc: 0.8047 - val_auc: 0.9654 - val_f1_score: 0.7983
Epoch 17/20
256/256 [=====] - 250s 978ms/step - loss: 0.2651 - acc: 0.8966 - auc: 0.9866 - f1_score: 0.8964 - val_loss: 0.2985 - val_acc: 0.8833 - val_auc: 0.9840 - val_f1_score: 0.8822
Epoch 18/20
256/256 [=====] - 252s 983ms/step - loss: 0.2082 - acc: 0.9235 - auc: 0.9914 - f1_score: 0.9235 - val_loss: 1.6038 - val_acc: 0.5669 - val_auc: 0.8250 - val_f1_score: 0.5619
Epoch 19/20
256/256 [=====] - 255s 994ms/step - loss: 0.2039 - acc: 0.9231 - auc: 0.9917 - f1_score: 0.9230 - val_loss: 0.2481 - val_acc: 0.9009 - val_auc: 0.9873 - val_f1_score: 0.8996
Epoch 20/20
256/256 [=====] - 256s 1s/step - loss: 0.1999 - acc: 0.9271 - auc: 0.9919 - f1_score: 0.9272 - val_loss: 0.3273 - val_acc: 0.8813 - val_auc: 0.9814 - val_f1_score: 0.8776
80/80 [=====] - 16s 201ms/step - loss: 0.3378 - acc: 0.8789 - auc: 0.9802 - f1_score: 0.8763
Testing Accuracy: 87.89%
80/80 [=====] - 16s 199ms/step
```

Рисунок 3.15 – Процес навчання моделі

|                          |                  |                |               |               |                    |
|--------------------------|------------------|----------------|---------------|---------------|--------------------|
| 80/80 [=====]            | - 16s 201ms/step | - loss: 0.3378 | - acc: 0.8789 | - auc: 0.9802 | - f1_score: 0.8763 |
| Testing Accuracy: 87.89% |                  |                |               |               |                    |
| 80/80 [=====]            | - 16s 199ms/step |                |               |               |                    |
|                          | precision        | recall         | f1-score      | support       |                    |
| NonDemented              | 0.95             | 0.97           | 0.96          | 639           |                    |
| VeryMildDemented         | 0.95             | 1.00           | 0.97          | 635           |                    |
| MildDemented             | 0.97             | 0.63           | 0.76          | 662           |                    |
| ModerateDemented         | 0.72             | 0.93           | 0.81          | 624           |                    |

Рисунок 3.16 – Результат навчання

Відповідно до рисунку 3.16:

1. Loss (втрати) – це значення функції втрати (loss function), яка вимірює, наскільки добре модель передбачає правильні класи для вхідних даних. Чим нижче значення втрати, тим краще модель працює.
2. Асс (точність) – це показник, що визначає відношення кількості правильно класифікованих прикладів до загальної кількості прикладів у наборі даних. Він виражається як доля правильних відповідей моделі. Чим ближче значення до 1, тим краще модель працює.
3. AUC (площа під ROC-кривою) – це міра якості моделі класифікації, яка використовується для оцінки її здатності розрізняти між двома класами. ROC-крива (Receiver Operating Characteristic curve) показує залежність між чутливістю (дійсно позитивні відповіді) і специфічністю (дійсно негативні відповіді) моделі при різних порогових значеннях. Площа під ROC-кривою (AUC) вказує на загальну здатність моделі розрізняти

класи. Значення AUC знаходиться між 0 і 1, де 1 означає ідеальну здатність до класифікації, а 0,5 - випадковий вибір.

4. F1-показник (F1 score) – це міра точності моделі, яка об'єднує інформацію про точність і повноту (чутливість) моделі в одне значення. F1-показник обчислюється як гармонічне середнє точності і повноти і має значення між 0 і 1, де 1 вказує на ідеальну точність і повноту, а 0 - на найгіршу.

На рисунку 3.18 показано результат прогнозування заданого зображення на клас, а також впевненість цього прогнозування, відповідно до заданого зображення (рис.3.17):

```
def load_and_preprocess_image(image_path):
    image = tf.keras.preprocessing.image.load_img(image_path, target_size=IMAGE_SIZE)
    image = tf.keras.preprocessing.image.img_to_array(image)
    image = image / 255.0 # Normalize image
    image = np.expand_dims(image, axis=0) # Add batch dimension
    return image

# Function to predict class and confidence
def predict_image(image_path):
    image = load_and_preprocess_image(image_path)
    predictions = model.predict(image)
    predicted_class_index = np.argmax(predictions)
    predicted_class = CLASSES[predicted_class_index]
    confidence = np.max(predictions) * 100
    return predicted_class, confidence

# Path to the image you want to test
image_path = 'dataset/ModerateDemented/28_(2).jpg'

# Display the predicted class and confidence
predicted_class, confidence = predict_image(image_path)
print("Predicted Class:", predicted_class)
print("Confidence:", confidence)
```

Рисунок 3.17 – Функція прогнозування

