

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Методи виявлення та класифікації пухлин головного мозку за МРТ-
зображеннями з використанням нейронних мереж»**

Виконав студент групи ІПЗм-24-2
ДРЖЕВЕЦЬКИЙ Микита Олегович

Керівник роботи:
ЄФРЕМОВ Юрій Миколайович

Рецензент:
ЧИЖМОТРЯ Олексій Володимирович

Житомир – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»
зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«18» вересня 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ДРЖЕВЕЦЬКИЙ Микита Олегович**

Керівник роботи: **ЄФРЕМОВ Юрій Миколайович**

Тема роботи: **«Методи виявлення та класифікації пухлин головного мозку за МРТ-зображеннями з використанням нейронних мереж»,**
затверджена Наказом закладу вищої освіти від «18» вересня 2025 р., № 491/с
Вихідні дані для роботи: **моделі та методи глибокого навчання і нейронних мереж, призначені для аналізу та класифікації МРТ зображень головного мозку.**

Консультанти з магістерської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Єфремов Ю.М.	22.09.2025р.	22.09.2025р.
2	Єфремов Ю.М.	22.09.2025р.	22.09.2025р.
3	Єфремов Ю.М.	22.09.2025р.	22.09.2025р.

РЕФЕРАТ

Кваліфікаційна робота освітнього ступеня «магістр» присвячена дослідженню та застосуванню методів автоматизованого виявлення і класифікації пухлин головного мозку за магнітно-резонансними томографічними зображеннями з використанням нейронних мереж та методів глибокого навчання.

Пояснювальна записка складається зі вступу, трьох розділів та списку використаної літератури. Пояснювальна записка до кваліфікаційної магістерської роботи містить 67 сторінок, 27 ілюстрацій, 8 таблиць, та 40 посилань на зовнішні джерела.

Метою роботи є підвищення точності автоматизованої класифікації пухлин головного мозку шляхом застосування методів глибокого навчання та трансферного навчання.

У роботі досліджено сучасні підходи до обробки медичних зображень і класифікації МРТ-знімків. Реалізовано дві моделі згорткових нейронних мереж на основі архітектур VGG16 та ResNet50, адаптовані до задачі багатокласової класифікації пухлин головного мозку. Проведено навчання та тестування моделей, виконано порівняльний аналіз їх ефективності. За результатами експериментів встановлено, що модель на основі архітектури ResNet50 демонструє вищу точність і кращу узагальнювальну здатність порівняно з VGG16.

КЛЮЧОВІ СЛОВА: ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, ГЛИБОКЕ НАВЧАННЯ, МРТ-ЗОБРАЖЕННЯ, ПУХЛИНИ ГОЛОВНОГО МОЗКУ, ВИЯВЛЕННЯ, КЛАСИФІКАЦІЯ, RESNET, VGG, ТРАНСФЕРНЕ НАВЧАННЯ, МЕДИЧНА ВІЗУАЛІЗАЦІЯ.

ABSTRACT

The master's qualification work is dedicated to the study and application of methods for automated detection and classification of brain tumors using magnetic resonance imaging (MRI) and employing neural networks and deep learning techniques.

The explanatory note consists of an introduction, three chapters, and a list of references. The explanatory note of the master's thesis contains 67 pages, 27 figures, 8 tables, and 40 references to external sources.

The aim of the work is to improve the accuracy of automated brain tumor classification by applying deep learning and transfer learning methods.

The work investigates modern approaches to medical image processing and MRI image classification. Two convolutional neural network (CNN) models based on VGG16 and ResNet50 architectures were implemented and adapted for the task of multi-class brain tumor classification. The models were trained and tested, and a comparative analysis of their effectiveness was conducted. Experimental results indicate that the ResNet50-based model demonstrates higher accuracy and better generalization ability compared to VGG16.

KEYWORDS: CONVOLUTIONAL NEURAL NETWORKS, DEEP LEARNING, MRI IMAGES, BRAIN TUMORS, DETECTION, CLASSIFICATION, RESNET, VGG, TRANSFER LEARNING, MEDICAL IMAGE ANALYSIS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1. Постановка задачі дослідження	10
1.2. Особливості МРТ-зображень та типи пухлин	11
1.3. Згорткові нейронні мережі та їх застосування у класифікації медичних зображень	15
1.4. Аналіз обраних архітектур (ResNet50, VGG16)	18
1.5. Огляд існуючих програмних рішень для класифікації медичних зображень	23
Висновки до розділу 1	26
РОЗДІЛ 2. МЕТОДИ ТА МОДЕЛІ КЛАСИФІКАЦІЇ ПУХЛИН ГОЛОВНОГО МОЗКУ ЗА МРТ-ЗОБРАЖЕННЯМИ	29
2.1. Загальна методологія класифікації МРТ-зображень та структура дослідження	29
2.2. Формування та адаптація архітектур згорткових нейронних мереж	35
2.3. Підготовка даних та алгоритм навчання моделей	39
2.4. Оцінювання ефективності та аналіз результатів класифікації	47
Висновки до розділу 2	54
РОЗДІЛ 3. ТЕСТУВАННЯ ТА ВАЛІДАЦІЯ МОДЕЛЕЙ КЛАСИФІКАЦІЇ ПУХЛИН ГОЛОВНОГО МОЗКУ	56
3.1. Візуалізація передбачень моделей VGG16 та ResNet50	56
3.2. Тестування моделей у прикладному програмному застосунку	59
3.3. Порівняльний аналіз та інтерпретація результатів роботи моделей VGG16 і ResNet50	63
Висновки до розділу 3	64
ВИСНОВОК.....	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68
ДОДАТКИ.....	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AUC – Area Under the Curve

CNN – Convolutional Neural Networks

MRI – Magnetic Resonance Imaging

ResNet – Residual Network

ROC – Receiver Operating Characteristic

VGG – Visual Geometry Group Network

ВСТУП

Актуальність теми. Пухлини головного мозку належать до найбільш небезпечних патологій центральної нервової системи, оскільки навіть незначні порушення у мозкових структурах можуть спричиняти тяжкі функціональні розлади. Висока смертність, складність морфологічних проявів та значна клінічна варіативність новоутворень зумовлюють потребу у підвищенні точності та швидкості їх діагностики. Епідеміологічні дослідження свідчать, що в Україні рівень захворюваності на злоякісні пухлини головного мозку у 2017–2021 роках становив у середньому близько 3,96 випадка на 100 000 населення на рік, що підтверджує важливість удосконалення діагностичних підходів.

Магнітно-резонансна томографія є ключовим інструментом виявлення та оцінки пухлин головного мозку, проте традиційна експертна інтерпретація МРТ-зображень залишається трудомісткою, суб'єктивною і залежною від кваліфікації лікаря. Висока складність структурних характеристик новоутворень, схожість деяких патологічних ознак та ризик помилкової оцінки меж пухлини створюють потребу у більш точних і автоматизованих підходах до аналізу діагностичної інформації.

Сучасні методи класифікації медичних зображень ґрунтуються переважно на підходах машинного та глибокого навчання, зокрема із застосуванням згорткових нейронних мереж. Використання таких методів дозволяє автоматизувати процес аналізу МРТ-зображень і забезпечити більш високу точність діагностики завдяки здатності моделей виявляти складні нелінійні залежності у великих обсягах даних, які є важкодоступними для традиційних алгоритмічних підходів та візуальної оцінки.

Активне впровадження нейронних мереж у сферу медичної візуалізації є одним із провідних напрямів сучасних наукових досліджень, оскільки такі технології сприяють зменшенню впливу людського фактора, підвищенню об'єктивності результатів аналізу та покращенню якості клінічних рішень. У

цьому контексті дослідження автоматизованих методів виявлення та класифікації пухлин головного мозку за МРТ-зображеннями є важливим як з наукової, так і з практичної точки зору, оскільки їх застосування може сприяти ранньому виявленню патологій, підвищенню ефективності процесу лікування пацієнтів та рівня медичної допомоги.

Метою дослідження є розробка та аналіз моделей автоматизованого виявлення і класифікації пухлин головного мозку за магнітно-резонансними томографічними зображеннями на основі згорткових нейронних мереж, а також дослідження ефективності застосування методів глибокого та трансферного навчання для підвищення точності класифікації.

Для досягнення поставленої мети необхідно вирішити такі **основні завдання**:

1. Проаналізувати сучасні підходи до обробки медичних зображень і класифікації МРТ-знімків, включаючи методи глибокого навчання та згорткові нейронні мережі.
2. Дослідити архітектурні особливості та можливості застосування моделей VGG16 та ResNet50 для задачі класифікації пухлин головного мозку з використанням трансферного навчання.
3. Розробити методологію підготовки та обробки МРТ-зображень, включаючи нормалізацію, аугментацію даних та формування навчальної, валідаційної і тестової вибірок.
4. Реалізувати та адаптувати архітектури згорткових нейронних мереж VGG16 та ResNet50 для задачі багатокласової класифікації пухлин головного мозку (glioma, meningioma, pituitary, no tumor).
5. Провести навчання моделей із застосуванням методів регуляризації, оптимізації та адаптивного налаштування гіперпараметрів, а також виконати порівняльний аналіз їх ефективності.
6. Оцінити якість класифікації за метриками accuracy, precision, recall, F1-score та AUC, проаналізувати матриці плутанини та ROC-криві для кожного класу пухлин.

7. Розробити прототип програмного застосунку для практичної демонстрації можливостей автоматизованої класифікації МРТ-зображень пухлин головного мозку.

Об'єктом дослідження є процес автоматизованого виявлення та класифікації пухлин головного мозку на МРТ-зображеннях.

Предметом дослідження є застосування методів глибокого навчання та нейронних мереж для аналізу та класифікації МРТ-зображень головного мозку.

Наукова новизна даної роботи полягає у дослідженні та порівняльному аналізі сучасних методів виявлення та класифікації пухлин головного мозку на МРТ-зображеннях із застосуванням нейронних мереж, а також у визначенні найбільш ефективних методів та алгоритмів глибокого навчання, які забезпечують підвищену точність, стійкість та здатність до узагальнення при класифікації різних типів пухлин головного мозку.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Постановка задачі дослідження

Постановка задачі у даному дослідженні полягає в обґрунтуванні, розробленні та експериментальній перевірці методів автоматичного виявлення та класифікації пухлин головного мозку за МРТ-зображеннями з використанням сучасних архітектур глибокого навчання. Враховуючи складність морфологічних характеристик пухлин, значну індивідуальну варіативність та обмеження традиційної експертної інтерпретації, виникає потреба у створенні алгоритмічних рішень, здатних забезпечити вищу точність, об'єктивність та відтворюваність результатів діагностики.

Для досягнення поставленої мети передбачається дослідження можливостей згорткових нейронних мереж, зокрема моделей VGG16 та ResNet50, які за рахунок використання трансферного навчання можуть адаптувати попередньо здобуті знання до специфіки медичних зображень. В межах задачі визначено необхідність розробки повного програмного конвеєра обробки даних, що включає нормалізацію МРТ-зображень, застосування методів аугментації, екстракцію ознак та побудову спеціалізованого класифікатора для визначення чотирьох клінічно значущих категорій: glioma, meningioma, pituitary та відсутність пухлини.

Важливою складовою постановки задачі є проведення порівняльного аналізу продуктивності досліджуваних моделей, оцінка їх узагальнювальної здатності та визначення факторів, що впливають на точність прогнозування у кожному з класів. Це передбачає обчислення таких показників, як accuracy, precision, recall, F1-score, ROC-AUC, а також аналіз матриць плутанини з метою виявлення найскладніших для класифікації патологічних груп.

Окремим етапом роботи є розроблення практичної частини, спрямованої на демонстрацію можливості реального застосування створених моделей. З цією метою передбачено створення веб-застосунку, який дозволяє

завантажувати МРТ-зображення та автоматично отримувати прогноз щодо наявності та типу пухлини на основі попередньо навченої моделі. Такий інструмент забезпечує наочність результатів дослідження, підтверджує функціональність розробленого підходу та може бути використаний як основа для подальших клінічних або навчальних систем підтримки прийняття рішень.

1.2. Особливості МРТ-зображень та типи пухлин

Магнітно-резонансна томографія є провідним методом неінвазивної візуалізації м'яких тканин головного мозку, що базується на принципі ядерного магнітного резонансу. На відміну від комп'ютерної томографії, МРТ не використовує іонізуюче випромінювання, а натомість реєструє відгук атомів водню на вплив сильного магнітного поля та радіочастотних імпульсів. Завдяки високій концентрації води у мозковій тканині, МРТ забезпечує надзвичайно детальне відображення анатомічних структур та патологічних утворень, що робить її незамінною у діагностиці пухлин головного мозку [2].

МРТ-зображення формуються як тривимірні набори послідовних зрізів, товщина яких зазвичай становить від 1 до 5 міліметрів залежно від протоколу сканування. Кожен зріз являє собою двовимірну матрицю пікселів, інтенсивність яких відображає локальні магнітні властивості тканин. Просторова роздільна здатність сучасних МРТ-сканерів дозволяє виявляти структури розміром до 1 мм, що є критичним для ранньої діагностики малих новоутворень та точного визначення їх меж [2].

Важливою особливістю МРТ є можливість отримання зображень у різних площинах – аксіальній (поперечній), сагітальній (поздовжній) та корональній (фронтальній) – без зміни положення пацієнта. Це дозволяє комплексно оцінити локалізацію, форму та взаємовідношення пухлини з оточуючими анатомічними структурами. Крім того, застосування різних імпульсних послідовностей (T1-зважених, T2-зважених, FLAIR, DWI) надає

додаткову інформацію про характер тканин, наявність набряку, некрозу чи крововиливів всередині новоутворення.

Однак МРТ-зображення мають низку специфічних особливостей, що ускладнюють їх автоматичну обробку. Оскільки МРТ формується як набір пошарових зрізів, новоутворення може виглядати по-різному на сусідніх знімках або частково не потрапляти в окремий зріз через анатомічні особливості або орієнтацію площини сканування. На зображеннях також можуть спостерігатися варіації контрастності та яскравості, які залежать від типу МРТ-послідовності, параметрів сканування, характеристик обладнання та навіть індивідуальних фізіологічних особливостей пацієнта [2].

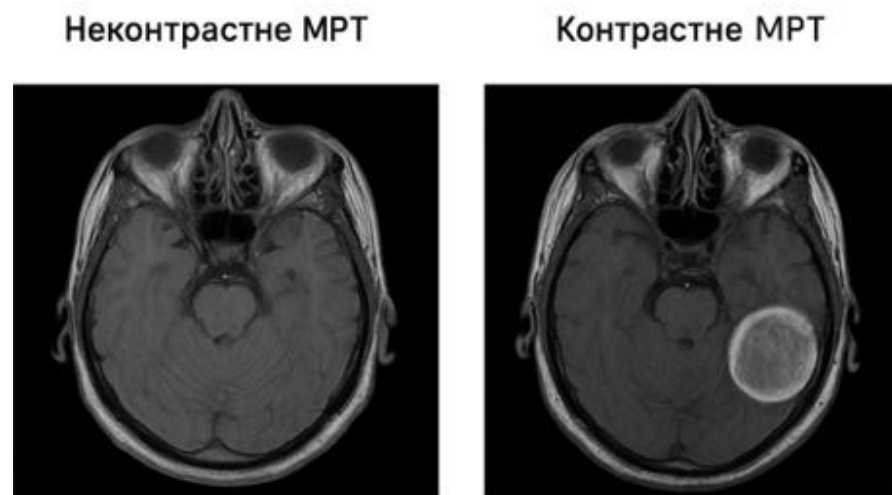


Рис. 1.1 — Приклад контрастного та неконтрастного МРТ знімків

Відмінності між контрастними та неконтрастними зображеннями необхідно враховувати під час розробки систем автоматичної класифікації, оскільки моделі штучного інтелекту можуть по-різному реагувати на структуру, контраст та інтенсивність зображення. У деяких випадках контрастування може суттєво змінити візуальні характеристики пухлини, виявивши неоднорідність її структури або приховані компоненти, які не помітні на неконтрастних знімках.

На МРТ-зображеннях виявляють різні типи пухлин головного мозку, кожна з яких має характерні ознаки локалізації, форми, структури та інтенсивності сигналу. Розуміння цих особливостей є критично важливим як

для радіологів, так і для розробки ефективних алгоритмів автоматичної класифікації.

Менінгіоми – це пухлини, що походять із клітин арахноїдальної оболонки головного мозку. Вони зазвичай розташовані біля поверхні мозку, часто прилягають до твердої мозкової оболонки та можуть спричиняти локальне здавлювання мозкової тканини. На МРТ-зображеннях менінгіоми характеризуються чіткими, добре визначеними контурами, часто мають округлу або овальну форму та демонструють інтенсивне однорідне накопичення контрастної речовини. Типовим є наявність "хвоста твердої мозкової оболонки" – лінійного підсилення сигналу вздовж прилеглої частини оболонки, що є майже патогномонічною ознакою менінгіом. Проте у випадках атипичних або анапластичних менінгіом можуть спостерігатися нечіткі контури, ознаки інвазії у мозкову тканину та неоднорідна структура [1].

Гліоми – це гетерогенна група первинних пухлин головного мозку, що походять із гліальних клітин. На відміну від менінгіом, гліоми розташовані всередині мозкової паренхіми та часто мають інфільтративний характер росту. На МРТ-зображеннях гліоми можуть виглядати менш виразно порівняно з іншими типами пухлин через схожість їх сигнальних характеристик із нормальною мозковою тканиною, особливо у випадку низькоградієнтних гліом. Високоградієнтні гліоми (гліобластоми) характеризуються нерівномірним накопиченням контрасту з формуванням кільцеподібної структури, наявністю центрального некрозу, значним перифокальним набряком та можливою геморагічною трансформацією. Межі гліом часто нечіткі через мікроскопічну інфільтрацію пухлинних клітин у оточуючу тканину, що ускладнює як візуальну, так і автоматизовану сегментацію [1].

Пухлини гіпофіза (аденоми) локалізуються в центральній частині мозку, у структурі турецького сідла – кісткової заглибини основи черепа. Ці новоутворення мають достатньо специфічну анатомічну локалізацію, що полегшує їх ідентифікацію. На МРТ пухлини гіпофіза можуть мати різну інтенсивність сигналу залежно від їх гістологічної структури та гормональної

активності. Макроаденоми (розміром понад 10 мм) часто демонструють супраселярне поширення з компресією зорового перехрещення, що є важливою діагностичною ознакою. Характер накопичення контрастної речовини може варіювати – від однорідного до гетерогенного, особливо у великих аденомах із внутрішньопухлинними крововиливами або кістозними змінами [1].