```
1/1 [=====] - 0s 194ms/step
Predicted Class: ModerateDemented
Confidence: 99.99920129776001
```

Рисунок 3.18 – Прогнозування

Друга модель VGG19 [25] (VGGstandsforVisualGeometryGroup) є однією з відомих архітектур глибоких нейронних мереж, яка була розроблена колективом дослідників з Університету Оксфорду. VGG19 є покращеною версією VGG16, яка в свою чергу була однією з перших глибоких моделей, які показали

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 48   |



значні результати в багатьох завданнях комп'ютерного зору. Архітектура VGG19 базується на принципі використання невеликих фільтрів (3x3) для згорткових шарів та пулінг шарів. Цей підхід дозволяє збільшити глибину мережі та знизити кількість параметрів, що дозволяє покращити якість розпізнавання зображень.

Основні характеристики VGG19:

- VGG19 має загальною 19 шарів, що включають 16 згорткових шарів (Convolutional layers) і 3 повнозв'язних шари (Fully Connected layers).
- Згорткові шари використовують фільтри розміром 3x3 з відступами 1 піксель і заміщенням (padding) типу "same", що забезпечує збереження розмірів зображення.
- Пулінг шари (Max Pooling) використовуються після кожного згорткового шару з фільтром розміром 2x2 та зсувом 2 пікселя, що допомагає зменшити розмірність шарів та об'єм параметрів.
- Після згорткових шарів слідує повнозв'язні шари, що використовуються для класифікації. Зазвичай останній повнозв'язний шар має 1000 нейронів для розпізнавання 1000 класів, але його можна модифікувати для вирішення інших задач.
- Для активації використовується функція активації ReLU (Rectified Linear Unit) на всіх шарах, за винятком останнього повнозв'язного шару, де використовується функція активації softmax для отримання ймовірностей для класифікації.
- VGG19 є досить глибокою моделлю, що призводить до великої кількості параметрів. У VGG19 майже 143 мільйони параметрів, що призводить до високого обчислювального навантаження під час навчання та використання.

Третя модель ResNet50 [26] є однією з моделей глибокого навчання для класифікації зображень. Вона базується на архітектурі ResNet (Residual Network) і має 50 шарів. ResNet50 став одним із важливих проривів у галузі комп'ютерного зору, оскільки дозволив зменшити проблему зникаючих градієнтів і додав можливість навчати глибокі нейронні мережі з багатьма шарами.

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 49   |

Основна ідея ResNet полягає у використанні "блоків з нейронними залишками" (residual blocks) для побудови моделі. В звичайних нейронних мережах, коли дані проходять через кожний шар, можуть виникати проблеми зі зникаючими градієнтами, що ускладнює навчання глибоких мереж. У ResNet це вирішується за допомогою "шляхів перехідних з'єднань" (skip connections), які дозволяють пропускати деякі шари та додають нейронні залишки (residuals) напряму до наступних шарів. Таким чином, градієнти можуть легше поширюватись назад через мережу, сприяючи кращому навчанню.

ResNet50 складається з 50 шарів, включаючи згорткові, пакетну нормалізацію (batch normalization), активаційні функції і повнозв'язні шари.

Повний код для InceptionV3, VGG19, ResNet50 знаходяться в додатках А, Б, В відповідно.

### 3.4 Аналіз результатів

У попередньому підрозділі коротко було описано три моделі – InceptionV3, VGG19 та ResNet50. Після навчання кожної з них було сформовано порівняльні таблиці метрик якості моделей для тестових даних, таблиці 3.1, 3.2, 3.3 відповідно.

Розглянемо результати першої моделі – InceptionV3:

Таблиця 3.1 – Метрики навчання InceptionV3

| Loss | Accuracy | Auc  | F1-score |
|------|----------|------|----------|
| 0.34 | 0.88     | 0.98 | 0.88     |

Результати для другої – VGG19 (табл. 3.2) та третьої моделей – ResNet50 (табл. 3.3):

Таблиця 3.2 – Метрики навчання VGG19

| Loss | Accuracy | Auc  | F1-score |
|------|----------|------|----------|
| 0.68 | 0.7      | 0.92 | 0.62     |

Таблиця 3.3 – Метрики навчання ResNet50

| Loss | Accuracy | Auc  | F1-score |
|------|----------|------|----------|
| 0.71 | 0.56     | 0.86 | 0.44     |

Отже, як бачимо, найбільшу точність показала модель InceptionV3, на відміну від інших двох запропонованих моделей.

### Висновки до розділу 3

Отже, при розробці згорткової нейронної мережі для задачі класифікації хвороби Альцгеймера було використано датасет – Alzheimer'sDataset ( 4 classofImages), з платформи Kaggle. Для даної задачі було побудовано три популярні згорткові мережі – InceptionV3, VGG19 та ResNet50. На прикладі моделі InceptionV3 було розглянуто базову структуру архітектури ЗНМ та детально описано код для навчання моделі. Після розробки, було проведено аналіз та порівняння результатів метрик на тестовій даних кожної моделі, та визначено ту, що найкраще підходить до вирішення даної задачі – InceptionV3.

## РОЗДІЛ 4

### ОХОРОНА ПРАЦІ

#### 4.1 Вступ

Дипломна робота була проведена на ТОВ «Скандіагностика».

Даний розділ дипломної роботи виконується за планом №3, оскільки його метою є розробка програмного забезпечення, яке використовує згорткові нейронні мережі для автоматизованого виявлення ранніх стадій хвороби Альцгеймера шляхом аналізу МРТ-зображень мозку пацієнтів. Це передбачає використання наявних приладів та вирішення питань, пов'язаних з організацією проведення досліджень.

Метою цього розділу є аналіз впливу шкідливих факторів та умов праці на виробництві, а також розробка відповідних заходів щодо усунення негативного впливу на персонал.

У роботі було розглянуто МРТ «Toshiba Vantage Titan 1.5T» та його використання для отримання МРТ-зображень, які були використані в дипломній роботі.

#### 4.2 Характеристика приміщення та робочого місця

Перш за все слід зазначити що блок приміщень для МРТ складається з трьох складових:

1. Пультова кімната або операторська – кімната, де знаходиться оператор, який здійснює сканування і візуальний контроль через оглядове вікно.
2. Технічна кімната – в ній розміщуються шафи МРТ, градієнтний і радіочастотні підсилювачі, шафа обробки даних, кріокомпресор.
3. Сканерна кімната – приміщення, де встановлюється томограф.

Загальні розміри блоку повинні бути приблизно 45-65м<sup>2</sup>. Нижче наведено характеристики приміщення, специфікація технологічного обладнання та оснащення в якому проводяться дослідження за завданням (табл. 4.1).

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   | 52   |
| Изм. | Лист | № докум. | Підпис | Дата |                   |      |

Таблиця 4.1 – Характеристика приміщення

| №                      | Найменування                                                       | Основні характеристики                                                                                                                                                                                                                                                                                  | Кількість | Позиція на рисунку |
|------------------------|--------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|--------------------|
| 1.                     | Параметри приміщення:<br>1) Пультова<br>2) Технічна<br>3) Сканерна | 5000*2500*3000; S=12,5м <sup>2</sup> ; V=37,5м <sup>3</sup><br>4000*2500*3000; S=10м <sup>2</sup> ; V=30м <sup>3</sup><br>5000*6000*3000; S=30м <sup>2</sup> ; V=90м <sup>3</sup>                                                                                                                       |           | -                  |
| 2.                     | Кількість працюючих                                                | оператор, рентгенолог                                                                                                                                                                                                                                                                                   | 2         | -                  |
| 3.                     | Штучне освітлення                                                  | ЛПБ 80                                                                                                                                                                                                                                                                                                  | 12        | -                  |
| 4.                     | Штучна вентиляція                                                  | Спеціальний шлюз для охолодження гелію (чиллер), радіочастотний фільтр та повітроводи з немагнітної нержавіючої сталі, загальна вентиляція                                                                                                                                                              |           | -                  |
| 5.                     | Опалення                                                           | Центральне водяне опалення низького тиску                                                                                                                                                                                                                                                               |           | -                  |
| 6.                     | Підлога                                                            | Електропровідний матеріал (клітка фарадея), бетон                                                                                                                                                                                                                                                       |           | -                  |
| 7.                     | Стіни                                                              | Електропровідний матеріал (клітка фарадея), бетон                                                                                                                                                                                                                                                       |           | -                  |
| Обладнання і оснащення |                                                                    |                                                                                                                                                                                                                                                                                                         |           |                    |
| 8.                     | Toshiba Vantage Titan 1.5T                                         | Габарити – 4500 (з ліжком)*2400*2200;<br>Живлення – 3х фазне 380В;<br>Тип – закритий;<br>Вага та тип магніту – 4 тони надпровідний;<br>Кількість каналу прийому і передачі – 16/32;<br>Поле огляду – 55*55*50;<br>Швидкість наростання градієнта – 120 мТ/м/мкс;<br>Макс. Величина градієнтів – 30мТ/м; | 1         | 1                  |
| 9.                     | Стіл для пацієнта МРТ апарату Toshiba Vantage Titan 1.5T           | Розміри - 2230*680*934; робочий хід – 2190 по горизонталі, 633 по вертикалі; максимальна вага пацієнта – 130 кг                                                                                                                                                                                         | 1         | 7                  |
| 10.                    | Крісло робоче Barsky Mesh BM-02                                    | Розміри - 440*420*105; матеріал – корпус – пластик, наповнювач – паралон, обшивка - текстиль                                                                                                                                                                                                            | 1         | 2                  |
| 10.                    | Стіл                                                               | Розміри – 1200*700*700; матеріал – стільниця - дерево (дуб), опори - ДСП                                                                                                                                                                                                                                | 1         | 3                  |
| 10.                    | Шафа                                                               | Розміри – 700*500*1600; матеріал – пластик, покриття – полімерно-порошкове                                                                                                                                                                                                                              | 2         | 4                  |
| 11.                    | Пульт управління                                                   | Комп'ютер з спеціальним програмним забезпеченням (M-Power, WorkFlow Driven Application (WFDA)) -                                                                                                                                                                                                        | 1         | 5                  |

|  |  |                                                         |  |  |
|--|--|---------------------------------------------------------|--|--|
|  |  | DellG3 3779 (i5-8300H/RAM 16GBDDR4/SSD 128GB + HDD 1TB) |  |  |
|--|--|---------------------------------------------------------|--|--|

#### Продовження таблиці 4.1 – Характеристика приміщення

|     |                        |                                     |   |   |
|-----|------------------------|-------------------------------------|---|---|
| 12. | Термографічний принтер | DRYSTAR 5302; розміри – 720*715*530 | 1 | 6 |
| 13. | Вогнегасник            | Розміри – 600*150*150               | 2 | - |
| 14. | Пожежний сповіслювач   | Розміри – 200*200                   | 1 | - |

На рисунку 4.1 показано план блоку:

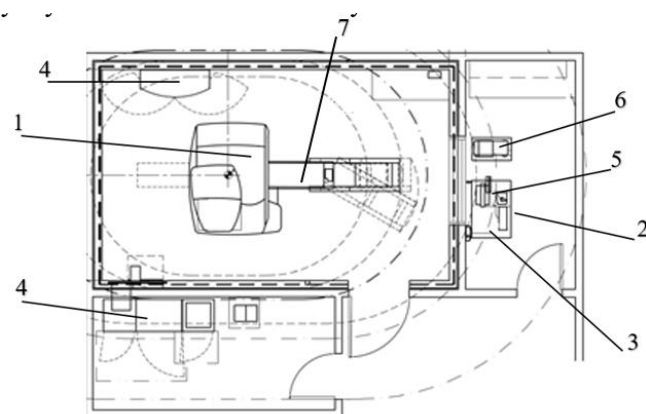


Рисунок 4.1 – План робочого приміщення

Встановлення на відповідність вимогам нормативних документів об'єму і площі приміщення на 1 працівника та місця розташування технологічного обладнання наведено в табл. 4.2

Таблиця 4.2 – Порівняння нормативних параметрів

| Параметр приміщення                                                     | Реальне значення                | Нормативне значення                                                                        |
|-------------------------------------------------------------------------|---------------------------------|--------------------------------------------------------------------------------------------|
| Площа S, м <sup>2</sup><br>1) Пультова:<br>2) Технічна:<br>3) Сканерна: | 12,5<br>10<br>30                | Не менше 10<br>Не менше 10<br>Не менше 24                                                  |
| Розміщення                                                              | Прибудова лікарні               | Нежитлове приміщення, над кабінетом МРТ не може знаходитися приміщення з високою вологістю |
| Захист від магнітних хвиль                                              | Клітка Фарадея                  | Клітка Фарадея                                                                             |
| Оснащення                                                               | Немагнітні нержавіючі матеріали | Немагнітні матеріали                                                                       |

### Продовження таблиці 4.2 – Порівняння нормативних параметрів

|                                             |      |              |
|---------------------------------------------|------|--------------|
| Мінімальна ширина проходу                   | 1,2  | Не менше 1,2 |
| Площа на одну людину $S$ , м <sup>2</sup>   | 17,5 | Не менше 4,5 |