Ці морфологічні та локалізаційні особливості різних типів пухлин безпосередньо впливають на точність їх автоматичної класифікації та визначають складність задачі для систем на основі штучного інтелекту. Приклади МРТ-зображень із різними типами пухлин наведено на (рис. 1.2).

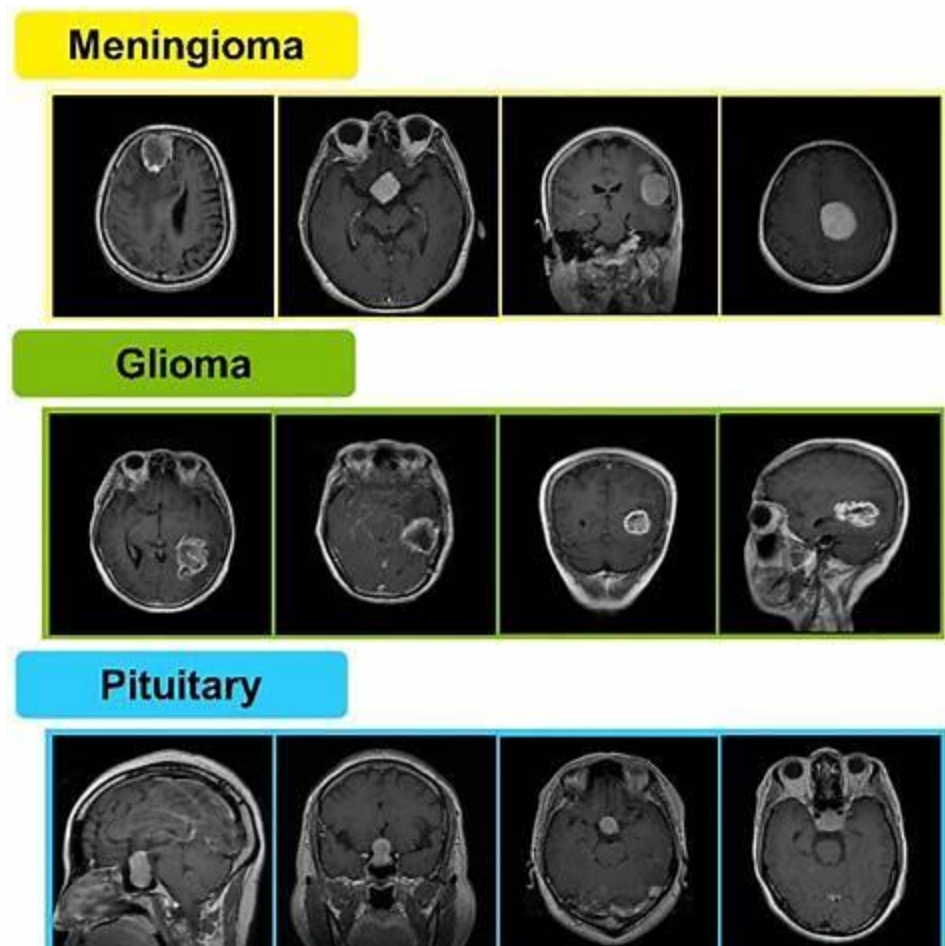


Рис. 1.2 — МРТ-зображення з різними типами пухлин: менінгіома, гліома та пухлина гіпофіза

Окрім варіативності, пов'язаної з біологічними характеристиками пухлин, МРТ-зображення можуть містити різноманітні артефакти, що виникають через рух пацієнта, магнітну неоднорідність, присутність

металевих імплантатів або технічні обмеження обладнання. Найпоширенішими є артефакти руху, артефакти хімічного зсуву, артефакти потоку та артефакти намотування. Такі спотворення можуть імітувати або маскувати патологічні зміни, що створює додаткові виклики для автоматизованих систем аналізу зображень.

1.3. Згорткові нейронні мережі та їх застосування у класифікації медичних зображень

У контексті класифікації навчання CNN відбувається шляхом оптимізації параметрів моделі на основі функції втрат, що відображає помилку між прогнозованими та реальними класами. Алгоритм зворотного поширення градієнта коригує ваги фільтрів та нейронів таким чином, щоб поступово покращувати здатність мережі ідентифікувати складні структури у зображеннях. Використання аугментації даних, регуляризації та оптимізаційних методів (наприклад, Adam) підвищує стабільність і точність класифікації [4].

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) є особливим різновидом глибоких моделей, запропонованих Яном ЛеКуном у 1989 році для ефективної обробки даних із просторовою структурою. Теоретичні основи таких мереж та їх застосування для розпізнавання образів детально описані у фундаментальних роботах з машинного навчання [9]. На відміну від класичних багатошарових персептронів, де кожен нейрон пов'язаний з усіма елементами вхідного вектора, згорткові мережі використовують механізм локальних рецептивних полів, що дозволяє моделі вибірково аналізувати невеликі фрагменти зображення та формувати ієрархічні ознаки. Такий підхід забезпечує інваріантність до зсувів, масштабування та деформацій, що є особливо важливим при роботі з медичними МРТ-знімками, де можливі зміни положення або розміру пухлини.

Архітектура згорткової нейронної мережі традиційно складається зі згорткових шарів, шарів субдискретизації та повнозв'язних рівнів, які здійснюють фінальну класифікацію [5]. Узагальнений вигляд архітектури CNN представлено на (рис. 1.3), де проілюстровано етапи перетворення вхідного МРТ-зображення у високорівневі ознаки.

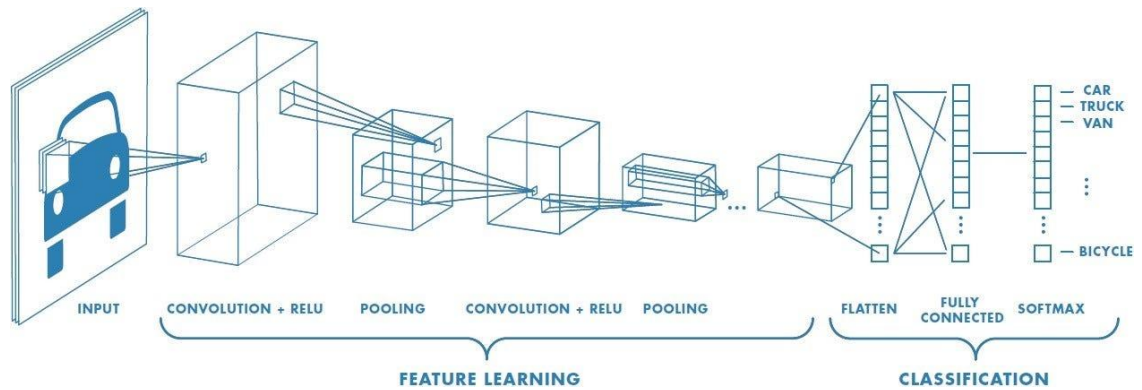


Рис. 1.3 — Архітектура CNN

Принцип роботи згорткового шару базується на застосуванні ядра, що ковзає по зображенню та обчислює локальний скалярний добуток, утворюючи карту активацій. Така операція дозволяє виділяти ключові структури, зокрема межі, текстурні особливості або характерні патерни пухлинної тканини. Загальний приклад виконання згортки наведено на (рис. 1.4).

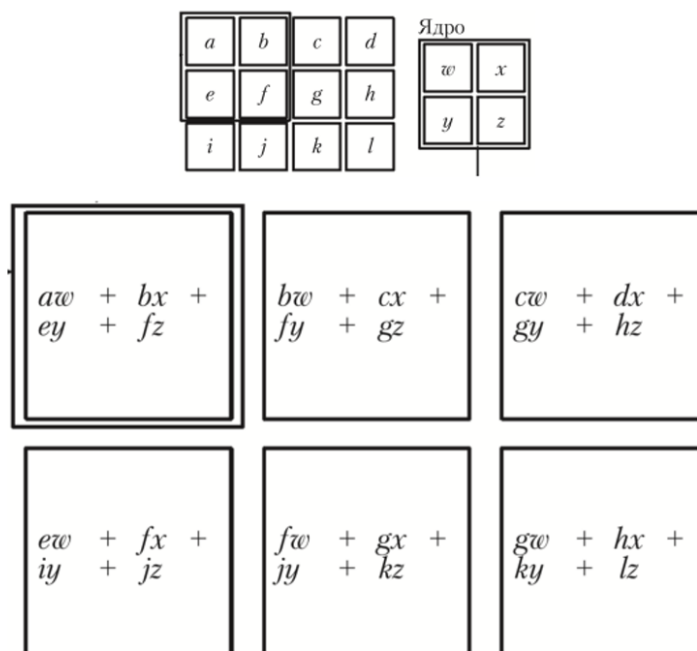


Рис. 1.4 — Приклад виконання операції згортки

Згорткові шари ґрунтуються на трьох фундаментальних концепціях: розрідженій взаємодії, спільному використанні параметрів та еквівалентності до трансляції. Використання компактних ядер дозволяє суттєво зменшити кількість параметрів у порівнянні зі звичайними повнозв'язними мережами, а спільне використання ваг забезпечує стійкість до переміщення об'єктів у межах зображення. Внаслідок цього CNN здатні ефективно виділяти локальні закономірності, що надзвичайно важливо під час аналізу МРТ головного мозку.

Після декількох згорткових шарів зазвичай використовується операція субдискретизації (pooling), яка зменшує просторові розміри карт ознак та робить модель менш чутливою до дрібних деформацій [6]. Формально вона визначається як вибір найбільшого значення з кожної локальної області карти активацій. Нехай $z_{i,j}^k$ — значення карти ознак на k -му шарі в позиції (i, j) . Тоді вихідне значення наступного шару $x_{i,j}^{k+1}$ обчислюється за формулою:

$$x_{i,j}^{k+1} = \max_{-d \leq a, b < d} z_{i+a, j+b}^k \quad (1.1)$$

де d визначає радіус локального вікна підвибіркування.

Таким чином, для кожної області розміром $2d \times 2d$ вибирається максимальне значення, яке найкраще відображає наявність значущої ознаки в даному фрагменті зображення. Застосування max pooling зменшує обчислювальну складність моделі, підвищує її стійкість до незначних зсувів і деформацій об'єктів та дозволяє зберігати найбільш інформативні ознаки, що є важливим при аналізі МРТ-знімків головного мозку. Приклад результату роботи шару pooling наведено на (рис. 1.5), де продемонстровано спосіб агрегування локальних фрагментів у компактніше представлення.

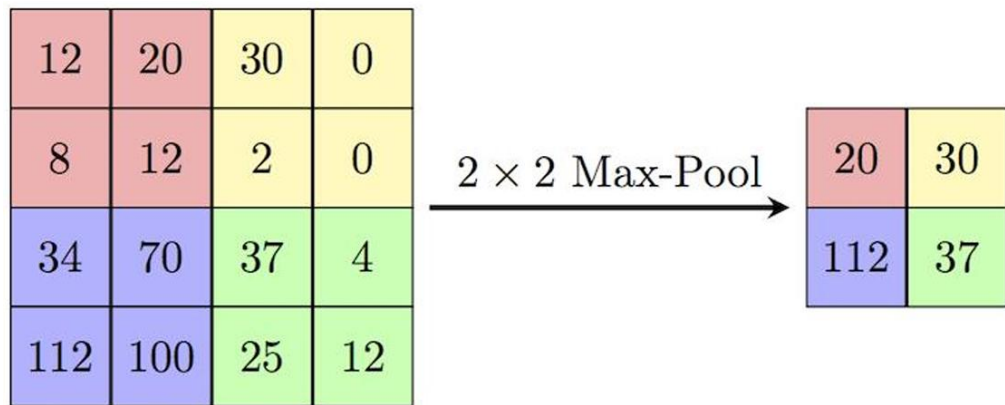


Рис. 1.5 — Приклад max pooling

Підсумковим етапом роботи CNN є повнозв'язний шар, що отримує сформовані карти ознак і виконує класифікацію. У задачі визначення типів пухлин головного мозку такий шар зазвичай використовує функцію активації softmax [5], щоб перетворити вихід моделі на ймовірнісну оцінку належності до певного класу. Поєднання згорткових, pooling- та повнозв'язних шарів забезпечує CNN високу точність, стійкість і здатність виявляти ключові патерни, притаманні пухлинним утворенням.

1.4. Аналіз обраних архітектур (ResNet50, VGG16)

У рамках даної роботи для класифікації МРТ-зображень було обрано дві базові архітектури: VGG16 та ResNet50. Ці моделі є одними з найпоширеніших у задачах комп'ютерного зору, добре задокументовані, підтримують трансферне навчання та демонструють високу ефективність у медичній візуалізації. Детальний аналіз їх внутрішньої структури наведено нижче.

Архітектура VGG-16 є класичною та широко відомою глибокою згортковою нейронною мережею, розробленою науковцями з Оксфордської групи комп'ютерного зору (Visual Geometry Group) у 2014 році [7]. Основною метою дослідження було системне вивчення впливу збільшення глибини мережі на точність розпізнавання зображень у масштабних задачах, таких як класифікація в наборі даних ImageNet.

Ключовою інновацією архітектури став принцип побудови глибокої мережі за допомогою каскадування малих згорткових фільтрів розміром 3×3 . Кожен такий фільтр з кроком (stride) 1 із застосуванням доповнення (padding) зберігає просторовий розмір карти ознак. Послідовне застосування декількох таких фільтрів із функцією активації ReLU між ними дозволяє імітувати рецептивне поле більшого фільтра (наприклад, 5×5 або 7×7), але з меншою кількістю параметрів та більшою нелінійністю, що підвищує представницьку потужність моделі [7].

Узагальнений вигляд архітектури VGG16 представлено на (рис. 1.6). Архітектура VGG16 має чітку модульну структуру та складається з п'яти послідовних згорткових блоків, за якими розташована класифікаційна гілка із трьох повнозв'язних (fully connected) шарів. Загальна кількість шарів з параметрами, що навчаються, становить 16 (13 згорткових + 3 повнозв'язних), що й дало назву моделі [7].

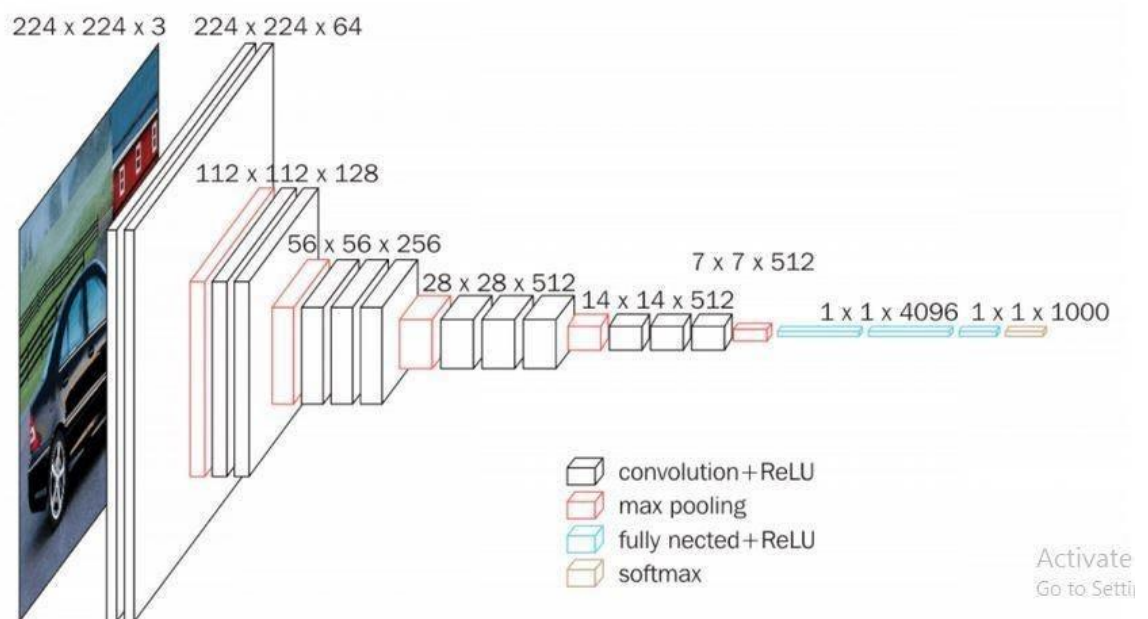


Рис. 1.6 — Архітектура VGG16

Кожен блок завершується шаром максимального пулінгу (MaxPooling) розміром 2×2 з кроком 2, що послідовно зменшує просторовий розмір карти ознак, подвоюючи при цьому їх семантичну абстрактність. Як показано на (рис. 1.7), кількість фільтрів зростає з кожним блоком за схемою $64 \rightarrow 128 \rightarrow$

256 → 512 → 512, що дозволяє мережі спочатку виявляти прості ознаки (краї, текстури), а на виході останніх блоків формувати складні абстрактні представлення об'єктів [8].

Після останнього пулінгу вектор ознак розгортається (flatten) та подається на класифікатор, що складається з двох прихованих повнозв'язних шарів по 4096 нейронів з регуляризацією Dropout та вихідного шару на 1000 нейронів (за кількістю класів ImageNet) з функцією активації softmax. Для задач трансферного навчання, як у даній роботі, цей вихідний шар зазвичай замінюється на новий, відповідно до кількості цільових класів [8].

Ця стандартна процедура налаштування (fine-tuning) передбачена та широко документована в основних фреймворках глибокого навчання, таких як Keras та PyTorch, що робить архітектуру VGG16 легко доступною та практичною для вирішення спеціалізованих завдань [8].

Архітектура ResNet50 належить до класу глибоких згорткових нейронних мереж і є однією з найуспішніших структур глибокого навчання, що дала змогу суттєво збільшити глибину моделей без погіршення їх здатності до оптимізації. В основі цієї архітектури лежить концепція резидуального навчання, запропонована Хе та співавторами у 2015 році [10]. На відміну від традиційних згорткових мереж, де кожен шар повинен безпосередньо апроксимувати цільове перетворення, ResNet50 використовує парадигму, в межах якої кожен блок навчається моделювати резидуальну функцію відносно вхідного сигналу. Такий підхід усуває проблему деградації продуктивності при збільшенні глибини моделі та забезпечує стабільне поширення градієнтів під час зворотного поширення помилки.

$$y = \mathcal{F}(x, W) + x, \quad (1.2)$$

де x — вхідний тензор блоку, $\mathcal{F}(x, W)$ описує послідовність згорткових перетворень із параметрами W ;

а y — його вихідний вектор активацій.

Наявність ідентичного відображення забезпечує можливість передавання інформації напряду між шарами, що суттєво спрощує процес оптимізації у глибоких мережах [10].