| Об'єм на одну людину $V$ , м <sup>3</sup>   | 52,5 | 15           |
| Відстань між робочим місцем та стіною, м    | 1    | Не менше 1   |
| Мінімальна відстань між робочими місцями, м | 2    | Не менше 1,2 |

Базуючись на даних з таблиці 4.2, можна зробити висновок, що кабінет відповідає нормативним параметрам, встановленим для цього типу приміщення. Вказані параметри кабінетів відповідають вимогам технічно-санітарних норм, що забезпечує безпечність та комфортність під час праці.

### 4.3 Оцінка небезпечних і шкідливих виробничих факторів

Відповідно до ДСТ 12.0.003-2014 [27], всі небезпечні та шкідливі чинники класифікуються на 4 групи. В таблиці 4.3 наведені результати з оцінки присутності чи відсутності цих чинників при виконанні роботи з тестування.

Таблиця 4.3 – Небезпечні та шкідливі виробничі фактори

| Група небезпечних факторів | Перелік основних чинників     |
|----------------------------|-------------------------------|
| Фізичні                    | Магнітне поле; електробезпека |
| Хімічні                    | Контрастні речовини           |
| Біологічні                 | Відсутні                      |
| Психофізіологічні          | Клаустрофобія                 |

Результати таблиці 4.3 показують фізичні, хімічні та біологічні небезпечні та шкідливі фактори. Найбільш вплив з яких становить магнітне поле та електробезпека, саме вони будуть розглянуті в роботі.

#### 4.3.1 Магнітне поле

Принцип роботи магнітно-резонансного томографа полягає в активації енергії протонів водню, що є у всіх тканинах людського організму. Під дією магнітного поля томографа енергію протонів водню можна зафіксувати. Інформація про джерела магнітного поля та пов'язані з ним небезпеки наведено в табл. 4.4.

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 55   |

Таблиця 4.4 – Фактори і наслідки небезпек від джерел випромінювання

| №  | Найменування обладнання            | Джерело небезпеки | Причини небезпеки                                                                                                                            | Наслідки небезпеки                                 |
|----|------------------------------------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| 1. | MPT<br>ToshibaVantageTitan<br>1.5T | Магнітне поле     | Наявність металевих (феромагнітних) конструкцій або речовин – судинні кліпси, кардіостимулятори, ортопедичні або слухові протези, татуювання | Тяжкі наслідки (переміщення) для здоров'я пацієнта |

Таблиця 4.5 – Реальні і нормативні значення показників

| №  | Фактор небезпеки | Реальне значення | Нормативне значення |
|----|------------------|------------------|---------------------|
| 1. | Магнітне поле    | 1,5 Т            | 0,2-3 Т             |

Для мінімізації шкідливих наслідків у таблиці 4.6 наведено заходи та засоби захисту від дії магнітного поля.

Таблиця 4.6 – Засоби і заходи захисту від впливу магнітного поля

| №  | Група номенклатурних заходів з ОП | Вид заходу                                                                        | Критерій вибору                  |
|----|-----------------------------------|-----------------------------------------------------------------------------------|----------------------------------|
| 1. | Технічні заходи                   | Екранування приміщення, використання екранованих кабелів                          | Захист від магнітного поля       |
| 2. | Організаційні заходи              | Заборона наявності металевих предметів у зоні МРТ                                 | Усвідомлення принципів безпеки   |
| 3. | Режимні заходи                    | Використання спеціальних протоколів безпеки для пацієнтів і персоналу             | Усвідомлення принципів безпеки   |
| 4. | Експлуатаційні заходи             | Регулярна перевірка технічного стану МРТ, проведення профілактичних обслуговувань | Контроль працездатності приладів |

#### 4.3.2 Електронебезпека

Кабінет магнітно-резонансної томографії містить прилади, які споживають електроенергію. В таблиці 4.7 наведені фактори та наслідки електронебезпеки.

Таблиця 4.7 – Фактори і наслідки електронебезпеки

| №  | Найменування обладнання            | Джерело небезпеки | Причини небезпеки                                                    | Наслідки небезпеки                                 |
|----|------------------------------------|-------------------|----------------------------------------------------------------------|----------------------------------------------------|
| 1. | MPT<br>ToshibaVantageTitan<br>1.5T | Мережевий кабель  | Електричний струм, коротке замикання, несправність електрообладнання | Пожежа, електрошок, вибух, пошкодження обладнання, |



Продовження таблиці 4.7 – Фактори і наслідки електробезпеки

|    |                                        |                  |                                                                         |                                                                                                      |
|----|----------------------------------------|------------------|-------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| 1. |                                        |                  |                                                                         | травми персоналу та пацієнтів                                                                        |
| 2. | Стационарний комп'ютер<br>DellG3 3779  | Мережевий кабель | Електричний струм, некоректне підключення, перевантаження, несправність | Пожежа, електрошок, втрата даних, відмова обладнання, нестабільна робота систем                      |
| 3. | Термографічний принтер<br>DRYSTAR 5302 | Мережевий кабель | Електричний струм, некоректне підключення, несправність обладнання      | Пожежа, електрошок, нестабільна робота систем, негативний вплив на клімат контрольованого середовища |

Таблиця 4.8 – Реальні і нормативні значення показників

| №  | Найменування електроприладу                | Реальне значення | Нормативне значення |
|----|--------------------------------------------|------------------|---------------------|
| 1. | Постійний струм                            | 350 мА           | до 500 мА           |
| 2. | Змінний струм в мережевих кабелях живлення | 11-14 А          | до 16 А             |

Засоби захисту від небезпеки ураження електричним струмом в даному приміщенні наведено в таблиці 4.9.

Таблиця 4.9 – Засоби і заходи захисту від електробезпеки

| №  | Група номенклатурних заходів з ОП | Вид заходу                                                                                                                                           | Критерій вибору                                                       |
|----|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| 1. | Технічні заходи                   | Заземлення та використання плавких запобіжників                                                                                                      | Електроізоляція                                                       |
| 2. | Організаційні заходи              | Інструктаж з питань електробезпеки                                                                                                                   | Навчання з питань безпеки при експлуатації обладнання                 |
| 3. | Режимні заходи                    | Використання спеціальних протоколів безпеки для пацієнтів і персоналу                                                                                | Усвідомлення принципів безпеки                                        |
| 4. | Експлуатаційні заходи             | Своєчасна заміна деталей техніки, що вийшли з ладу чи пошкоджені, перевірка електричних апаратів за допомогою мегомметра не менше одного разу на рік | Контроль працездатності приладів, контроль показників стрибків струму |

## Висновки до розділу 4

В даному розділі було проаналізовано умови праці в приміщенні для МРТ, згідно розрахунків вони задовольняють нормативні значення. Також розглянуто основні види небезпек такі як електробезпека та дія магнітного поля. Для кожної з небезпек розроблено ряд заходів для зменшення ризику їх виникнення. Кабінет МРТ відповідає вимогам електро- та пожежної безпеки.

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   | 58   |
| Изм. | Лист | № докум. | Підпис | Дата |                   |      |

## ВИСНОВКИ

В цій дипломній роботі було проведено дослідження та аналіз проблеми діагностування хвороби Альцгеймера, що є однією з основних проблем сучасної медицини. Було розглянуто сутність хвороби, її стадії, причини, розвиток та клінічні типи. Особлива увага була приділена біомаркерам структурних змін головного мозку, що дозволяють виявити хворобу на МРТ-зображеннях пацієнтів.

Проведений аналіз існуючих методів автоматизованого розпізнавання хвороби Альцгеймера на МРТ-зображеннях підтвердив, що найкращим підходом для вирішення цієї проблеми є використання згорткових нейронних мереж. Згорткові нейронні мережі є ефективними у розпізнаванні об'єктів і обробці зображень, що є важливими аспектами при аналізі МРТ-зображень головного мозку.

У контексті розробки програмного забезпечення для розпізнавання хвороби Альцгеймера на МРТ-зображеннях, була сформульована постановка задачі, яка полягає в розробці згорткової нейронної мережі для класифікації зображень. Для цього був використаний датасет Alzheimer's Dataset (4 class of Images) з платформи Kaggle, а також побудовані три популярні згорткові мережі: InceptionV3, VGG19 та ResNet50.

Після порівняння результатів метрик на тестових даних кожної моделі, було встановлено, що InceptionV3 демонструє найкращі показники та найбільш підходить для розпізнавання хвороби Альцгеймера на МРТ-зображеннях.

Отже, розроблена згорткова нейронна мережа на основі моделі InceptionV3 може бути використана для раннього виявлення хвороби Альцгеймера шляхом аналізу МРТ-зображень головного мозку пацієнтів. Результати даної дипломної роботи мають потенціал покращити точність діагнозу та сприяти вчасному початку лікування цієї нейродегенеративної хвороби.

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 59   |

## СПИСОК ЛІТЕРАТУРИ

- 1) Yadav S. S., Jadhav S. M. Deep convolutional neural network based medical image classification for disease diagnosis. Journal of Big Data. 2019. Vol. 6, no. 1. DOI: <https://doi.org/10.1186/s40537-019-0276-2>
- 2) Lakhani P., Sundaram B. Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. Radiology. 2017. Vol. 284, no. 2. P. 574–582. DOI: <https://doi.org/10.1148/radiol.2017162326>
- 3) Reproducible and interpretable deep learning for the diagnosis, prognosis and subtyping of Alzheimer’s disease from neuroimaging data – Wikipedia: веб-сайт. URL: [https://en.wikipedia.org/wiki/Comparison\\_of\\_code\\_generation\\_tools](https://en.wikipedia.org/wiki/Comparison_of_code_generation_tools) (дата звернення: 03.05.2023).
- 4) Computer-Aided Diagnosis of Dementia based on structural MRI data. – Grand Challenge: веб-сайт. URL: <https://caddementia.grand-challenge.org/> (дата звернення: 05.05.2023).
- 5) Deep Learning with Convolutional Neural Network for Differentiation of Liver Masses at Dynamic Contrast-enhanced CT: A Preliminary Study / K. Yasaka et al. Radiology. 2018. Vol. 286, no. 3. P. 887–896.
- 6) Chris Albon, Machine learning with Python Cookbook/ Chris Albon – Mar 2018. 384 p.
- 7) Deep Convolutional Neural Networks for Automated Diagnosis of Alzheimer’s Disease and Mild Cognitive Impairment Using 3D Brain MRI – CodeSmith Researchgate: веб-сайт. URL: [https://www.researchgate.net/figure/Deep-CNN-architecture-used-for-the-proposed-Alzheimers-Disease-diagnosis-framework\\_fig5\\_332212620](https://www.researchgate.net/figure/Deep-CNN-architecture-used-for-the-proposed-Alzheimers-Disease-diagnosis-framework_fig5_332212620) (дата звернення: 06.05.2023).
- 8) Convolutional neural network – Wikipedia: веб-сайт. URL: [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network#:~:text=A%20CNN%20architecture%20is%20formed,of%20layers%20are%20commonly%20used](https://en.wikipedia.org/wiki/Convolutional_neural_network#:~:text=A%20CNN%20architecture%20is%20formed,of%20layers%20are%20commonly%20used) (дата звернення: 10.05.2023).

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 60   |

- 9) Activation Functions in Neural Networks – Towards Data Science: веб-сайт. URL: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6> (дата звернення: 11.05.2023).
- 10) Нейронні мережі – шлях до глибинного навчання – Codeguida: веб-сайт. URL: <https://codeguida.com/post/739> (дата звернення: 18.05.2023).
- 11) Татарин М. М., Якимович Г. М., Панкевич Н. А. Хвороба Альцгеймера (Бібліографічний покажчик): Івано-Франківськ 2021. 15 с.
- 12) Хвороба Альцгеймера: симптоми, лікування. Веб-сайт. URL: <https://medikom.ua/bolezni-alcgejmira-simptomy-lechenie/#Etiologiya> (дата звернення: 18.05.2023).
- 13) Dickson, W. Identification of normal and pathological aging in prospectively studied nondemented elderly humans/ Wan Dickson. – Feb 1992. 179 – 189 p.
- 14) The Radiology Assistant: Dementia - Role of MRI. The Radiology Assistant : Home. URL: <https://radiologyassistant.nl/neuroradiology/dementia/role-of-mri> (date of access: 14.05.2023)
- 15) Олещенко Л.М. Машинне навчання: навч. посіб. / Київ. нац. ун-т ім. Ігоря Сікорського. Київ: 2022. 92 с.
- 16) Метод опорних векторів – Wikipedia: веб-сайт. URL: [https://upload.wikimedia.org/wikipedia/commons/thumb/7/72/SVM\\_margin.png/1024px-SVM\\_margin.png](https://upload.wikimedia.org/wikipedia/commons/thumb/7/72/SVM_margin.png/1024px-SVM_margin.png) (дата звернення: 23.05.2023).
- 17) Згорткові нейронні мережі – YouTube: веб-сайт. URL: [https://www.youtube.com/watch?v=Qkq0VLuw5Vg&list=PL\\_ug4sOLLpQiHmRGMki4ID6By9nPzAH0A&index=23](https://www.youtube.com/watch?v=Qkq0VLuw5Vg&list=PL_ug4sOLLpQiHmRGMki4ID6By9nPzAH0A&index=23) (дата звернення: 23.05.2023).
- 18) Тимченко А.О, Скачков В.О. Функція активації нейрона ReLu: Житомирський державний технологічний університет. Житомир: 2018. 1 с.
- 19) Навчання штучної нейронної мережі методом зворотного розповсюдження помилки – Studfile: веб-сайт. URL: <https://studfile.net/preview/5461803/> (дата звернення: 23.05.2023).
- 20) М.А. Новотарський, Б.Б. Нестеренко. Штучні нейронні мережі: Обчислення: Інститут математики НАН України. Київ: 2004. 207 с.

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 61   |

- 21) Зінченко О. В., Звенігородський О. С., Кисіль Т. М. Згорткові нейронні мережі для вирішення задач комп'ютерного зору: Державний університет телекомунікацій. Київ: 2022. 73 с.
- 22) Класифікація зображень за допомогою ЗНМ – Codeguida: веб-сайт. URL: <https://codeguida.com/post/1400>(дата звернення: 25.05.2023).
- 23) WhatisKaggle – Kaggle: веб-сайт. URL: <https://www.datacamp.com/blog/what-is-kaggle>(дата звернення: 25.05.2023).
- 24) Inception-v3 for flower classification – Ieexplore:веб-сайт. URL: <https://ieeexplore.ieee.org/abstract/document/7984661>(дата звернення: 26.05.2023).
- 25) Exposing Computer Generated Images by Eye's Region Classification via Transfer Learning of VGG19 CNN –Ieexplore:веб-сайт. URL: <https://ieeexplore.ieee.org/abstract/document/8260745>(дата звернення: 26.05.2023).
- 26) Detecting Affect States Using VGG16, ResNet50 and SE-ResNet50 Networks – Link.Springer: веб-сайт. URL: <https://link.springer.com/article/10.1007/s42979-020-0114-9>(дата звернення: 26.05.2023).
- 27) Закон України «Про небезпечні та шкідливі виробничі фактори. Класифікація»/ Законодавство України про охорону праці. – К. Нова редакція 2002 р.

## Додаток А

### Лістингкоду для моделі InceptionV3