Архітектура ResNet50 складається з 50 навчуваних шарів і реалізована на основі bottleneck-блоків. Кожен блок містить три послідовні згорткові шари: перший із ядром 1×1 зменшує розмірність простору ознак, другий із ядром 3×3 відповідає за основне просторове перетворення, а третій із ядром 1×1 відновлює кількість каналів до початкової. Загальна структура мережі, представлена на (рис. 1.7), включає початковий великий згортковий шар, чотири групи резидуальних блоків та завершальний повнозв'язний шар. Пропускні (skip) з'єднання забезпечують ефективне навчання навіть при великій глибині моделі.

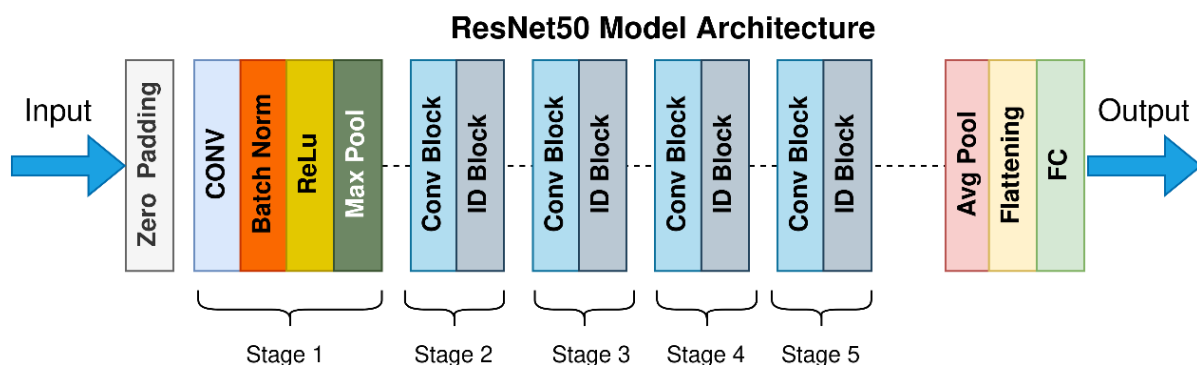


Рис. 1.7 — Архітектура ResNet50

Для забезпечення відтворюваності та коректного технічного опису реалізації ResNet50 важливо посилається також на офіційну документацію сучасних бібліотек глибокого навчання, які визначають конфігурацію шарів, формат вхідних даних і доступні параметри моделі. Ці деталі є стандартизованими та відкритими, що підтверджується документацією таких платформ, як Keras [8].

Проведений аналіз дозволяє зробити висновок, що обидві архітектури — VGG16 та ResNet50 — є надійними та добре вивченими моделями для задач класифікації зображень, однак вони реалізують різні підходи до побудови глибоких нейронних мереж. VGG16 характеризується простотою структури та

стабільною поведінкою, що робить її зручною для базових експериментів і задач з обмеженими обчислювальними ресурсами. Натомість ResNet50 завдяки резидуальним блокам демонструє значно кращу здатність до навчання на глибших рівнях і забезпечує вищу точність на складних наборах даних, що є важливою перевагою для медичних зображень, які містять дрібні та малопомітні патерни. Для узагальнення ключових характеристик нижче наведено порівняльну таблицю, яка демонструє основні технічні відмінності між двома обраними архітектурами та пояснює, чому доцільно використовувати їх у задачах класифікації МРТ-знімків мозку.

Таблиця 1.1

Порівняння архітектур VGG16 та ResNet50

Характеристика	VGG16	ResNet50
Кількість навчуваних шарів	16 (13 згорткових + 3 повнозв'язних)	50 (згорткові шари у складі bottleneck-блоків + fully connected)
Тип основного блоку	Стандартні згорткові шари 3×3	Резидуальні bottleneck-блоки ($1 \times 1 \rightarrow 3 \times 3 \rightarrow 1 \times 1$)
Ключова ідея	Збільшення глибини за рахунок каскаду малих фільтрів 3×3	Використання пропускних (skip) з'єднань для уникнення деградації глибоких мереж
Кількість параметрів	138 млн	25,6 млн
Глибина мережі	середня глибина (16 шарів)	велика глибина (50 шарів)
Розмірність фільтрів	3×3 у всіх згорткових шарах	1×1 та 3×3 у bottleneck-структурах
Обчислювальна ефективність	Нижча	Вища

Недоліки	Дуже велика кількість параметрів, високе споживання пам'яті та обчислювальних ресурсів	Складніша структура; важча у ручній модифікації
Тип класифікатора	Три fully connected шари (4096–4096–1000)	Один fully connected шар (1000) після global average pooling
Придатність для МРТ-зображень	Підходить для базових експериментів та невеликих вибірок	Краще виявляє складні патерни, придатна для високоточних медичних задач

1.5. Огляд існуючих програмних рішень для класифікації медичних зображень.

Сучасний етап розвитку медичної візуалізації характеризується активним упровадженням програмних систем, орієнтованих на автоматизований аналіз та інтерпретацію діагностичних зображень. Зростання обсягів медичних даних, ускладнення діагностичних протоколів і підвищені вимоги до точності та відтворюваності результатів обстеження зумовлюють необхідність використання інтелектуальних програмних рішень, здатних ефективно підтримувати лікаря на етапі прийняття клінічних рішень. У цьому контексті класифікація медичних зображень розглядається як один із ключових механізмів перетворення складної візуальної інформації на формалізовані діагностично значущі ознаки.

Особливої актуальності задачі класифікації набувають в межах багатокласових діагностичних сценаріїв, зокрема при розрізненні різних типів

пухлин головного мозку за МРТ-зображеннями. Автоматизоване віднесення зображень до клінічно значущих категорій дозволяє підвищити точність первинної діагностики, зменшити суб'єктивність інтерпретації результатів та створює основу для подальших етапів аналізу, включно з сегментацією пухлинного утворення, оцінкою його просторових і морфологічних характеристик, а також прогнозуванням перебігу захворювання. У цьому сенсі класифікація виступає фундаментальним компонентом комплексних систем підтримки прийняття рішень у медичній візуалізації.

Впровадження методів штучного інтелекту та глибокого навчання сприяло формуванню широкого спектру програмних рішень, призначених для автоматизованого аналізу та класифікації медичних зображень. Такі системи відрізняються за рівнем клінічної готовності, функціональними можливостями та доступністю для користувачів, що визначає доцільність їх застосування у клінічній або дослідницькій практиці. Найбільш зрілими з точки зору впровадження є комерційні системи комп'ютерної діагностики, які розробляються відповідно до медичних стандартів, проходять процедури регуляторної сертифікації та інтегруються безпосередньо у робочі процеси лікарів-радіологів.

Прикладом такого рішення є програмний комплекс BrainLab Elements, який забезпечує обробку мультимодальних МРТ-даних головного мозку, автоматичну сегментацію пухлинних утворень та кількісну оцінку їх об'ємних характеристик. Інтерфейс системи та результати автоматизованої сегментації пухлини головного мозку наведено на (рис. 1.8). Застосування подібних інструментів дозволяє візуалізувати патологічні зміни, аналізувати їх динаміку в процесі лікування та підвищувати об'єктивність діагностичних висновків.

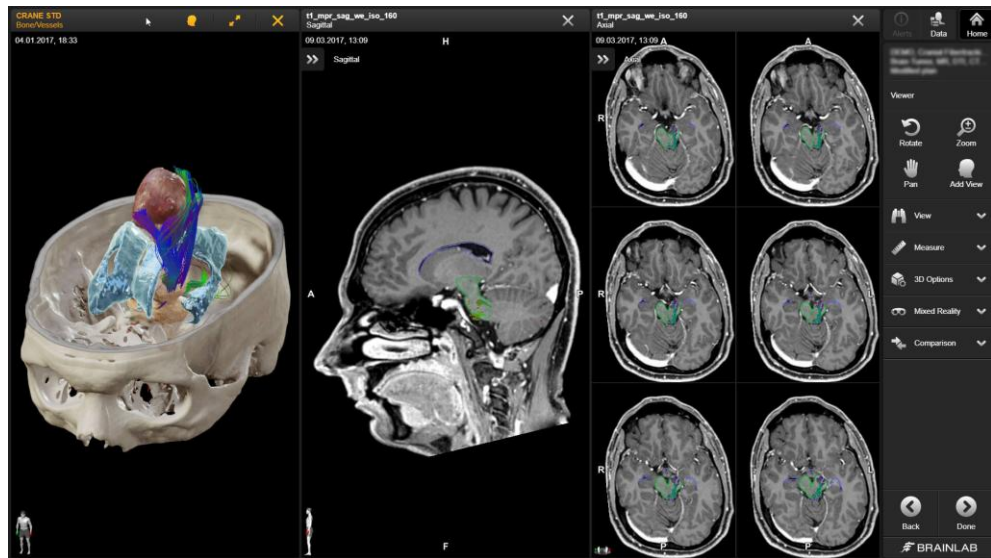


Рис. 1.8 — Інтерфейс BrainLab Elements з прикладом автоматичної сегментації пухлини головного мозку

Аналогічні принципи реалізовано у програмній платформі syngo.via компанії Siemens Healthineers, яка поєднує засоби постобробки КТ- та МРТ-зображень із модулями підтримки клінічних рішень. Приклад використання модуля нейровізуалізації для аналізу МРТ-зрізів головного мозку з автоматизованим виділенням анатомічних структур наведено на (рис. 1.9). Важливою особливістю цієї платформи є можливість розширення функціональності шляхом інтеграції сторонніх алгоритмів штучного інтелекту, що дозволяє адаптувати систему до нових діагностичних сценаріїв та актуальних наукових розробок.

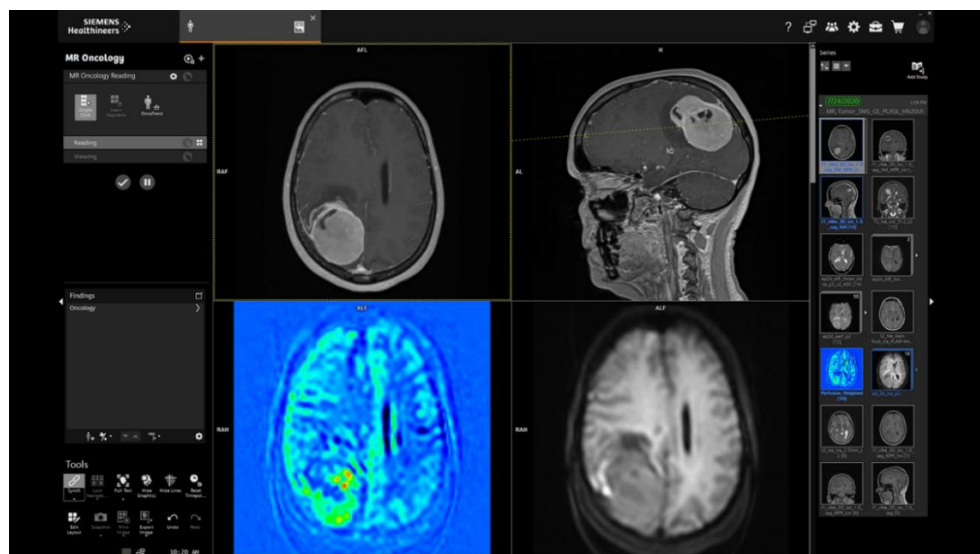


Рис. 1.9 — Аналіз МРТ головного мозку у середовищі Siemens

Поряд із комерційними рішеннями важливу роль у розвитку методів класифікації медичних зображень відіграють дослідницькі платформи з відкритим вихідним кодом. Однією з найбільш поширених є середовище 3D Slicer, яке широко використовується у наукових дослідженнях та експериментальній розробці алгоритмів обробки медичних зображень. На (рис. 1.10) наведено приклад автоматичної сегментації пухлини головного мозку з використанням нейронних мереж у середовищі 3D Slicer, що демонструє можливості інтеграції сучасних методів глибокого навчання у дослідницький робочий процес.

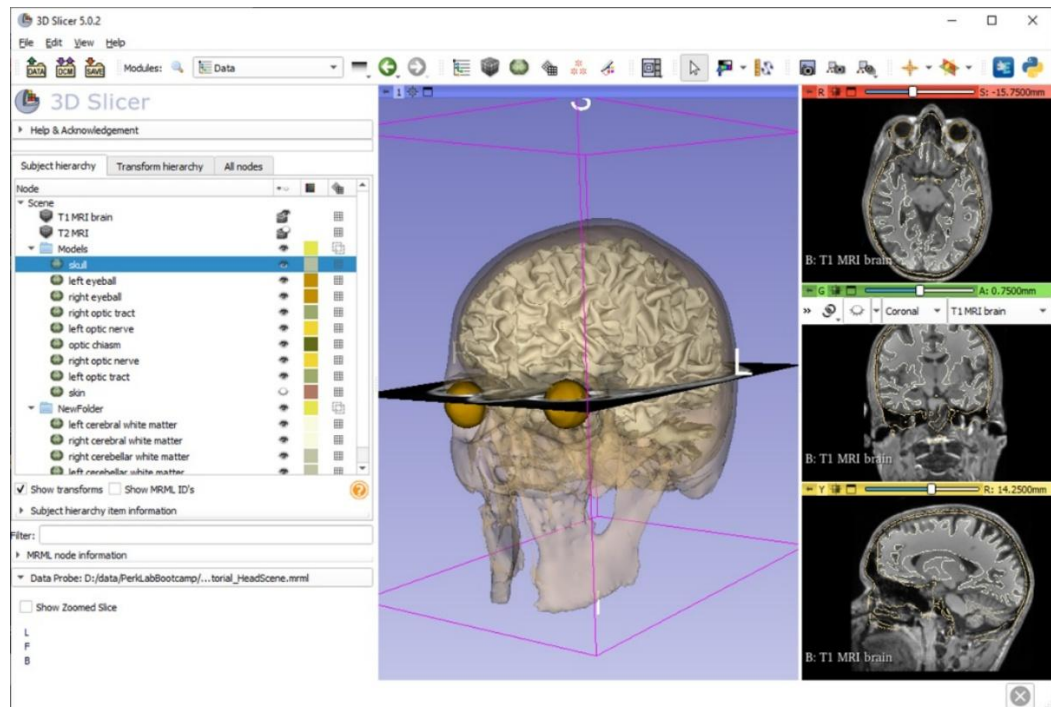


Рис. 1.10 — Сегментація пухлини головного мозку у середовищі 3D Slicer

Висновки до розділу 1

У першому розділі виконано комплексний теоретичний аналіз проблеми автоматизованого виявлення та класифікації пухлин головного мозку за МРТ-зображеннями. Показано, що специфіка магнітно-резонансної візуалізації, зокрема багат шаровий характер даних, варіативність інтенсивності сигналу, відмінності між контрастними та неконтрастними послідовностями, а також

складна морфологія пухлинних утворень, суттєво ускладнюють процес діагностики та зумовлюють потребу в застосуванні інтелектуальних методів аналізу зображень.

У ході розгляду основних типів пухлин головного мозку встановлено, що різна локалізація, форма та структура новоутворень безпосередньо впливають на складність їх автоматичного розпізнавання. Це підтверджує актуальність задачі багатокласової класифікації МРТ-зображень та необхідність використання моделей, здатних виявляти як глобальні, так і локальні просторові ознаки патологій.

Проаналізовано теоретичні основи згорткових нейронних мереж як ключового інструменту сучасної медичної візуалізації. Розглянуто принципи формування ієрархії ознак за допомогою згорткових, pooling- та повнозв'язних шарів, а також показано їхню здатність забезпечувати інваріантність до зсувів, масштабування та шуму, що є критично важливим при аналізі МРТ-знімків головного мозку. Обґрунтовано доцільність застосування методів аугментації даних, регуляризації та сучасних алгоритмів оптимізації для підвищення стабільності й узагальнювальної здатності моделей.

Окрему увагу приділено аналізу архітектур VGG16 та ResNet50, які були обрані як базові моделі для подальших експериментальних досліджень. Встановлено, що VGG16 вирізняється простою та структурованою архітектурою, що робить її зручною для базових експериментів і задач із відносно обмеженими обчислювальними ресурсами. Натомість ResNet50 завдяки використанню резидуальних з'єднань демонструє вищу ефективність навчання на глибших рівнях та кращу здатність до узагальнення, що є суттєвою перевагою при роботі зі складними медичними зображеннями.

Проведений огляд сучасних програмних рішень для класифікації медичних зображень показав, що інтелектуальні системи на основі глибокого навчання відіграють важливу роль у сучасній діагностичній практиці та наукових дослідженнях. Комерційні платформи комп'ютерної діагностики та дослідницькі інструменти з відкритим кодом підтверджують практичну

ефективність застосування алгоритмів штучного інтелекту для аналізу МРТ-зображень і водночас демонструють перспективність подальших розробок у цьому напрямі.

Таким чином, результати теоретичного аналізу першого розділу сформували цілісну методологічну основу для подальшої реалізації практичної частини дослідження. Обґрунтовано вибір архітектур глибинного навчання, визначено ключові вимоги до підготовки та обробки МРТ-даних, а також окреслено критерії оцінювання ефективності моделей. Це створює підґрунтя для розроблення, навчання та експериментальної перевірки системи автоматичної класифікації пухлин головного мозку, що є предметом наступного розділу роботи.

РОЗДІЛ 2. МЕТОДИ ТА МОДЕЛІ КЛАСИФІКАЦІЇ ПУХЛИН ГОЛОВНОГО МОЗКУ ЗА МРТ-ЗОБРАЖЕННЯМИ

2.1. Загальна методологія класифікації МРТ-зображень та структура дослідження

Більшість сучасних рішень ґрунтується на використанні глибинного навчання, зокрема згорткових нейронних мереж, які здатні автоматично виділяти ознаки та точно класифікувати складні медичні зображення. Розробка ефективної системи автоматичної класифікації пухлин головного мозку за МРТ-зображеннями потребує систематичного підходу, що охоплює всі етапи обробки даних — від первинної підготовки зображень до фінальної оцінки результатів, тому в дослідженні запропоновано комплексну методологію на основі глибоких згорткових нейронних мереж і трансферного навчання, реалізовану у вигляді послідовного конвеєра обробки.

Таблиця 2.1

Розподіл зображень датасету за класами

Клас	Навчальна вибірка	Тестова вибірка	Загалом
Гліома (glioma)	1321	300	1621
Менінгіома (meningioma)	1339	306	1645
Пухлина гіпофіза (pituitary)	1457	300	1757
Без пухлини (notumor)	1595	405	2000
Загалом	5712	1311	7023

Аналіз таблиці 2.1 показує, що розподіл за класами в навчальній вибірці є відносно збалансованим, що є важливою умовою для якісного навчання моделі без значного зміщення у бік окремих класів. Тестова вибірка, що

становить приблизно 18.7% від загального обсягу даних, дозволяє адекватно оцінити узагальнювальну здатність алгоритму на незалежних даних, які не брали участі у процесі навчання. Використання цього структурованого та публічного датасету забезпечує відтворюваність експериментів та можливість порівняння отриманих результатів з іншими дослідженнями в галузі медичної діагностики на основі штучного інтелекту.

Приклади МРТ-зображень з кожного класу датасету представлено на (рис. 2.1), де можна спостерігати характерні візуальні ознаки кожного типу пухлини та нормальних тканин головного мозку.

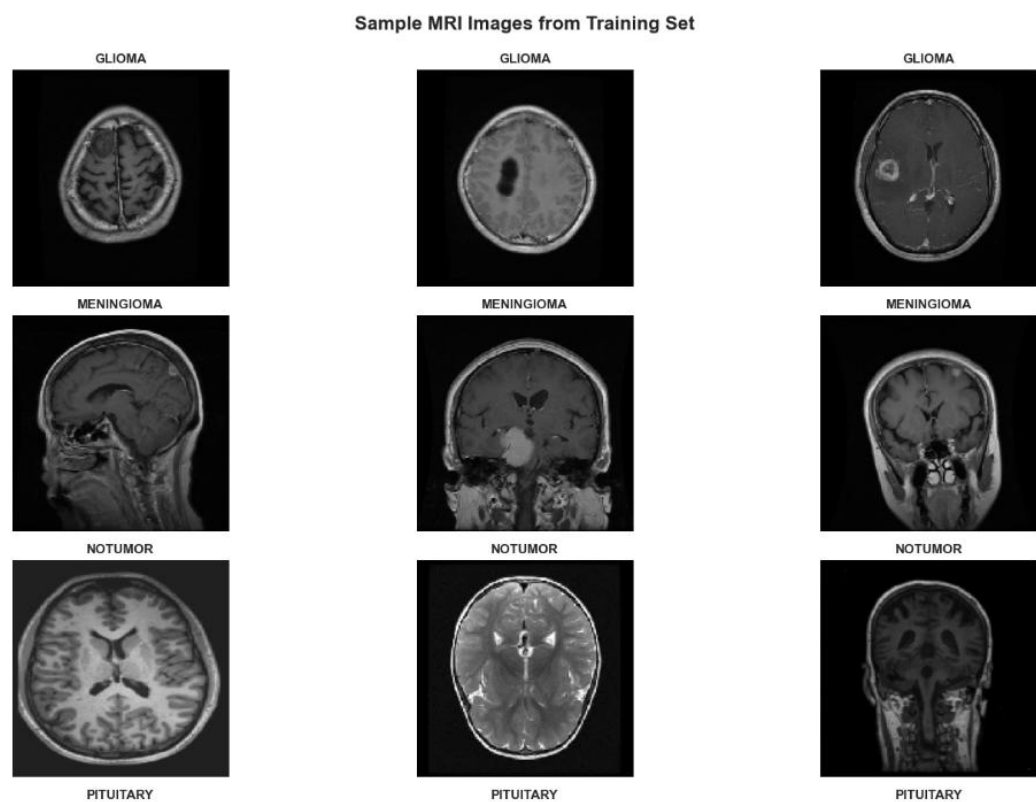


Рис. 2.1 — Приклад зображень з датасету Brain Tumor MRI.

Концептуальна схема процесу класифікації. Загальна методологія класифікації МРТ-зображень реалізована у вигляді семиетапного конвеєра обробки даних, схематично представленого на (рис. 2.2). Кожен етап виконує специфічні функції перетворення та аналізу інформації, забезпечуючи поступовий перехід від вхідних зображень до фінального рішення про наявність та тип пухлини.

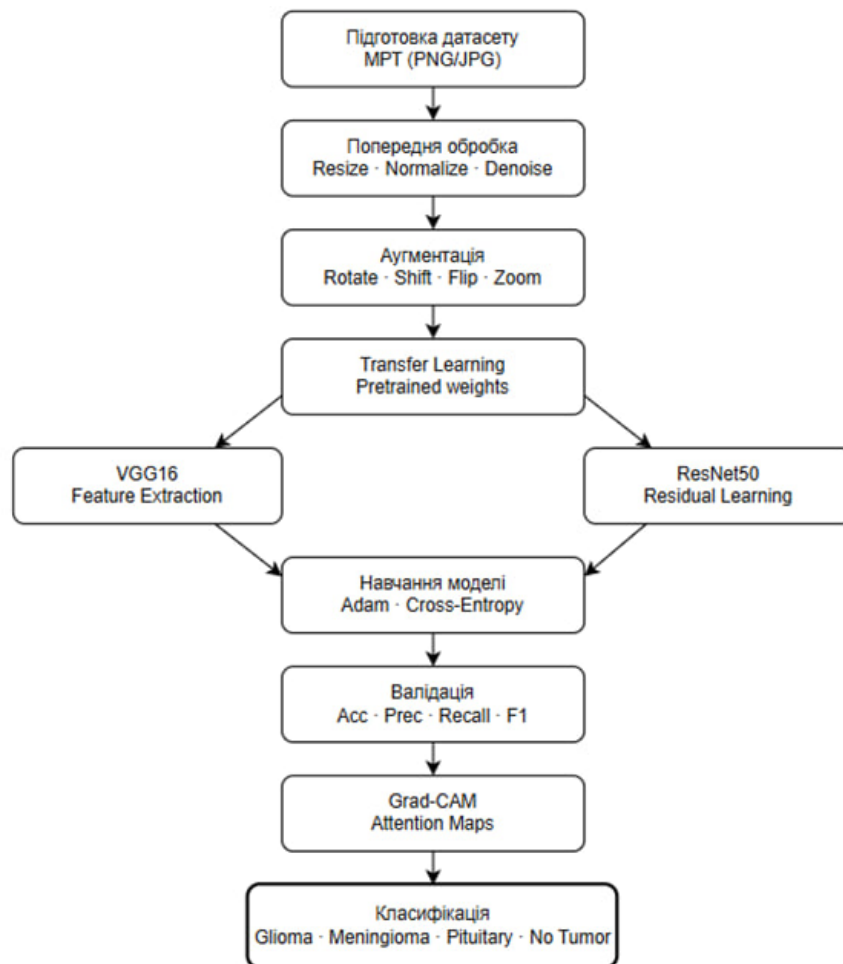


Рис. 2.2 — Концептуальна схема процесу класифікації МРТ-зображень

Перший етап — підготовка датасету — включає організацію файлової структури, перевірку цілісності даних та формування навчальної, валідаційної та тестової вибірок. На цьому етапі виконується аналіз розподілу класів, виявлення потенційних аномалій у даних та забезпечення коректного формату зображень для подальшої обробки. Навчальна вибірка додатково розділяється на тренувальну та валідаційну частини у співвідношенні 80:20, що дозволяє контролювати процес навчання та запобігати перенавчанню моделі.

Другий етап — попередня обробка МРТ-зображень — передбачає стандартизацію розмірів усіх зображень до єдиного формату 224×224 пікселі, що відповідає вхідним вимогам архітектур VGG16 та ResNet50. Крім зміни розміру, виконується нормалізація інтенсивності пікселів відповідно до статистичних параметрів датасету ImageNet, на якому були попередньо

навчені базові моделі. Математично нормалізація описується наступним чином:

$$x_{norm} = \frac{x - \mu}{\sigma} \quad (2.1)$$

де x — вектор значень пікселів вхідного зображення;

μ — вектор середніх значень каналів RGB, $\mu = [103.939, 116.779, 123.68]$;

σ — вектор стандартних відхилень каналів, що дорівнює одиниці для спрощення обчислень в рамках preprocessing функції ResNet50 та VGG16.

Така нормалізація забезпечує узгодженість між розподілом значень у медичних зображеннях та очікуваним розподілом для попередньо навчених моделей, що є критичною умовою ефективності трансферного навчання.

Третій етап підготовки даних присвячений аугментації навчальної вибірки, метою якої є штучне підвищення різноманітності вхідних даних та зменшення ризику перенавчання моделей. Аугментація реалізується шляхом застосування випадкових геометричних і фотометричних перетворень до вхідних МРТ-зображень, що дозволяє імітувати можливі варіації положення, масштабу та інтенсивності сигналу, які можуть виникати в реальних клінічних умовах. У базовому підході, що використовується під час навчання моделі VGG16, застосовуються стандартні методи аугментації, зокрема випадкові обертання, горизонтальні віддзеркалення, масштабування та незначні зсуви. Для моделі ResNet50 реалізовано розширену стратегію аугментації, яка додатково включає метод Міхур. Цей метод формує нові віртуальні навчальні приклади шляхом лінійної інтерполяції пар зображень та відповідних міток класів, що сприяє згладжуванню меж між класами та підвищенню стійкості моделі до шуму у вхідних даних.

Четвертий етап полягає у застосуванні трансферного навчання, що є ключовим підходом для ефективного навчання глибоких нейронних мереж на обмежених за обсягом медичних датасетах. В межах даного дослідження використано попередньо навчені на великомасштабному наборі даних

ImageNet моделі VGG16 та ResNet50, які виконують роль універсальних екстракторів ознак. Згорткові частини цих архітектур зберігають здатність виділяти базові візуальні патерни, такі як краї, текстури та просторові структури, що є релевантними не лише для природних зображень, але й для МРТ-знімків головного мозку. Оригінальні класифікаційні шари моделей було вилучено та замінено новими класифікаційними головами, спеціально спроектованими для задачі чотирикласової класифікації пухлин головного мозку. Такий підхід дозволив адаптувати загальні візуальні ознаки до специфіки медичних зображень без необхідності повного перенавчання всієї моделі з нуля.

П'ятий етап охоплює безпосередній процес навчання моделей, який реалізує оптимізацію параметрів нейронної мережі шляхом мінімізації функції втрат. Навчання здійснюється ітеративно протягом фіксованої кількості епох, при цьому на кожній ітерації модель обробляє мініпакет зображень, обчислює значення помилки класифікації та коригує вагові коефіцієнти за допомогою алгоритму зворотного поширення помилки. Для моделі VGG16 використано оптимізатор Adam із фіксованою початковою швидкістю навчання та механізмом адаптивного зменшення кроку навчання ReduceLROnPlateau, який знижує швидкість навчання у разі стабілізації значення функції втрат на валідаційній вибірці. Для моделі ResNet50 застосовано оптимізатор AdamW, що враховує вагову регуляризацию, у поєднанні з планувальником Cosine Annealing with Warm Restarts, який забезпечує циклічну зміну швидкості навчання. Вибір різних стратегій навчання зумовлений відмінностями в архітектурній складності моделей та їх чутливістю до налаштування гіперпараметрів.

Шостий етап спрямований на валідацію та тестування навчених моделей і забезпечує об'єктивну оцінку їх узагальнювальної здатності. Валідаційна вибірка використовується безпосередньо під час навчання для моніторингу динаміки якості моделі та прийняття рішень щодо доцільності подальшого навчання або його дострокового завершення за допомогою механізму ранньої

зупинки. Тестова вибірка застосовується виключно після завершення навчального процесу та використовується для фінальної оцінки якості класифікації. На цьому етапі обчислюється комплекс кількісних метрик, що включає точність класифікації, повноту, прецизійність, F1-міру та площу під ROC-кривою, а також будується матриця плутанини, яка дозволяє детально проаналізувати характер помилок для кожного з класів пухлин.

Обґрунтування порівняльного підходу. Центральною особливістю методології даного дослідження є паралельна реалізація двох стратегій класифікації, що ґрунтуються на різних архітектурах та методах навчання. Базовий підхід на основі VGG16 представляє класичну парадигму трансферного навчання з використанням стандартних технологій, добре задокументованих у літературі та широко застосовуваних у практичних задачах. Розширений підхід на основі ResNet50 інтегрує сучасні методи оптимізації та регуляризації, що з'явилися у останні роки як результат активних досліджень у галузі глибокого навчання. Таке порівняння дозволяє не лише об'єктивно оцінити ефективність кожного підходу, але й виявити специфічні методологічні рішення, що найбільше впливають на точність класифікації медичних зображень.

Ключові відмінності між двома підходами систематизовано за наступними компонентами: вибір оптимізатора (Adam проти AdamW), функція втрат (Categorical Cross-Entropy проти Focal Loss), методи аугментації (стандартні трансформації проти Mixup), механізми регуляризації (Dropout проти L2 kernel regularization), планувальники швидкості навчання (ReduceLROnPlateau проти Cosine Annealing), стратегії fine-tuning (повне заморожування проти прогресивного розморожування) та наявність інструментів інтерпретації результатів (відсутність проти Grad-CAM).

Критерії оцінки ефективності методів. Для забезпечення об'єктивності порівняльного аналізу визначено єдиний набір критеріїв оцінки, що застосовується до обох моделей. Первинним критерієм є загальна точність класифікації (accuracy) на тестовій вибірці, що відображає частку правильно

класифікованих зображень. Однак, враховуючи важливість збалансованої продуктивності у медичній діагностиці, де пропуск пухлини (false negative) є значно критичнішим за хибне спрацьовування (false positive), додатково аналізуються метрики precision, recall та F1-score для кожного класу окремо. Матриці плутанини надають детальну інформацію про характер помилок класифікації, дозволяючи ідентифікувати найскладніші для розрізнення пари класів та потенційні слабкі місця моделей.

Додатковим критерієм є стабільність навчання та швидкість конвергенції, що оцінюються на основі аналізу кривих навчання (learning curves), які відображають динаміку зміни функції втрат та метрик точності протягом епох навчання. Модель вважається стабільною, якщо валідаційні метрики не демонструють значних флуктуацій та монотонно покращуються до досягнення плато. Швидкість конвергенції визначається кількістю епох, необхідних для досягнення цільової точності, що має практичне значення з точки зору обчислювальних витрат та часу розробки.

Нарешті, для моделі ResNet50 додатковим критерієм є якість інтерпретації результатів за допомогою Grad-CAM. Візуалізації активаційних карт аналізуються з точки зору анатомічної релевантності — чи фокусується модель на регіонах мозку, характерних для відповідного типу пухлини, чи спирається на артефакти та нерелевантні ознаки. Цей якісний критерій є суттєвим для оцінки придатності моделі до реального клінічного застосування.

2.2. Формування та адаптація архітектур згорткових нейронних мереж

У даній роботі для розв’язання задачі багатокласової класифікації МРТ-зображень пухлин головного мозку застосовано метод контрольованого навчання згорткових нейронних мереж із використанням трансферного навчання. Обраний підхід реалізовано шляхом адаптації попередньо навчених

архітектур VGG16 та ResNet50 до специфіки медичних зображень і формування спеціалізованих класифікаційних голів для розпізнавання чотирьох класів: glioma, meningioma, pituitary та відсутність пухлини. Використання трансферного навчання дозволяє ефективно залучати знання, набуті моделями під час навчання на великих загальних датасетах, та адаптувати їх до задачі з обмеженою кількістю медичних даних. Реалізація методу базується на використанні згорткової частини попередньо навчених моделей як фіксованого екстрактора ознак. Вхідні МРТ-зображення розміром $224 \times 224 \times 3$ пікселі подаються на конволюційну основу моделей із замороженими вагами, у результаті чого формується тензор високорівневих просторових ознак, що відображає характерні структурні особливості зображень пухлин головного мозку. Для агрегування просторових ознак і зменшення розмірності вихідного тензора застосовано операцію Global Average Pooling, яка дозволяє перетворити багатовимірне представлення ознак у компактний вектор фіксованої довжини. Такий підхід суттєво зменшує кількість параметрів, що навчаються, та сприяє зниженню ризику перенавчання, що є критично важливим для медичних задач із невеликими вибірками даних. Остаточна класифікація здійснюється за допомогою спеціалізованих класифікаційних голів, адаптованих до кількості класів і специфіки МРТ-зображень. Для адаптації архітектури VGG16 до задачі класифікації пухлин головного мозку стандартну класифікаційну частину моделі було повністю замінено на спеціалізовану класифікаційну голову. Конволюційна основа VGG16 складається з 13 згорткових шарів та 5 операцій субдискретизації та містить приблизно 14,7 млн параметрів, які під час навчання були повністю заморожені. Після операції Global Average Pooling, що редукує тензор розміром $7 \times 7 \times 512$ до вектора з 512 елементів, у класифікаційній голові послідовно застосовано повнозв'язний шар із 256 нейронами та функцією активації ReLU, після якого використано шар Dropout з коефіцієнтом 0,5 для регуляризації. Далі застосовано шар Batch Normalization [13], що стабілізує процес навчання. Наступним етапом є повнозв'язний шар

із 128 нейронами та активацією ReLU, після якого знову використано Dropout з коефіцієнтом 0,3 та шар Batch Normalization. Завершальним елементом класифікаційної голови є повнозв'язний вихідний шар із чотирма нейронами та функцією активації softmax, який формує ймовірнісний розподіл між класами. Загальна кількість параметрів, що навчаються у класифікаційній голові VGG16, становить близько 397 тисяч, що відповідає лише 2,7% від загальної кількості параметрів моделі. Така конфігурація забезпечує ефективну адаптацію мережі до обмеженого медичного датасету та суттєво знижує ймовірність перенавчання. Архітектуру адаптованої класифікаційної голови VGG16 представлено на (рис. 2.3).

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 7, 7, 512)	14,714,688
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
batch_normalization (BatchNormalization)	(None, 512)	2,048
dense (Dense)	(None, 512)	262,656
dropout (Dropout)	(None, 512)	0
batch_normalization_1 (BatchNormalization)	(None, 512)	2,048
dense_1 (Dense)	(None, 256)	131,328
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 4)	1,028

Total params: 15,113,796 (57.65 MB)

Trainable params: 397,060 (1.51 MB)

Non-trainable params: 14,716,736 (56.14 MB)

Рис. 2.3 — Архітектура адаптованої класифікаційної голови VGG16

Для архітектури ResNet50 використано більш складну класифікаційну голову порівняно з VGG16, що обумовлено більшою глибиною мережі та багатшим представленням ознак. Конволюційна основа ResNet50 включає 50 згорткових шарів із механізмом залишкових зв'язків і містить загалом близько 24,8 млн параметрів, які також були заморожені під час навчання. Після операції Global Average Pooling, що редукує тензор розміром $7 \times 7 \times 2048$ до вектора з 2048 елементів, у класифікаційній голові послідовно застосовано

повнозв'язний шар із 512 нейронами та функцією активації ReLU, шар Batch Normalization і Dropout з коефіцієнтом 0,5. Далі використано повнозв'язний шар із 256 нейронами та активацією ReLU, після якого знову застосовано Batch Normalization і Dropout з коефіцієнтом 0,3. Наступним етапом є повнозв'язний шар із 128 нейронами та функцією активації ReLU, доповнений шаром Batch Normalization і Dropout з коефіцієнтом 0,2. Завершує класифікаційну голову вихідний повнозв'язний шар із чотирма нейронами та функцією активації softmax.