```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import os
from distutils.dir_util import copy_tree, remove_tree
from random import randint
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import tensorflow_addons as tfa
from keras.utils.vis_utils import plot_model
from tensorflow.keras import Sequential, Input
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.layers import Conv2D, Flatten
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.preprocessing.image import ImageDataGenerator as IDG
from tensorflow.keras.layers import SeparableConv2D, BatchNormalization,
MaxPool2D
print("TensorFlow Version:", tf.__version__)

##Data Pre-Processing

base_dir = "Alzheimer_s Dataset/"
root_dir = "/"
test_dir = base_dir + "test/"
train_dir = base_dir + "train/"
work_dir = root_dir + "dataset/"
```

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 63   |

```

# if os.path.exists(work_dir):
#     remove_tree(work_dir)
#
# os.mkdir(work_dir)
# copy_tree(train_dir, work_dir)
# copy_tree(test_dir, work_dir)
print("Working Directory Contents:", os.listdir(work_dir))
WORK_DIR = './dataset/'
CLASSES = ['MildDemented',
            'ModerateDemented',
            'NonDemented',
            'VeryMildDemented']
IMG_SIZE = 176
IMAGE_SIZE = [176, 176]
DIM = (IMG_SIZE, IMG_SIZE)

# Performing Image Augmentation to have more data samples
ZOOM = [.99, 1.01]
BRIGHT_RANGE = [0.8, 1.2]
HORZ_FLIP = True
FILL_MODE = "constant"
DATA_FORMAT = "channels_last"
work_dr = IDG(rescale=1. / 255, brightness_range=BRIGHT_RANGE,
              zoom_range=ZOOM, data_format=DATA_FORMAT,
              fill_mode=FILL_MODE, horizontal_flip=HORZ_FLIP)
train_data_gen=work_dr.flow_from_directory(directory=WORK_DIR,
target_size=DIM, batch_size=6500, shuffle=False)

# Display Train Images
def show_images(generator, y_pred=None):

```

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 64   |



```

"""
Input: An image generator, predicted labels (optional)
Output: Displays a grid of 9 images with labels
"""

# get image labels
labels = dict(zip([0, 1, 2, 3], CLASSES))

# get a batch of images
x, y = generator.next()

# display a grid of 9 images
plt.figure(figsize=(10, 10))

if y_pred is None:
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        idx = randint(0, 6400)
        plt.imshow(x[idx])
        plt.axis("off")
        plt.title("Class: {}".format(labels[np.argmax(y[idx])]))
else:
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(x[i])
        plt.axis("off")
        plt.title("Actual: {} \n Predicted: {}".format(labels[np.argmax(y[i])],
labels[y_pred[i]]))
    plt.show()

show_images(train_data_gen)

#Retrieving the data from the ImageDataGenerator iterator
train_data, train_labels = train_data_gen.next()

```

```

#Getting to know the dimensions of our dataset
print(train_data.shape, train_labels.shape)

#Performing over-sampling of the data, since the classes are imbalanced
sm = SMOTE(random_state=42)
train_data, train_labels = sm.fit_resample(train_data.reshape(-1, IMG_SIZE *
IMG_SIZE * 3), train_labels)
train_data = train_data.reshape(-1, IMG_SIZE, IMG_SIZE, 3)
print(train_data.shape, train_labels.shape)

#Splitting the data into train, test, and validation sets
train_data, test_data, train_labels, test_labels = train_test_split(train_data, train_labels,
test_size = 0.2, random_state=42)
train_data, val_data, train_labels, val_labels = train_test_split(train_data, train_labels,
test_size = 0.2, random_state=42)