Поступове зменшення коефіцієнтів Dropout від 0,5 до 0,3 та 0,2 дозволяє застосовувати більш інтенсивну регуляризацию на ранніх етапах обробки ознак і водночас зберігати інформативність представлення на фінальних шарах мережі. Загальна кількість параметрів, що навчаються у класифікаційній голові ResNet50, становить приблизно 1,2 млн, що відповідає 4,91% від загальної кількості параметрів моделі. Архітектуру адаптованої класифікаційної голови ResNet50 представлено на (рис. 2.4).

Model: "ResNet50_BrainTumor"

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 7, 7, 2048)	23,587,712
global_avg_pooling (GlobalAveragePooling2D)	(None, 2048)	0
batch_norm_1 (BatchNormalization)	(None, 2048)	8,192
dense_512 (Dense)	(None, 512)	1,049,088
dropout_1 (Dropout)	(None, 512)	0
batch_norm_2 (BatchNormalization)	(None, 512)	2,048
dense_256 (Dense)	(None, 256)	131,328
dropout_2 (Dropout)	(None, 256)	0
batch_norm_3 (BatchNormalization)	(None, 256)	1,024
dense_128 (Dense)	(None, 128)	32,896
dropout_3 (Dropout)	(None, 128)	0
output (Dense)	(None, 4)	516

Total params: 24,812,804 (94.65 MB)

Trainable params: 1,219,460 (4.65 MB)

Non-trainable params: 23,593,344 (90.00 MB)

Model Parameters Summary:

- Total parameters: 24,812,804
- Trainable parameters: 1,219,460 (4.91%)
- Non-trainable parameters: 23,593,344 (95.09%)

Рис. 2.4 — Архітектура адаптованої класифікаційної голови ResNet50

Основні відмінності між реалізаціями VGG16 та ResNet50 полягають у глибині класифікаційної голови, розмірності вхідного вектора ознак та стратегії регуляризації. У моделі VGG16 використано два проміжні повнозв'язані шари з вхідною розмірністю 512, тоді як у ResNet50 застосовано три проміжні повнозв'язані шари з вхідною розмірністю 2048. Архітектура ResNet50 використовує більш складну класифікаційну голову для адаптації багатшого представлення ознак, сформованого глибшою мережею з механізмом залишкових зв'язків, що дозволяє підвищити якість класифікації у складній задачі аналізу МРТ-зображень.

2.3. Підготовка даних та алгоритм навчання моделей

Якість та узагальнювальна здатність моделей глибинного навчання значною мірою залежать від коректної підготовки вхідних даних і вибору алгоритму навчання. У даній роботі процес підготовки МРТ-зображень та навчання моделей було організовано у вигляді послідовного конвеєра, що включає попередню обробку зображень, аугментацію даних, формування навчальних і тестових вибірок, а також оптимізацію параметрів нейронних мереж.

Вхідні МРТ-зображення приводилися до уніфікованого формату з розмірністю $224 \times 224 \times 3$, що відповідає вимогам попередньо навчених архітектур VGG16 та ResNet50. Для забезпечення коректної роботи моделей одноканальні МРТ-знімки було приведено до трьохканального вигляду шляхом дублювання каналу інтенсивності. Датасет було розподілено на навчальну, валідаційну та тестову вибірки з загальною кількістю 5712 навчальних, 1141 валідаційних, 1311 тестових, співвідношення 70% / 15% / 15%.

Значення пікселів нормалізувалися шляхом масштабування до діапазону $[0, 1]$, що дозволяє стабілізувати процес навчання та пришвидшити збіжність алгоритму оптимізації. Така нормалізація зменшує вплив абсолютних значень

інтенсивності та забезпечує числову стабільність під час обчислення градієнтів.

Зважаючи на обмежений обсяг медичних даних та високу варіативність проявів пухлин, у роботі застосовано метод аугментації зображень, спрямований на штучне збільшення різноманітності навчальної вибірки. Аугментація виконувалася лише для тренувального набору даних та не застосовувалася до валідаційних і тестових зображень. У процесі навчання до МРТ-зображень випадковим чином застосовувалися наступні геометричні перетворення: обертання в діапазоні $\pm 20^\circ$, горизонтальні та вертикальні зсуви до 20% від розміру зображення, горизонтальне відзеркалення з ймовірністю 0.5, масштабування в діапазоні $\pm 20\%$ та зсув з коефіцієнтом до 0.15. Приклади застосування аугментації до МРТ-зображень наведені на (рис. 2.5), де продемонстровано варіативність трансформацій одного вихідного зображення.

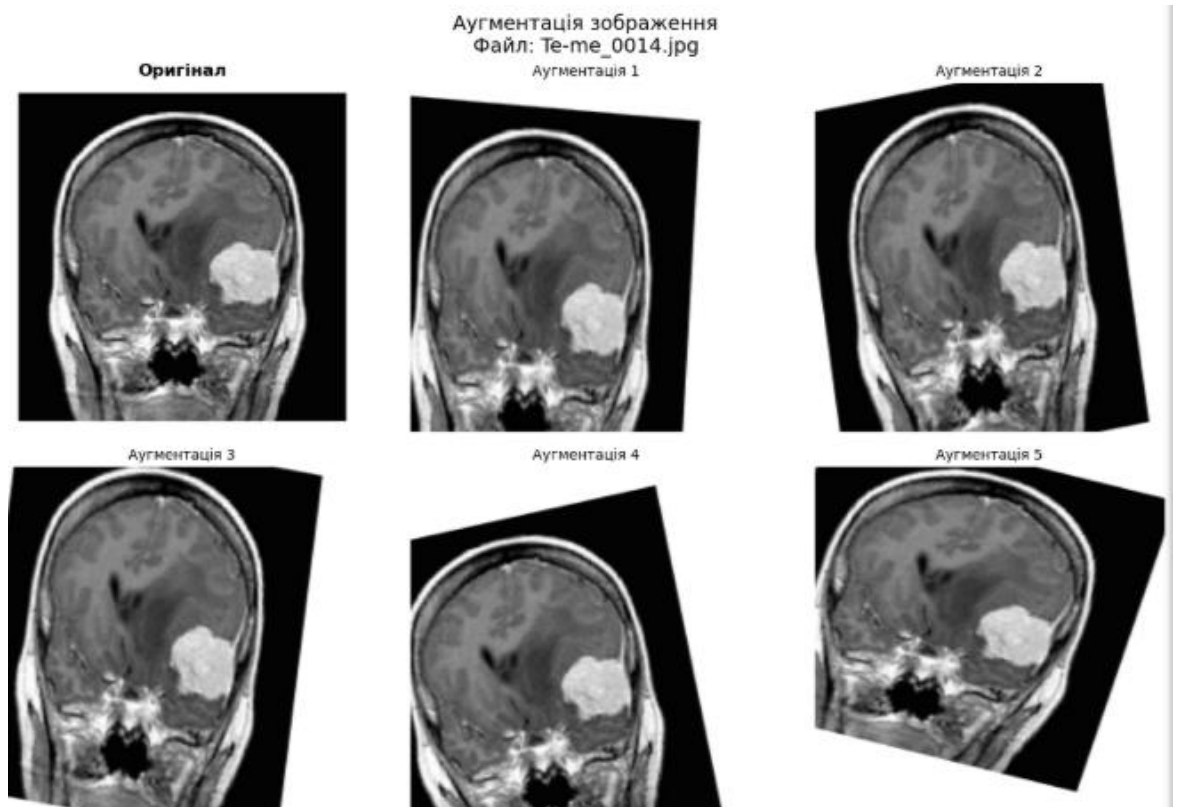


Рис. 2.5 — Приклади аугментації МРТ-зображень

Такі перетворення відтворюють можливі варіації положення голови пацієнта під час сканування та не спотворюють клінічно значущу структуру

зображень. Використання аугментації дозволяє зменшити перенавчання та підвищити здатність моделей до узагальнення на нових даних.

Навчання моделей здійснювалося шляхом оптимізації параметрів класифікаційних голів при зафіксованих вагах згорткових основ. Як функцію втрат для задачі багатокласової класифікації було використано categorical cross-entropy, яка визначається наступним чином:

$$L = - \sum_{i=1}^C y_i \log \hat{y}_i \quad (2.2)$$

де C — кількість класів ($C=4$);

y_i — істинна мітка класу;

$\hat{y}_i$ — ймовірність, передбачена моделлю для відповідного класу.

Оптимізація функції втрат виконувалася за допомогою алгоритму Adam з початковою швидкістю навчання 0.0001, який поєднує переваги методів стохастичного градієнтного спуску з адаптивним налаштуванням швидкості навчання для кожного параметра. Оновлення ваг у методі Adam відбувається відповідно до виразів:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad (2.3)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \quad (2.4)$$

$$\theta_t = \theta_{t-1} - \eta (\hat{m}_t) / (\sqrt{\hat{v}_t} + \epsilon), \quad (2.5)$$

де g_t — градієнт функції втрат на ітерації t ;

m_t — експоненційно згладжена оцінка першого моменту градієнта;

v_t — експоненційно згладжена оцінка другого моменту градієнта;

β_1 — коефіцієнт згладжування першого моменту ($\beta_1=0.9$);

β_2 — коефіцієнт згладжування другого моменту ($\beta_2=0.999$);

θ_t — вектор параметрів моделі на ітерації t ;

η — швидкість навчання ($\eta=0.0001$);

$\hat{m}_t$ — нормалізована оцінка першого моменту;

$\hat{v}_t$ — нормалізована оцінка другого моменту;

ϵ — константа для запобігання діленню на нуль ($\epsilon=10^{-8}$).

Навчання проводилося з розміром міні-батча 32 зображення протягом максимум 50 епох для кожної моделі. Для динамічної адаптації швидкості навчання застосовано callback ReduceLROnPlateau, який автоматично зменшує швидкість навчання вдвічі, якщо валідаційна втрата не покращується протягом 5 послідовних епох, при цьому мінімальна швидкість навчання обмежена значенням 10^{-7} . Для запобігання перенавчанню використано механізм EarlyStopping з терпимістю 10 епох, який зупиняє навчання у разі відсутності покращення валідаційної точності. Додатково застосовано ModelCheckpoint для збереження найкращої моделі за критерієм валідаційної точності.

Процес навчання моделі VGG16 тривав 46 епох до спрацювання механізму EarlyStopping. Найкраща валідаційна точність 84.57% була досягнута на 36-й епосі. Динаміка навчання VGG16 представлена на (рис. 2.6), де графіки точності та функції втрат демонструють стабільну конвергенцію моделі.



Рис. 2.6 — Динаміка навчання VGG16: точність класифікації та функція втрат

Аналіз (рис. 2.6) показує, що тренувальна точність зростає з 45.23% на першій епосі до 91.56% на фінальній епосі, тоді як валідаційна точність досягає максимуму 84.57% на 36-й епосі, після чого спостерігається незначне погіршення. Тренувальна втрата зменшується з 1.26 до 0.23, а валідаційна втрата стабілізується на рівні 0.49. Різниця між тренувальною та валідаційною

точністю становить 7%, що вказує на помірне перенавчання, успішно контрольоване завдяки комбінації аугментації даних, Dropout та BatchNormalization.

Додаткові метрики навчання VGG16 представлені на (рис. 2.7), де графіки precision та recall демонструють збалансоване зростання обох метрик.

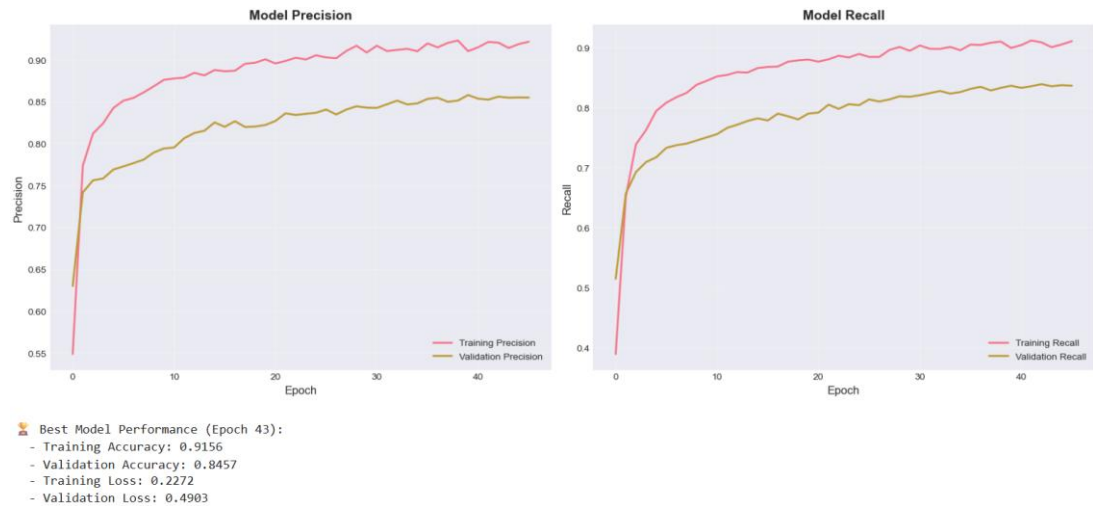


Рис. 2.7 — Динаміка метрик навчання VGG16: precision та recall

З (рис. 2.7) видно, що тренувальний precision досягає значення близько 0.92, тоді як валідаційний precision стабілізується на рівні 0.85. Тренувальний recall зростає до 0.91, а валідаційний recall — до 0.84. Паралельне зростання обох метрик свідчить про збалансований розвиток здатності моделі як правильно ідентифікувати позитивні випадки (recall), так і уникати хибнопозитивних класифікацій (precision). Найкращі показники моделі VGG16 на епосі 36 наведено в (табл. 2.2).

Таблиця 2.2

Найкращі показники навчання моделі VGG16

Метрика	Навчальна вибірка	Валідаційна вибірка
Accuracy	0.9156	0.8457
Loss	0.2314	0.4926
Precision	0.9234	0.8512
Recall	0.9087	0.8398
F1-Score	0.9160	0.8454

Аналіз (табл. 2.2) показує, що модель досягла високих показників на навчальній вибірці при збереженні прийнятної узагальнювальної здатності на валідаційних даних. Різниця між навчальними та валідаційними метриками становить 7-8%, що є прийнятним для медичних задач класифікації.

Процес навчання моделі ResNet50 тривав 50 епох. Найкраща валідаційна точність була досягнута на 47-й епосі. Загальний час навчання становив 2 години 34 хвилини 45 секунд, що приблизно вдвічі довше порівняно з VGG16 через значно більшу глибину мережі. Динаміка навчання ResNet50 представлена на (рис. 2.8), де графіки точності та функції втрат демонструють стабільну конвергенцію без різких коливань.



Рис. 2.8 — Динаміка навчання ResNet50: точність класифікації та функція втрат

Аналіз (рис. 2.8) показує, що тренувальна точність зростає від 53.12% на першій епосі до понад 92% на фінальних епохах, при цьому валідаційна точність демонструє паралельне зростання, досягаючи свого піку на 47-й епосі. Тренувальна втрата зменшується з 1.12 до приблизно 0.20, тоді як валідаційна втрата стабілізується на рівні близько 0.39. Різниця між тренувальною та валідаційною точністю становить приблизно 5%, що є кращим показником порівняно з VGG16 та свідчить про ефективність більш складної багатошарової регуляризації з прогресивним Dropout.

Додаткові метрики навчання ResNet50 представлені на (рис. 2.9), де графіки precision та recall демонструють стабільне зростання без різких коливань.

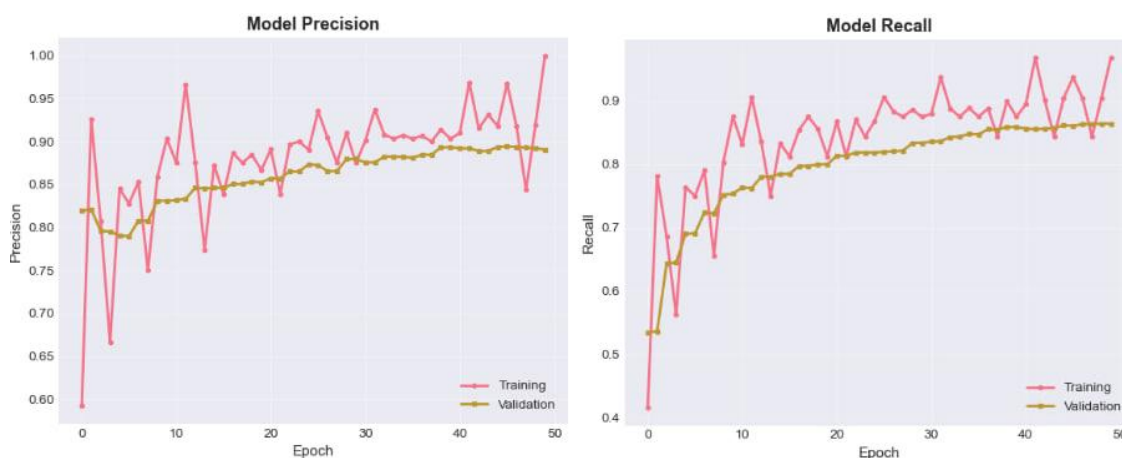


Рис. 2.9 — Динаміка метрик навчання ResNet50: precision та recall

Аналіз (рис. 2.9) показує, що тренувальний precision досягає значення близько 0.93, тоді як валідаційний precision стабілізується на рівні 0.88. Тренувальний recall зростає до 0.92, а валідаційний recall досягає 0.87. Різниця між тренувальними та валідаційними значеннями становить приблизно 5%, що підтверджує кращий контроль перенавчання порівняно з VGG16. Найкращі показники моделі ResNet50 на епосі 47 наведено в (табл. 2.3), яка демонструє детальні метрики навчання та валідації.

Таблиця 2.3

Найкращі показники навчання моделі ResNet50

Метрика	Навчальна вибірка	Валідаційна вибірка
Accuracy	0.9247	0.8789
Loss	0.2012	0.3894
Precision	0.9312	0.8823
Recall	0.9198	0.8734
F1-Score	0.9254	0.8778

Порівняльний аналіз результатів навчання двох архітектур наведено в (табл. 2.4). Дані таблиці дозволяють виявити як спільні риси навчального

процесу обох моделей, так і суттєві відмінності у їх продуктивності та обчислювальній ефективності.

Таблиця 2.4

Порівняння результатів навчання VGG16 та ResNet50

Параметр	VGG16	ResNet50
Кількість епох до зупинки	46	50
Епоха з найкращою точністю	36	47
Найкраща валідаційна точність	84.57%	87.89%
Різниця train-validation (%)	7.0%	5.0%
Час навчання	6:20:30	2:34:45
Кількість параметрів для навчання	397K	1.2M

Аналіз (табл. 2.4) показує, що модель ResNet50 демонструє кращі результати на валідаційній вибірці (87.89% проти 84.57%) та кращий контроль перенавчання (5% проти 7%), при цьому потребує значно менше часу для навчання (2:34:45 проти 6:20:30) завдяки ефективнішій архітектурі з residual connections, хоча має більшу кількість параметрів, що навчаються (1.2M проти 397K). Обидві моделі демонструють стабільну конвергенцію без ознак недонавчання або критичного перенавчання, що підтверджує коректність обраної стратегії регуляризації та аугментації даних.

2.4. Оцінювання ефективності та аналіз результатів класифікації

Оцінювання ефективності розроблених моделей класифікації є завершальним етапом експериментального дослідження та спрямоване на визначення здатності нейронних мереж коректно розпізнавати типи пухлин головного мозку на раніше невідомих даних. Аналіз проводився на незалежній тестовій вибірці, яка не використовувалася ані під час навчання, ані під час валідації моделей, що дозволяє об'єктивно оцінити узагальнювальну здатність архітектур VGG16 та ResNet50.

Для комплексної оцінки ефективності багатокласової класифікації використано набір стандартних метрик, які широко застосовуються у медичних задачах аналізу зображень. Основною метрикою є точність класифікації (accuracy), яка визначає частку правильно класифікованих зображень від загальної кількості тестових прикладів:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \quad (2.6)$$

де TP — кількість істинно позитивних класифікацій;

TN — кількість істинно негативних класифікацій;

FP — кількість хибнопозитивних класифікацій;

FN — кількість хибнонегативних класифікацій.

Для детальнішого аналізу якості розпізнавання кожного класу додатково використано метрики precision та recall. Precision характеризує частку правильних позитивних передбачень серед усіх позитивних передбачень моделі та обчислюється за формулою:

$$Precision = \frac{TP}{TP+FP}, \quad (2.7)$$

де TP — кількість істинно позитивних передбачень;

FP — кількість хибнопозитивних передбачень.

Recall (чутливість) відображає здатність моделі виявляти всі позитивні приклади відповідного класу:

$$Recall = \frac{TP}{TP+FN}, \quad (2.8)$$

де FN — кількість хибнонегативних передбачень.

Для узагальнення значень precision та recall використано F1-міру, яка є їх гармонічним середнім:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \quad (2.9)$$

Застосування F1-міри є особливо важливим у медичних задачах, де дисбаланс між класами може призводити до завищеної оцінки якості моделі лише за показником точності.

Оцінювання обох моделей проводилося на незалежній тестовій вибірці обсягом 1311 зображень, що становить приблизно 18,7% від загального датасету. Узагальнені результати тестування наведено в (табл. 2.5).

Таблиця 2.5

Результати оцінювання моделей на тестовій вибірці

Модель	Accuracy	Precision	Recall	F1-Score
VGG16	87.64%	88.37%	86.96%	87.66%
ResNet50	90.01%	91.08%	88.79%	89.92%

Аналіз (табл. 2.5) показує, що обидві моделі демонструють високу якість класифікації МРТ-зображень пухлин головного мозку. Модель ResNet50 забезпечує вищі значення всіх метрик порівняно з VGG16: перевага у точності становить 2,37 відсоткових пункти (90,01% проти 87,64%), у precision — 2,71 в.п. (91,08% проти 88,37%), у recall — 1,83 в.п. (88,79% проти 86,96%), та у F1-score — 2,26 в.п. (89,92% проти 87,66%). Ця перевага пояснюється більш глибокою архітектурою ResNet50 та здатністю формувати багатші ієрархічні представлення ознак завдяки механізму залишкових зв'язків.

Для оцінки здатності моделей розрізняти класи незалежно від порогу класифікації використано аналіз ROC-кривих (Receiver Operating Characteristic) та обчислення площі під кривою (AUC — Area Under Curve). ROC-крива відображає залежність між часткою істинно позитивних

класифікацій (True Positive Rate) та часткою хибнопозитивних класифікацій (False Positive Rate) при варіюванні порогу прийняття рішення. Метрика AUC приймає значення від 0 до 1, де значення 1 відповідає ідеальному класифікатору, а значення 0.5 — випадковому розподілу. ROC-криві для моделей VGG16 та ResNet50 представлено на (рис. 2.10) та (рис. 2.11) відповідно.

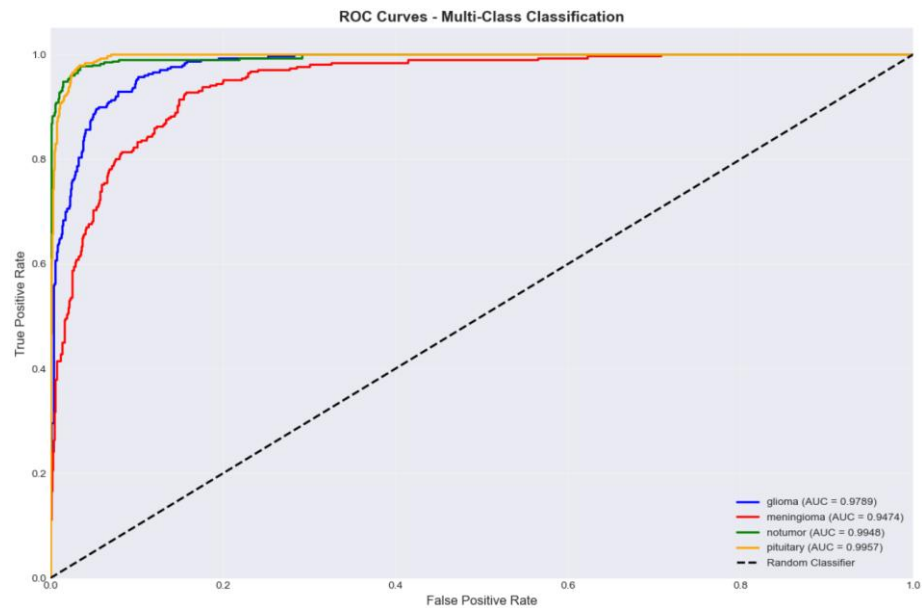


Рис. 2.10 — ROC-криві для багатокласової класифікації (модель VGG16)

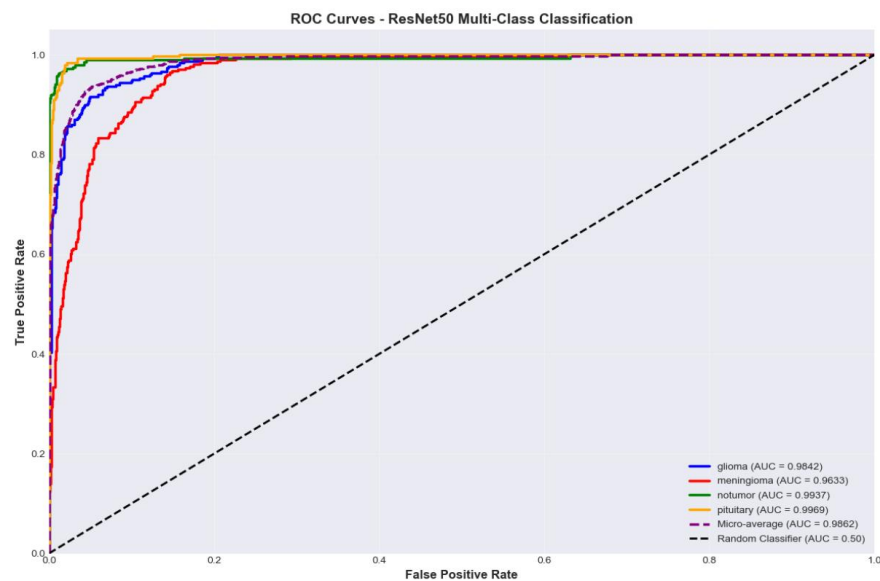


Рис. 2.11 — ROC-криві для багатокласової класифікації (модель ResNet50)

Для детального розуміння поведінки моделей на окремих типах пухлин проведено аналіз метрик для кожного класу. Результати для моделі VGG16 наведено в (табл. 2.6).

Таблиця 2.6

Детальні метрики класифікації моделі VGG16

Клас	Precision	Recall	F1-Score	Кількість зразків
Glioma	86.97%	82.33%	84.59%	300
Meningioma	79.93%	71.57%	75.52%	306
No Tumor	94.87%	95.80%	95.33%	405
Pituitary	85.76%	95.33%	91.61%	300

Таблиця 2.7

Детальні метрики класифікації моделі ResNet50

Клас	Precision	Recall	F1-Score	Кількість зразків
Glioma	87.38%	87.67%	87.52%	300
Meningioma	84.81%	74.84%	79.51%	306
No Tumor	94.70%	97.04%	95.85%	405
Pituitary	90.77%	98.33%	94.40%	300

Порівняльний аналіз (табл. 2.6) та (табл. 2.7) виявляє декілька важливих закономірностей. По-перше, обидві моделі найкраще розпізнають клас "no tumor" з F1-score понад 95%, що пояснюється чіткими відмінностями здорових тканин від патологічних утворень.

По-друге, найвищий recall спостерігається для класу "pituitary" (98,33% для обох моделей), що свідчить про характерну локалізацію та морфологічні особливості пухлин гіпофіза. По-третє, найнижчі показники зафіксовано для класу "meningioma" (F1-score 75,52% для VGG16 та 79,51% для ResNet50), що

вказує на складність диференціації менінгіом через їх морфологічну подібність до інших типів пухлин.

ResNet50 демонструє суттєве покращення розпізнавання класу "glioma" порівняно з VGG16: recall збільшився з 82,33% до 87,67% (на 5,34 в.п.), що є критично важливим з клінічної точки зору, оскільки пропуск гліоми може мати серйозні наслідки для пацієнта. Для класу "meningioma" ResNet50 також показує кращі результати з precision 84,81% проти 79,93% у VGG16, що свідчить про зменшення кількості хибнопозитивних класифікацій.

Для візуального аналізу помилок класифікації використано матрицю невідповідностей (confusion matrix), яка відображає розподіл передбачених та істинних класів.

Матриці невідповідностей для моделей VGG16 та ResNet50 наведено на (рис. 2.12) та (рис. 2.13) відповідно.

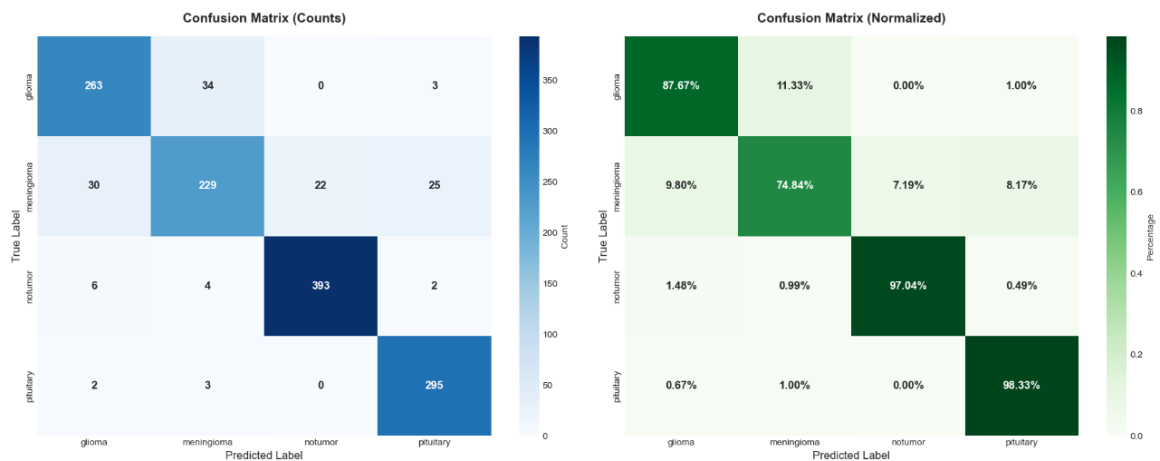


Рис. 2.12 — Матриці невідповідностей для моделі VGG16

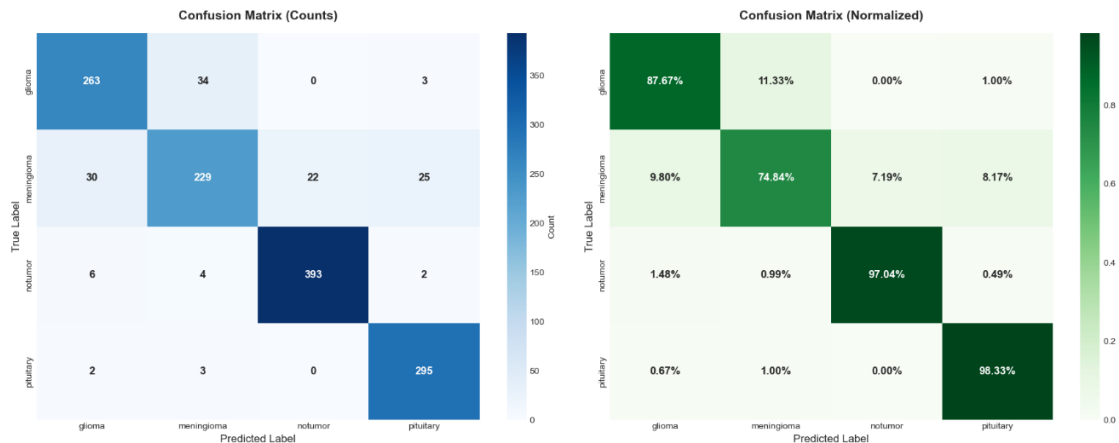


Рис. 2.13 — Матриці невідповідностей для моделі ResNet50

Аналіз матриць невідповідностей показує характерні особливості помилок класифікації для кожної моделі. Для моделі VGG16 (рис. 2.12) найбільша кількість помилок спостерігається при класифікації менінгіом: 87 випадків (28,43%) були неправильно класифіковані. Основна частина цих помилок припадає на плутанину з гліомою — 35 зображень менінгіом були помилково визначені як гліома, що становить 11,44% від загальної кількості менінгіом у тестовій вибірці. Також спостерігається 26 випадків помилкової класифікації менінгіом як зображень без пухлини та 26 випадків плутанини з пухлинами гіпофіза.

Для класу "glioma" модель VGG16 правильно класифікує 247 з 300 випадків (82,33%). Серед 53 помилок найбільш проблемною є плутанина з менінгіомою — 30 зображень гліом були неправильно визначені як менінгіоми. Це пояснюється морфологічною подібністю цих двох типів пухлин на МРТ-зображеннях, що ускладнює їх автоматичну диференціацію.

Модель ResNet50 (рис. 2.13) демонструє помітне покращення у розпізнаванні обох проблемних класів. Кількість помилково класифікованих менінгіом зменшується до 77 випадків (25,16% проти 28,43% у VGG16). Для гліом кількість помилок знижується з 53 до 37 випадків, що підтверджує кращу здатність ResNet50 виділяти дискримінативні ознаки для диференціації морфологічно подібних типів пухлин. Зокрема, кількість випадків плутанини між гліомою та менінгіомою зменшується з 30 (VGG16) до 22 (ResNet50).

Обидві моделі демонструють високу точність розпізнавання класу "no tumor": VGG16 правильно класифікує 388 з 405 випадків (95,80%), а ResNet50 — 393 з 405 (97,04%). Невелика кількість помилок для цього класу пояснюється чіткими візуальними відмінностями здорових тканин головного мозку від будь-яких патологічних утворень. Для ResNet50 лише 6 зображень без пухлини були помилково класифіковані як гліома, 4 — як менінгіома та 2 — як пухлина гіпофіза.

Клас "pituitary" демонструє найкращу чутливість серед усіх класів: обидві моделі правильно виявляють 295 з 300 випадків (98,33%). Це свідчить

про характерні та легко розпізнавані візуальні ознаки пухлин гіпофіза, пов'язані з їх специфічною локалізацією у селярній ділянці та типовою морфологією на МРТ-зображеннях. Однак спостерігається незначна кількість помилок: 2-3 випадки плутанини з іншими класами для кожної моделі.

Порівняльний аналіз матриць невідповідностей підтверджує, що основним джерелом помилок класифікації є плутанина між гліомою та мєнінгіомою, що є типовим для медичних задач через складність диференціації цих морфологічно подібних типів пухлин. ResNet50 краще справляється з цією проблемою завдяки більш глибокій архітектурі та механізму залишкових зв'язків, що дозволяє виділяти тонкі відмінності у текстах та структурах зображень.

Проведене оцінювання ефективності підтверджує доцільність використання трансферного навчання та спеціалізованих класифікаційних голів для задачі багатокласової класифікації МРТ-зображень пухлин головного мозку. Поєднання попередньої обробки даних, аугментації, регуляризації та адаптивної оптимізації параметрів дозволило досягти високої узагальнювальної здатності моделей на незалежній тестовій вибірці.

Модель ResNet50 демонструє кращу загальну продуктивність з точністю 90,01% порівняно з 87,64% у VGG16, що обумовлено декількома архітектурними та методологічними факторами. По-перше, більш глибока архітектура з 50 згортковими шарами дозволяє формувати багатші ієрархічні представлення ознак, здатні захоплювати тонкі текстурні та структурні відмінності між типами пухлин. По друге, механізм залишкових зв'язків (residual connections) полегшує навчання глибокої мережі та запобігає проблемі зникаючого градієнта, забезпечуючи стабільну конвергенцію навчального процесу. По-третє, більш складна класифікаційна голова з трьома проміжними повнозв'язаними шарами та прогресивною регуляризацією забезпечує кращу адаптацію високорівневих ознак до специфіки медичних зображень.

Важливо відзначити, що обидві моделі показують збалансовані значення precision та recall для більшості класів, що свідчить про відсутність значного зміщення у бік хибнопозитивних або хибнонегативних помилок. Це є критично важливим для медичних застосувань, де як пропуск патології (низький recall), так і хибне спрацьовування (низький precision) можуть мати серйозні клінічні наслідки для пацієнтів.