##Constructing a Convolutional Neural Network Architecture

def conv_block(filters, act='relu'):
    """Defining a Convolutional NN block for a Sequential CNN model. """
    block = Sequential()
    block.add(Conv2D(filters, 3, activation=act, padding='same'))
    block.add(Conv2D(filters, 3, activation=act, padding='same'))
    block.add(BatchNormalization())
    block.add(MaxPool2D())
    return block

def dense_block(units, dropout_rate, act='relu'):
    """Defining a Dense NN block for a Sequential CNN model. """
    block = Sequential()

```

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 66   |

```

block.add(Dense(units, activation=act))
block.add(BatchNormalization())
block.add(Dropout(dropout_rate))
return block

```

```
def construct_model(act='relu'):
```

```

    """Constructing a Sequential CNN architecture for performing the classification
    task. """

```

```

    model = Sequential([
        Input(shape=(*IMAGE_SIZE, 3)),
        Conv2D(16, 3, activation=act, padding='same'),
        Conv2D(16, 3, activation=act, padding='same'),
        MaxPool2D(),
        conv_block(32),
        conv_block(64),
        conv_block(128),
        Dropout(0.2),
        conv_block(256),
        Dropout(0.2),
        Flatten(),
        dense_block(512, 0.7),
        dense_block(128, 0.5),
        dense_block(64, 0.3),
        Dense(4, activation='softmax')
    ], name="cnn_model")
    return model

```

```

# Defining a custom callback function to stop training our model when accuracy goes
above 99%

```

```
class MyCallback(tf.keras.callbacks.Callback):
```

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 67   |

```

def on_epoch_end(self, epoch, logs={}):
    if logs.get('val_acc') > 0.99:
        print("\nReached accuracy threshold! Terminating training.")
self.model.stop_training = True
my_callback = MyCallback()

# EarlyStopping callback to make sure model is always learning
early_stopping = EarlyStopping(monitor='val_loss', patience=2)

# Defining other parameters for our CNN model
model = construct_model()
METRICS = [tf.keras.metrics.CategoricalAccuracy(name='acc'),
            tf.keras.metrics.AUC(name='auc'),
            tfa.metrics.F1Score(num_classes=4)]
CALLBACKS = [my_callback]

model.compile(optimizer='adam',
              loss=tf.losses.CategoricalCrossentropy(),
              metrics=METRICS)

model.summary()

##Training & Testing the Model
#Fit the training data to the model and validate it using the validation data
EPOCHS = 25
history = model.fit(train_data, train_labels, validation_data=(val_data, val_labels),
                    callbacks=CALLBACKS, epochs=EPOCHS)

#Evaluating the model on the data
test_scores = model.evaluate(test_data, test_labels)

```

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   | 68   |
| Изм. | Лист | № докум. | Підпис | Дата |                   |      |

```

#print("Training Accuracy: %.2f%%"%(train_scores[1] * 100))
#print("Validation Accuracy: %.2f%%"%(val_scores[1] * 100))
print("Testing Accuracy: %.2f%%"%(test_scores[1] * 100))

#Predicting the test data
pred_labels = model.predict(test_data)

#Print the classification report of the tested data
#Since the labels are softmax arrays, we need to roundoff to have it in the form of 0s
and 1s,
#similar to the test_labels
def roundoff(arr):
    """To round off according to the argmax of each predicted label array. """
    arr[np.argmax(arr) != 0] = 0
    arr[np.argmax(arr) == 0] = 1
    return arr
for labels in pred_labels:
    labels = roundoff(labels)
print(classification_report(test_labels, pred_labels, target_names=CLASSES))

#Saving the model for future use
# model_dir = work_dir + "alzheimer_cnn_model"
model_dir = "InceptionV3/alzheimer_cnn_model"
model.save(model_dir, save_format='h5')
os.listdir(work_dir)
pretrained_model = tf.keras.models.load_model(model_dir)

# Function to load and preprocess a single image
def load_and_preprocess_image(image_path):

```

```

image = tf.keras.preprocessing.image.load_img(image_path,
target_size=IMAGE_SIZE)

image = tf.keras.preprocessing.image.img_to_array(image)

image = image / 255.0 # Normalize image

image = np.expand_dims(image, axis=0) # Add batch dimension

return image

# Function to predict class and confidence
def predict_image(image_path):
    image = load_and_preprocess_image(image_path)
    predictions = model.predict(image)
    predicted_class_index = np.argmax(predictions)
    predicted_class = CLASSES[predicted_class_index]
    confidence = np.max(predictions) * 100
    return predicted_class, confidence

# Path to the image you want to test
image_path = 'Alzheimer_s Dataset/test/ModerateDemented/29 (2).jpg'

# Display the predicted class and confidence
predicted_class, confidence = predict_image(image_path)
print("Predicted Class:", predicted_class)
print("Confidence:", confidence)

```

Додаток Б  
Лістингкоду для моделі VGG19

```
import matplotlib.pyplot as plt
import tensorflow_addons as tfa
import numpy as np
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow import keras
import tensorflow as tf
from tensorflow.keras.applications.vgg19 import VGG19 # VGG19

#define image dataset
from tensorflow.keras.callbacks import ModelCheckpoint
image_generator=ImageDataGenerator(rescale=1/255, validation_split=0)
train_dataset = image_generator.flow_from_directory(batch_size=16,
                                                    directory='Alzheimer_s Dataset/train',
                                                    shuffle=True,
                                                    target_size=(224,224),
                                                    subset="training",
                                                    class_mode='categorical')

image_generator = ImageDataGenerator(rescale=1/255,validation_split=0.2)
validation_dataset = image_generator.flow_from_directory(batch_size=16,
                                                         directory='Alzheimer_s Dataset/test',
                                                         shuffle=True,
                                                         target_size=(224,224),
                                                         class_mode='categorical')
```

```

submission = image_generator.flow_from_directory(
    directory='Alzheimer_s Dataset/test',
    shuffle=False,
    subset="validation",
    target_size=(224,224),
    class_mode=None)