Аналіз помилок класифікації виявив, що найскладнішим класом для обох моделей є менінгіома з найнижчими значеннями F1-score (75,52% для VGG16 та 79,51% для ResNet50). Основні помилки пов'язані з плутаниною між морфологічно подібними типами пухлин, що є типовим для задач медичної візуальної діагностики та відображає об'єктивну складність диференціації певних патологічних утворень навіть для досвідчених фахівців-радіологів.

Отримані результати створюють надійну основу для практичного застосування розроблених моделей у системах комп'ютерної діагностики та демонструють перспективність використання глибинного навчання для задач автоматичного аналізу медичних зображень. Подальше вдосконалення системи може включати збільшення обсягу навчальних даних для проблемних класів, застосування більш складних методів аугментації, використання ансамблевих підходів або інтеграцію додаткових клінічних даних для покращення точності класифікації складних випадків.

Висновки до розділу 2

У другому розділі роботи було розроблено та детально обґрунтовано методи й моделі класифікації пухлин головного мозку за МРТ-зображеннями на основі згорткових нейронних мереж. Запропонована методологія охоплює повний цикл побудови системи автоматичної класифікації — від підготовки та аналізу датасету до навчання, валідації та комплексного оцінювання ефективності моделей.

У ході дослідження сформовано семиетапний конвеєр обробки даних, який забезпечує послідовну та узгоджену трансформацію МРТ-зображень у

фінальне класифікаційне рішення. Особливу увагу приділено підготовці даних, нормалізації та аугментації зображень, що дозволило зменшити вплив обмеженого обсягу медичного датасету та підвищити узагальнювальну здатність моделей.

Для розв'язання задачі багатокласової класифікації було адаптовано дві попередньо навчені архітектури — VGG16 та ResNet50 — із використанням трансферного навчання. Розроблені спеціалізовані класифікаційні голови дозволили ефективно адаптувати універсальні візуальні ознаки, сформовані на базі ImageNet, до специфіки медичних МРТ-зображень. Запропоновані архітектурні рішення та стратегії регуляризації забезпечили стабільне навчання моделей без критичного перенавчання.

Порівняльний аналіз результатів навчання та тестування показав, що обидві моделі демонструють високу якість класифікації на незалежній тестовій вибірці. Водночас модель ResNet50 забезпечує кращі показники точності, precision, recall та F1-score порівняно з VGG16, що пояснюється більш глибокою архітектурою та використанням залишкових з'єднань, які дозволяють формувати багатіші ієрархічні представлення ознак.

Детальний аналіз метрик за класами та матриць невідповідностей показав, що найбільш складним для автоматичної диференціації є клас менінгіоми, тоді як класи «відсутність пухлини» та «пухлина гіпофіза» розпізнаються з високою точністю. Виявлені помилки класифікації мають об'єктивний характер і пов'язані з морфологічною подібністю окремих типів пухлин на МРТ-зображеннях.

Загалом результати другого розділу підтверджують ефективність використання методів глибинного навчання та трансферного навчання для задач автоматичної класифікації пухлин головного мозку. Отримані моделі є достатньо точними, стабільними та придатними для подальшої практичної перевірки, що створює методологічну основу для реалізації та аналізу програмної системи класифікації, представленої у наступному розділі роботи.

РОЗДІЛ 3. ТЕСТУВАННЯ ТА ВАЛІДАЦІЯ МОДЕЛЕЙ КЛАСИФІКАЦІЇ ПУХЛИН ГОЛОВНОГО МОЗКУ

3.1. Візуалізація передбачень моделей VGG16 та ResNet50

Для детального аналізу роботи моделей класифікації пухлин головного мозку було застосовано метод візуалізації їхніх передбачень на тестових МРТ-зображеннях. Такий підхід дозволяє дослідити поведінку нейронних мереж на конкретних прикладах, оцінити характер прийнятих рішень та проаналізувати рівень упевненості моделей у різних клінічних ситуаціях. На відміну від агрегованих числових показників, візуалізація забезпечує більш глибоке розуміння того, як модель інтерпретує ознаки зображення та формує остаточне рішення. Для кожної з моделей було випадковим чином відібрано по 16 зображень із тестової вибірки, що дозволяє охопити різні класи пухлин та варіації МРТ-зрізів.

У процесі візуалізації для кожного МРТ-зображення відображалися істинна мітка класу, клас, передбачений моделлю, а також рівень упевненості у відсотках. Для підвищення наочності результати класифікації були доповнені кольоровим кодуванням: правильні передбачення позначалися зеленим кольором, тоді як помилкові — червоним. Такий спосіб подання інформації дозволяє швидко ідентифікувати коректні та проблемні випадки, а також спрощує подальший аналіз характеру помилок.

Результати візуалізації передбачень моделі VGG16 наведено на (рис. 3.1). Як видно з рисунка, модель правильно класифікувала 14 із 16 тестових МРТ-зображень, що відповідає точності 87,5% на даній підвибірці. Рівень упевненості у коректних передбаченнях змінювався у широкому діапазоні — від 50,3% до 100%, що свідчить про залежність упевненості моделі від складності конкретного зображення та виразності морфологічних ознак пухлини. У більшості випадків правильні рішення супроводжувалися високими значеннями ймовірності, що перевищували 90%. Особливо

впевнено модель розпізнавала зображення гліом, для яких у низці прикладів рівень упевненості досягав значень 99–100%.

Водночас на (рис. 3.1) зафіксовано дві помилкові класифікації, обидві з яких пов'язані з плутаниною між класами гліоми та менингіоми. В одному випадку зображення гліоми було віднесено до класу менингіоми з рівнем упевненості 56,5%, а в іншому — з упевненістю 65,0%. Характерною особливістю цих помилок є відносно низькі значення впевненості порівняно з коректними передбаченнями. Це може свідчити про наявність суперечливих ознак у вхідному зображенні та про невизначеність рішення, що в реальних умовах використання системи може потребувати додаткового аналізу з боку спеціаліста.

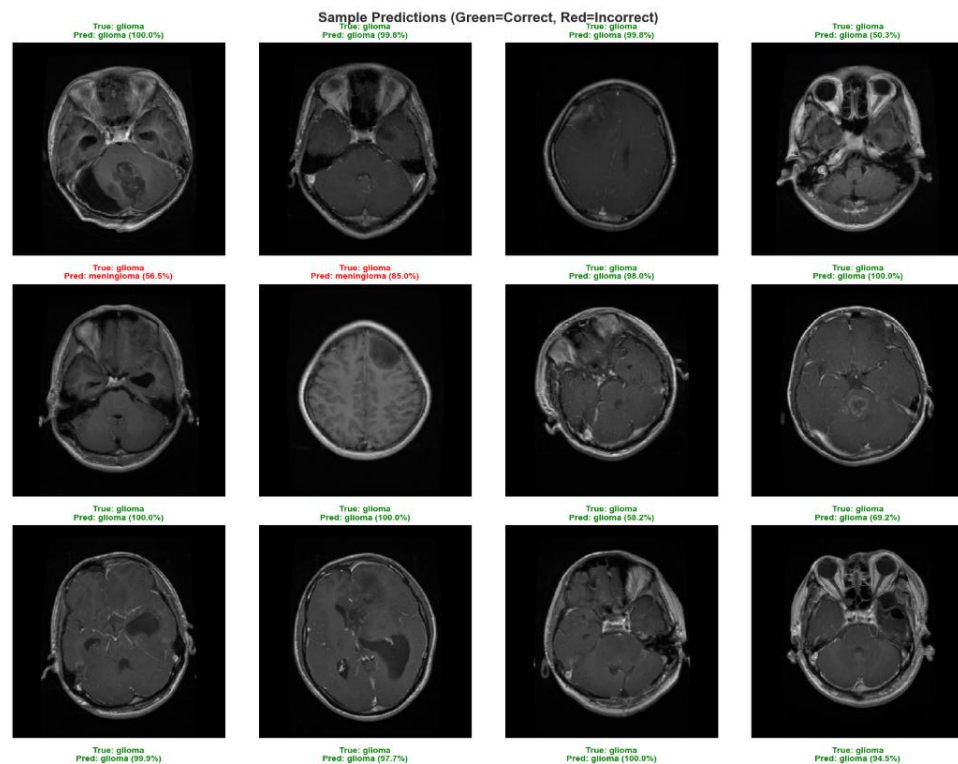


Рис. 3.1 — Візуалізація передбачень моделі VGG16 на 16 тестових МРТ-зображеннях

Візуалізація передбачень моделі ResNet50 представлена на (рис. 3.2). Аналіз наведеного рисунка показує, що модель коректно класифікувала всі 16 тестових МРТ-зображень. Для більшості прикладів рівень упевненості перевищував 90%, а в окремих випадках досягав 100%, що вказує на стабільність формування передбачень. Коректна ідентифікація спостерігалася

незалежно від варіацій анатомічних зрізів, локалізації пухлинних утворень та відмінностей у контрастності МРТ-зображень.

Особливістю результатів, наведених на (рис. 3.2), є відсутність випадків із вираженою невизначеністю рішення. Навіть у ситуаціях, де пухлина має нечіткі контури або складну внутрішню структуру, значення ймовірностей залишалися стабільно високими. Це свідчить про здатність моделі ResNet50 ефективно виділяти інформативні ознаки та узагальнювати отримані знання на різні типи зображень без істотного зниження впевненості.

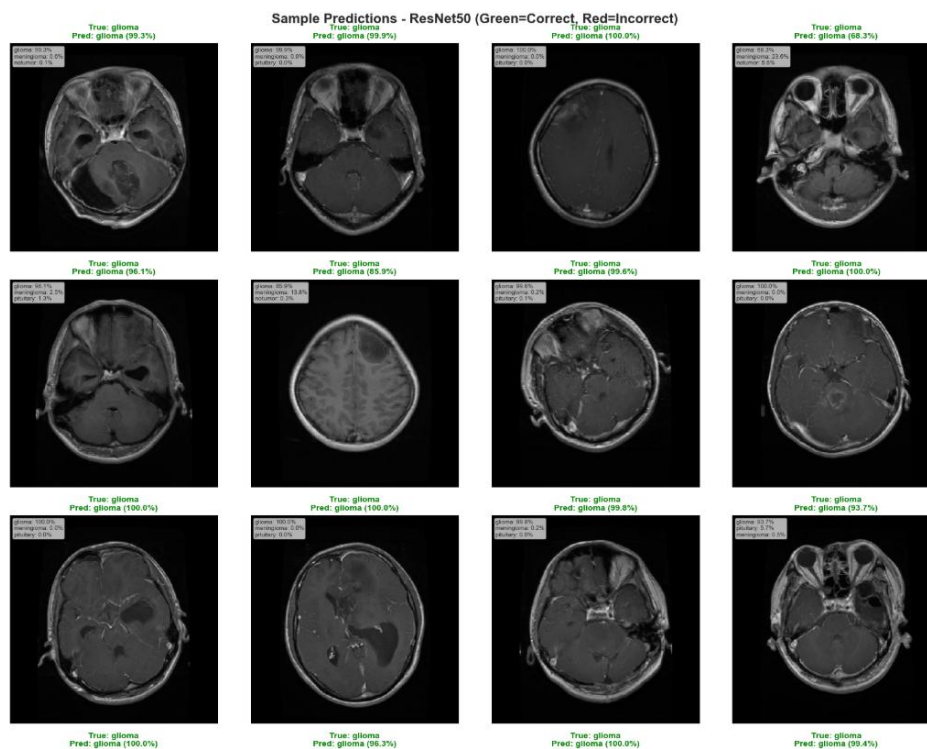


Рис. 3.2 — Візуалізація передбачень моделі ResNet50 на 16 тестових МРТ-зображеннях

Загалом, наведені результати візуалізації дозволяють детально простежити особливості роботи кожної з моделей на тестових МРТ-зображеннях. Такий формат представлення результатів забезпечує наочність аналізу, спрощує виявлення типових помилок класифікації та дозволяє оцінити рівень упевненості моделей у різних клінічних сценаріях, що є важливим для подальшого дослідження та вдосконалення систем автоматизованої діагностики.

Використання візуалізації передбачень у межах даного дослідження дозволяє не лише продемонструвати результати роботи моделей, але й створює основу для подальшого аналізу їхньої поведінки. Такий підхід є корисним інструментом для дослідження помилок класифікації, оцінювання рівня упевненості моделей та аналізу складних випадків, що має важливе значення при розробці та тестуванні систем комп'ютерної підтримки медичної діагностики.

3.2. Тестування моделей у прикладному програмному застосунку

З метою перевірки працездатності та практичної придатності розроблених моделей класифікації пухлин головного мозку було реалізовано веб-застосунок для автоматизованої діагностики МРТ-зображень із використанням методів глибокого навчання. Створення такого застосунку дозволило наблизити процес оцінювання якості моделей до реальних умов їх використання у клінічній практиці, а також забезпечити зручний і зрозумілий інтерфейс для взаємодії з нейронними мережами без необхідності володіння навичками програмування.

Веб-застосунок розроблено з використанням фреймворку Streamlit, який орієнтований на швидке створення інтерактивних інтерфейсів для систем машинного навчання. Архітектура застосунку має модульну організацію та об'єднує інтерфейс користувача, модуль попередньої обробки зображень і модуль класифікації, що базується на попередньо навчених нейронних мережах. Такий підхід забезпечує гнучкість системи, спрощує її масштабування та дозволяє легко інтегрувати нові моделі у майбутньому.

Інтерфейс користувача побудовано за принципом інтуїтивної навігації та розділено на дві функціональні області. У лівій панелі розміщено елементи для завантаження МРТ-зображень, вибору архітектури нейронної мережі (VGG16 або ResNet50) та налаштування параметрів відображення результатів. Права панель призначена для виведення результатів класифікації та містить

прогнозований діагноз, рівень упевненості моделі та детальний розподіл ймовірностей між усіма класами. Застосунок підтримує роботу з файлами форматів PNG, JPG та JPEG із максимальним розміром до 200 МБ, що дозволяє використовувати зображення без попередньої ручної підготовки.

Попередня обробка зображень виконується автоматично у відповідному модулі та відповідає вимогам обраної архітектури нейронної мережі. Після завантаження МРТ-знімок масштабується до стандартного розміру 224×224 пікселів, перетворюється у трьохканальний формат RGB та проходить процедуру нормалізації значень пікселів. Для моделі VGG16 використовується віднімання середніх значень каналів RGB, характерних для датасету ImageNet, тоді як для ResNet50 застосовується стандартна функція попередньої обробки бібліотеки Keras. Це забезпечує відповідність умов тестування умовам, у яких відбувалося навчання моделей, і запобігає спотворенню результатів класифікації.

Модуль класифікації відповідає за завантаження попередньо навчених моделей із локального сховища та формування передбачень на основі підготовлених зображень. Для оптимізації продуктивності використано механізм кешування Streamlit, що дозволяє уникнути повторного завантаження моделей при кожному запиті користувача. Результат класифікації формується у вигляді вектора ймовірностей, що відповідають чотирьом класам пухлин: гліома, менінгіома, пухлина гіпофіза та відсутність патології. Клас із найбільшою ймовірністю автоматично визначається як основний діагноз і відображається разом із відсотковим рівнем упевненості моделі.

Особливу увагу приділено візуалізації результатів класифікації. Прогнозований клас виділяється кольоровим акцентом, що дозволяє швидко ідентифікувати результат аналізу. Для кожного можливого діагнозу відображається окрема інформаційна картка з українською назвою класу, числовим значенням ймовірності та горизонтальним прогрес-баром. Така форма подання інформації сприяє наочній оцінці розподілу ймовірностей і

дозволяє користувачу проаналізувати не лише фінальне рішення системи, а й альтернативні варіанти.

Окрім результатів класифікації, застосунок надає можливість порівняння характеристик використаних моделей. У боковій панелі відображаються ключові показники продуктивності VGG16 та ResNet50, зокрема загальна точність на тестовому наборі, тривалість навчання та кількість параметрів. Ці дані автоматично оновлюються при зміні обраної архітектури, що дозволяє оцінити компроміс між точністю моделі та її обчислювальною складністю.

Важливим елементом інтерфейсу є постійно відображуване медичне застереження, яке підкреслює допоміжний характер системи. Користувач інформується про те, що результати класифікації не є остаточним медичним діагнозом і не можуть замінити консультацію кваліфікованого лікаря. Остаточне рішення має прийматися на основі комплексного аналізу клінічних даних та результатів додаткових досліджень.

Для перевірки роботи застосунку в умовах, наближених до реального використання, було сформовано два репрезентативних тест-кейси, які демонструють як коректну роботу системи, так і її типові помилки. Приклад успішної класифікації менінгіоми з використанням моделі ResNet50 наведено на (рис. 3.3).

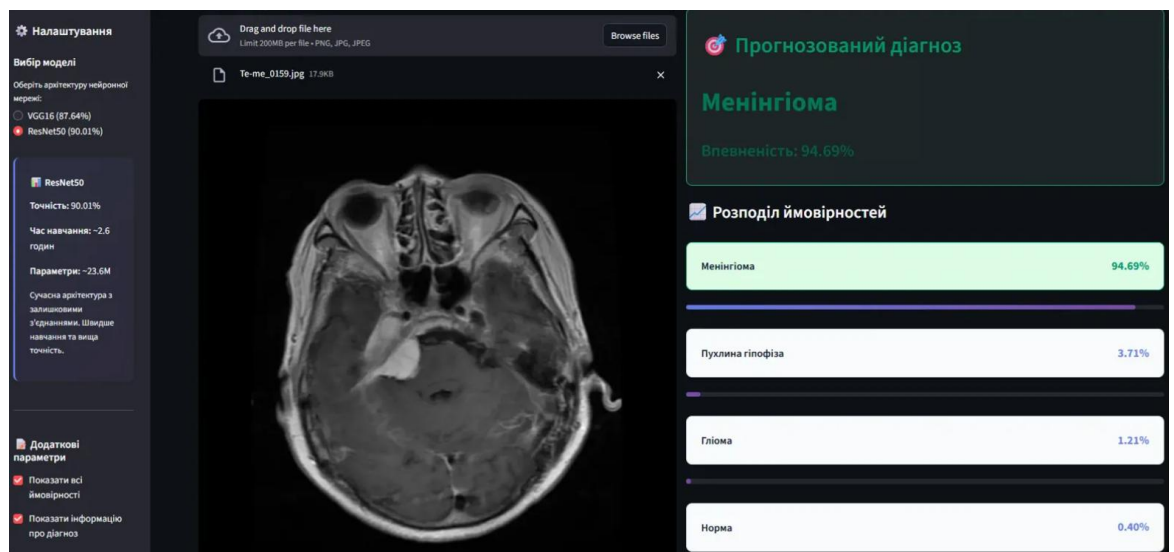


Рис. 3.3 — Успішна класифікація менінгіоми, модель Resnet50

У цьому випадку система коректно визначила тип пухлини з рівнем упевненості 94.69%, що суттєво перевищує ймовірності альтернативних класів та свідчить про надійність прийнятого рішення.