#show flwoers for the first batch
batch_1_img = train_dataset[0]
for i in range(0,2):
    img = batch_1_img[0][i]
    lab = batch_1_img[1][i]
    plt.imshow(img)
    plt.title(lab)
    plt.axis('off')
    plt.show()

vgg = VGG19(input_shape=(224,224,3), weights='imagenet', include_top=False)
for layer in vgg.layers:
    layer.trainable = False
x = Flatten()(vgg.output)
prediction = Dense(4, activation='softmax')(x)
modelvgg = Model(inputs=vgg.input, outputs=prediction)
modelvgg.summary()
modelvgg.compile(optimizer='adam',
    loss=tf.losses.CategoricalCrossentropy(),
    metrics=[keras.metrics.AUC(name='auc'),'acc', tfa.metrics.F1Score(num_classes=4)])

# Define the checkpoint filepath
checkpoint_filepath = 'VGG19/model_checkpoint.h5'

```



```

# Define the ModelCheckpoint callback
checkpoint_callback = ModelCheckpoint(
    filepath=checkpoint_filepath,
    monitor='val_loss',
    mode='min',
    save_best_only=True,
    save_weights_only=False,
    verbose=1
)

callback = keras.callbacks.EarlyStopping(monitor='val_loss',
                                         patience=8,
                                         restore_best_weights=True)

modelvgg.fit(train_dataset, epochs=20, validation_data=validation_dataset,
             callbacks=callback)

loss, auc, accuracy, f1_score = modelvgg.evaluate(validation_dataset)
print("Loss: ", loss)
print("AUC: ", auc)
print("Accuracy: ", accuracy)
print("F1 Score: ", f1_score)

# Path to the image you want to test
image_path = 'Alzheimer_s Dataset/test/ModerateDemented/29 (2).jpg'

# Function to load and preprocess a single image
def load_and_preprocess_image(image_path):
    image = tf.keras.preprocessing.image.load_img(image_path, target_size=(224,
224))

    image = tf.keras.preprocessing.image.img_to_array(image)
    image = image / 255.0 # Normalize image
    image = np.expand_dims(image, axis=0) # Add batch dimension

```

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 73   |

```

return image

# Load and preprocess the image
input_image = load_and_preprocess_image(image_path)

# Make predictions on the image
predictions = modelvgg.predict(input_image)
class_index = np.argmax(predictions[0])
class_confidence = predictions[0][class_index]

# Get the class label from the dataset
class_label = train_dataset.class_indices
class_label = dict((v, k) for k, v in class_label.items())
class_name = class_label[class_index]

# Print the predicted class and confidence
print("Predicted Class:", class_name)
print("Confidence:", class_confidence)

```

## Додаток В

### Лістинг коду для моделі ResNet50

```
import matplotlib.pyplot as plt
import tensorflow_addons as tfa
from tensorflow.keras.applications.resnet50 import ResNet50
import numpy as np
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow import keras
import tensorflow as tf

#define image dataset
from tensorflow.keras.callbacks import ModelCheckpoint
image_generator = ImageDataGenerator(rescale=1/255, validation_split=0)
train_dataset = image_generator.flow_from_directory(batch_size=16,
                                                    directory='Alzheimer_s Dataset/train',
                                                    shuffle=True,
                                                    target_size=(176,176),
                                                    subset="training",
                                                    class_mode='categorical')
image_generator = ImageDataGenerator(rescale=1/255, validation_split=0.2)
validation_dataset = image_generator.flow_from_directory(batch_size=16,
                                                         directory='Alzheimer_s Dataset/test',
                                                         shuffle=True,
                                                         target_size=(176,176),
                                                         class_mode='categorical')
```

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   | 75   |
| Изм. | Лист | № докум. | Підпис | Дата |                   |      |

```

submission = image_generator.flow_from_directory(
    directory='Alzheimer_s Dataset/test',
    shuffle=False,
    subset="validation",
    target_size=(176,176),
    class_mode=None)

#show flwoers for the first batch
batch_1_img = train_dataset[0]
for i in range(0,2):
    img = batch_1_img[0][i]
    lab = batch_1_img[1][i]
    plt.imshow(img)
    plt.title(lab)
    plt.axis('off')
    plt.show()

rn = ResNet50(input_shape=(176,176,3), weights='imagenet', include_top=False)
for layer in rn.layers:
    layer.trainable = False
x = Flatten()(rn.output)
prediction = Dense(4, activation='softmax')(x)
modelvgg = Model(inputs=rn.input, outputs=prediction)
modelvgg.compile(optimizer='adam',
    loss=tf.losses.CategoricalCrossentropy(),
    metrics=[keras.metrics.AUC(name='auc'),'acc', tfa.metrics.F1Score(num_classes=4)])
callback = keras.callbacks.EarlyStopping(monitor='val_loss',
    patience=8,
    restore_best_weights=True)

modelvgg.fit(train_dataset, epochs=20, validation_data=validation_dataset,
    callbacks=callback)

```

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 76   |

```

loss, auc, accuracy, f1_score = modelvgg.evaluate(validation_dataset)
print("Loss: ", loss)
print("AUC: ", auc)
print("Accuracy: ", accuracy)
print("F1 Score: ", f1_score)

# Path to the image you want to test
image_path = 'Alzheimer_s Dataset/test/ModerateDemented/29 (2).jpg'

# Function to load and preprocess a single image
def load_and_preprocess_image(image_path):
    image = tf.keras.preprocessing.image.load_img(image_path, target_size=(176,
176))
    image = tf.keras.preprocessing.image.img_to_array(image)
    image = image / 255.0 # Normalize image
    image = np.expand_dims(image, axis=0) # Add batch dimension
    return image

# Load and preprocess the image
input_image = load_and_preprocess_image(image_path)

# Make predictions on the image
predictions = modelvgg.predict(input_image)
class_index = np.argmax(predictions[0])
class_confidence = predictions[0][class_index]

# Get the class label from the dataset
class_label = train_dataset.class_indices
class_label = dict((v, k) for k, v in class_label.items())
class_name = class_label[class_index]

```

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   |      |
| Изм. | Лист | № докум. | Підпис | Дата |                   | 77   |

```
# Print the predicted class and confidence
print("Predicted Class:", class_name)
print("Confidence:", class_confidence)
```

|      |      |          |        |      |                   |      |
|------|------|----------|--------|------|-------------------|------|
|      |      |          |        |      | БМ92.30.3105.2106 | Лист |
|      |      |          |        |      |                   | 78   |
| Изм. | Лист | № докум. | Підпис | Дата |                   |      |