Приклад помилкової класифікації зображено на (рис. 3.4), де модель VGG16 неправильно ідентифікувала менінгіому як гліому. Аналіз цього випадку показав, що помилка зумовлена атиповим виглядом пухлини, зокрема розмитими межами, внутрішньою неоднорідністю та внутрішньомозковою локалізацією, що більше відповідає характерним ознакам гліом. Подібні помилки фіксувалися і під час тестування на розширеному наборі даних, де модель VGG16 плутала ці два класи приблизно у 10–11% випадків.

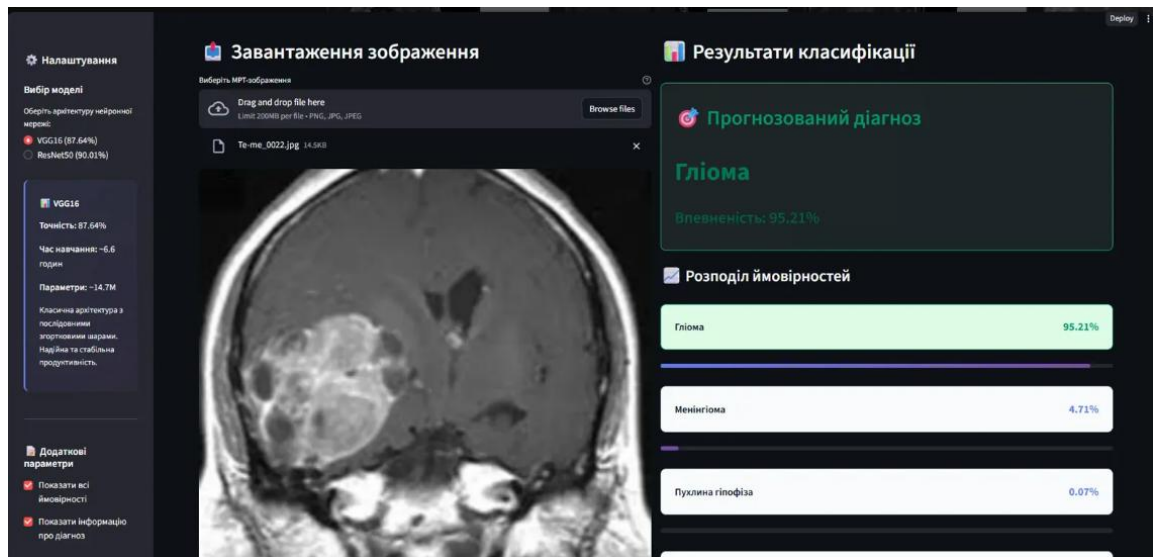


Рис. 3.4 — Невірна класифікація менінгіоми, модель VGG16

Застосунок розроблено з урахуванням вимог до відтворюваності експериментів та зручності подальшої модифікації програмного коду. Усі ключові параметри, зокрема вибір архітектури нейронної мережі та режим відображення результатів, керуються через інтерфейс користувача без необхідності зміни внутрішньої логіки системи. Такий підхід спрощує проведення повторних експериментів, дозволяє використовувати застосунок у навчальних і дослідницьких цілях, а також полегшує інтеграцію нових моделей або додаткових функціональних модулів у межах існуючої архітектури.

3.3. Порівняльний аналіз та інтерпретація результатів роботи моделей VGG16 і ResNet50

Для більш глибокого розуміння отриманих результатів класифікації було проведено порівняльний аналіз ефективності моделей VGG16 та ResNet50 на основі експериментальних даних, отриманих у процесі тестування та практичного застосування. Основна увага приділялася інтерпретації відмінностей у точності передбачень, рівні упевненості моделей і характері допущених помилок під час аналізу МРТ-зображень пухлин головного мозку.

Отримані результати свідчать про загальну перевагу моделі ResNet50 за більшістю показників якості. Вища точність на тестовій вибірці, а також стабільні значення precision, recall та F1-score вказують на здатність цієї моделі ефективно узагальнювати отримані знання та коректно працювати з різноманітними МРТ-зображеннями. Особливо важливим є те, що високі значення метрик зберігалися не лише для типових зразків, але й для зображень зі складною морфологією пухлин, що підтверджує надійність моделі в умовах підвищеної варіативності вхідних даних.

Модель VGG16, незважаючи на загалом прийнятні показники точності, виявила більшу чутливість до складних або атипових випадків. Це проявлялося як у зниженні рівня упевненості передбачень, так і у виникненні помилкових класифікацій, зокрема між морфологічно подібними класами пухлин. Аналіз таких випадків показав, що модель у подібних ситуаціях формує менш виражений розподіл ймовірностей, що може ускладнювати інтерпретацію результатів та потребувати додаткового контролю з боку спеціаліста.

Важливою особливістю отриманих результатів є кореляція між рівнем упевненості моделей і коректністю їхніх передбачень. У більшості випадків правильні класифікації супроводжувалися високими значеннями ймовірностей, тоді як помилки характеризувалися зниженим рівнем упевненості. Така закономірність має практичне значення, оскільки може бути

використана як додатковий індикатор надійності рішення системи в реальних умовах застосування.

Результати тестування моделей у прикладному програмному застосунку підтвердили висновки, отримані під час експериментального аналізу. Модель ResNet50 демонструвала стабільну поведінку та узгоджені результати для різних зображень, тоді як для VGG16 частіше спостерігалися випадки невизначеності або неоднозначного розподілу ймовірностей між класами. Це свідчить про те, що відмінності, зафіксовані на тестовій вибірці, зберігаються і в умовах, наближених до практичного використання.

З практичної точки зору отримані результати дозволяють розглядати модель ResNet50 як більш придатну для використання у системах комп'ютерної підтримки медичної діагностики. Водночас модель VGG16 може бути використана як допоміжний інструмент або базова модель для порівняльних експериментів, а також для попереднього аналізу зображень у випадках, де не вимагається максимальна точність.

Загалом проведений аналіз підтверджує, що вибір моделі істотно впливає не лише на числові показники якості, але й на характер прийнятих рішень та рівень їхньої інтерпретованості. Отримані результати створюють основу для подальшого вдосконалення системи автоматизованої класифікації пухлин головного мозку, зокрема шляхом розширення навчальної вибірки, оптимізації порогів прийняття рішень та інтеграції додаткових методів аналізу складних випадків.

Висновки до розділу 3

У третьому розділі було здійснено експериментальне оцінювання ефективності моделей глибокого навчання для класифікації пухлин головного мозку за МРТ-зображеннями та проаналізовано їхню поведінку в умовах, наближених до практичного використання. Застосування методів візуального аналізу передбачень дозволило глибше дослідити процес прийняття рішень

нейронними мережами, оцінити рівень їхньої упевненості та виявити характерні помилки класифікації, які не завжди очевидні при аналізі узагальнених числових показників.

Отримані результати свідчать, що модель ResNet50 демонструє стабільні та впевнені передбачення для різних типів пухлин і варіацій МРТ-зображень, зберігаючи високий рівень ймовірностей навіть у складних випадках із нечіткими морфологічними ознаками. Натомість модель VGG16 в окремих ситуаціях проявляє невизначеність, що переважно пов'язано з морфологічною подібністю між певними класами пухлин, зокрема гліомами та менінгіомами. Характерною ознакою таких помилок є знижений рівень упевненості, що може розглядатися як індикатор потенційно проблемних випадків.

Практичне тестування розроблених моделей у прикладному програмному застосунку підтвердило можливість їх використання в інтерактивному режимі без спеціальних технічних знань з боку користувача. Реалізований інтерфейс забезпечує наочне представлення результатів класифікації, дозволяє аналізувати розподіл ймовірностей між класами та підкреслює допоміжний характер системи у процесі прийняття клінічних рішень.

Порівняльний аналіз архітектур показав, що використання глибшої нейронної мережі із залишковими з'єднаннями забезпечує кращу здатність до узагальнення та більш високу стійкість до складних і атипових вхідних даних. Модель ResNet50 продемонструвала вищі значення основних метрик якості та більш стабільну поведінку на тестовій вибірці, що робить її доцільним вибором для побудови систем автоматизованої класифікації пухлин головного мозку за МРТ-зображеннями.

ВИСНОВОК

У даній кваліфікаційній магістерській роботі розв'язано науково-практичну задачу автоматизованого виявлення та класифікації пухлин головного мозку за МРТ-зображеннями з використанням методів глибокого та трансферного навчання. Основним результатом є розробка та порівняння моделей класифікації на основі згорткових нейронних мереж архітектур VGG16 та ResNet50.

У ході виконання роботи проведено комплексний аналіз предметної області, досліджено особливості МРТ-зображень та морфологічні характеристики основних типів пухлин (гліома, менінгіома, пухлина гіпофіза). Обґрунтовано доцільність застосування згорткових нейронних мереж як найбільш ефективного інструменту для аналізу високорозмірних медичних зображень.

Розроблено методологію автоматизованої класифікації МРТ-зображень у вигляді послідовного конвеєра обробки даних. Для розв'язання задачі використано публічний датасет Brain Tumor MRI, що містить 7023 МРТ-зображення, збалансовані за чотирма класами.

Реалізовано дві моделі класифікації на основі архітектур VGG16 та ResNet50 із замороженими згортковими основами та спеціалізованими класифікаційними головами. Навчання здійснювалося з використанням оптимізатора Adam, механізмів Early Stopping, ReduceLROnPlateau та Model Checkpoint.

За результатами експериментального дослідження на незалежній тестовій вибірці модель VGG16 досягла точності 87,64%, тоді як модель ResNet50 продемонструвала вищий результат — 90,01%. Аналіз метрик precision, recall та F1-score підтвердив перевагу ResNet50 за всіма показниками.

Детальний аналіз показав, що найкраще розпізнається клас «відсутність пухлини», а найскладнішим виявився клас «менінгіома». Модель ResNet50

продемонструвала суттєве покращення показників recall для класу «гліома», що є критично важливим з клінічної точки зору.

Практична цінність результатів полягає у можливості інтеграції розроблених моделей у веб-застосунки або медичні інформаційні системи для попереднього скринінгу МРТ-зображень, зменшення навантаження на лікарів-рентгенологів та підвищення об'єктивності діагностичних рішень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Litjens G. et al. A survey on deep learning in medical image analysis. *Medical Image Analysis*. 2017. Vol. 42. P. 60–88. DOI: 10.1016/j.media.2017.07.005
2. Bauer S. et al. A survey of MRI-based medical image analysis for brain tumor studies. *Physics in Medicine & Biology*. 2013. Vol. 58, No. 13. P. R97–R129. DOI: 10.1088/0031-9155/58/13/R97
3. Louis D. N. et al. The 2016 World Health Organization Classification of Tumors of the Central Nervous System: a summary. *Acta Neuropathologica*. 2016. Vol. 131, No. 6. P. 803–820. DOI: 10.1007/s00401-016-1545-1
4. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*. 2015. Vol. 521. P. 436–444. DOI: 10.1038/nature14539
5. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. Cambridge: MIT Press, 2016. 800 p.
6. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*. 2017. Vol. 60, No. 6. P. 84–90. DOI: 10.1145/3065386
7. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv*. 2015. URL: <https://arxiv.org/abs/1409.1556> (дата звернення: 28.10.2024).
8. Chollet F. *Deep Learning with Python*. 2nd ed. Shelter Island: Manning Publications, 2021. 504 p.
9. Bishop C. M. *Pattern Recognition and Machine Learning*. New York: Springer, 2006. 738 p.
10. He K. et al. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA, 2016. P. 770–778. DOI: 10.1109/CVPR.2016.90
11. Szeliski R. *Computer Vision: Algorithms and Applications*. 2nd ed. Cham: Springer, 2022. 1073 p.

12. Pan S. J., Yang Q. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*. 2010. Vol. 22, No. 10. P. 1345–1359. DOI: 10.1109/TKDE.2009.191
13. Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning*. Lille, France, 2015. Vol. 37. P. 448–456.
14. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization. *arXiv*. 2017. URL: <https://arxiv.org/abs/1412.6980> (дата звернення: 03.12.2024).
15. Srivastava N. et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 2014. Vol. 15, No. 1. P. 1929–1958.
16. Shorten C., Khoshgoftaar T. M. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*. 2019. Vol. 6. Article 60. DOI: 10.1186/s40537-019-0197-0
17. Fawcett T. An introduction to ROC analysis. *Pattern Recognition Letters*. 2006. Vol. 27, No. 8. P. 861–874. DOI: 10.1016/j.patrec.2005.10.010
18. Esteva A. et al. A guide to deep learning in healthcare. *Nature Medicine*. 2019. Vol. 25, No. 1. P. 24–29. DOI: 10.1038/s41591-018-0316-z
19. Topol E. J. High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine*. 2019. Vol. 25, No. 1. P. 44–56. DOI: 10.1038/s41591-018-0300-7
20. Shen D. et al. Deep Learning in Medical Image Analysis. *Annual Review of Biomedical Engineering*. 2017. Vol. 19. P. 221–248. DOI: 10.1146/annurev-bioeng-071516-044442
21. Abiwinanda N. et al. Brain Tumor Classification Using Convolutional Neural Network. *World Congress on Medical Physics and Biomedical Engineering* 2018. Prague, Czech Republic: Springer, 2019. P. 183–189. DOI: 10.1007/978-981-10-9035-6_33

22. Deepak S., Ameer P. M. Brain tumor classification using deep CNN features via transfer learning. *Computers in Biology and Medicine*. 2019. Vol. 11. Article 103345. DOI: 10.1016/j.compbimed.2019.103345
23. Swati Z. N. K. et al. Brain tumor classification for MR images using transfer learning and fine-tuning. *Computerized Medical Imaging and Graphics*. 2019. Vol. 75. P. 34–46. DOI: 10.1016/j.compmedimag.2019.05.001
24. Tandel G. S. et al. A Review on a Deep Learning Perspective in Brain Cancer Classification. *Cancers*. 2019. Vol. 11, No. 1. Article 111. DOI: 10.3390/cancers11010111
25. Ostrom Q. T. et al. CBTRUS Statistical Report: Primary Brain and Other Central Nervous System Tumors Diagnosed in the United States in 2015–2019. *Neuro-Oncology*. 2022. Vol. 24, Suppl. 5. P. v1–v95. DOI: 10.1093/neuonc/noac202
26. Weller M. et al. Glioma. *Nature Reviews Disease Primers*. 2015. Vol. 1. Article 15017. DOI: 10.1038/nrdp.2015.17
27. Goldbrunner R. et al. EANO guidelines for the diagnosis and treatment of meningiomas. *The Lancet Oncology*. 2016. Vol. 17, No. 9. P. e383–e391. DOI: 10.1016/S1470-2045(16)30321-7
28. Melmed S. Pituitary-Tumor Endocrinopathies. *New England Journal of Medicine*. 2020. Vol. 382, No. 10. P. 937–950. DOI: 10.1056/NEJMr1810772
29. Menze B. H. et al. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging*. 2015. Vol. 34, No. 10. P. 1993–2024. DOI: 10.1109/TMI.2014.2377694
30. Bakas S. et al. Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features. *Scientific Data*. 2017. Vol. 4. Article 170117. DOI: 10.1038/sdata.2017.117
31. Szegedy C. et al. Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA, 2015. P. 1–9. DOI: 10.1109/CVPR.2015.7298594

32. Huang G. et al. Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA, 2017. P. 2261–2269. DOI: 10.1109/CVPR.2017.243
33. Yosinski J. et al. How transferable are features in deep neural networks? Advances in Neural Information Processing Systems. 2014. Vol. 27. P. 3320–3328.
34. Tajbakhsh N. et al. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? IEEE Transactions on Medical Imaging. 2016. Vol. 35, No. 5. P. 1299–1312. DOI: 10.1109/TMI.2016.2535302
35. Loshchilov I., Hutter F. Decoupled Weight Decay Regularization. arXiv. 2019. URL: <https://arxiv.org/abs/1711.05101> (дата звернення: 09.12.2024).
36. Zhang H. et al. mixup: Beyond Empirical Risk Minimization. arXiv. 2018. URL: <https://arxiv.org/abs/1710.09412> (дата звернення: 13.12.2024).
37. Perez L., Wang J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. arXiv. 2017. URL: <https://arxiv.org/abs/1712.04621> (дата звернення: 15.12.2024).
38. Cheng J. et al. Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition. PLOS ONE. 2015. Vol. 10, No. 10. Article e0140381. DOI: 10.1371/journal.pone.0140381
39. Badza M. M., Barjaktarovic M. C. Classification of Brain Tumors from MRI Images Using a Convolutional Neural Network. Applied Sciences. 2020. Vol. 10, No. 6. Article 1999. DOI: 10.3390/app10061999
40. Selvaraju R. R. et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. International Journal of Computer Vision. 2020. Vol. 128, No. 2. P. 336–359. DOI: 10.1007/s11263-019-01228-7

ДОДАТКИ

```

print("\n" + "="*80)
print("PHASE 3: DATA PREPROCESSING AND AUGMENTATION")
print("="*80)

# Image parameters
IMG_HEIGHT = 224
IMG_WIDTH = 224
BATCH_SIZE = 32
VALIDATION_SPLIT = 0.2

print(f"\n🔧 Preprocessing Configuration:")
print(f" - Image Size: {IMG_HEIGHT}x{IMG_WIDTH}")
print(f" - Batch Size: {BATCH_SIZE}")
print(f" - Validation Split: {VALIDATION_SPLIT*100}%")

# Data Augmentation for Training
train_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input, # VGG16 preprocessing
    validation_split=VALIDATION_SPLIT,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True,
    zoom_range=0.2,
    shear_range=0.15,
    fill_mode='nearest'

```

```

)

# Only preprocessing for validation (no augmentation)
val_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input,
    validation_split=VALIDATION_SPLIT
)

# Test data generator (no augmentation)
test_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input
)

# Create data generators
train_generator = train_datagen.flow_from_directory(
    TRAIN_DIR,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training',
    shuffle=True,
    seed=42
)

validation_generator = val_datagen.flow_from_directory(
    TRAIN_DIR,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    batch_size=BATCH_SIZE,

```

```

class_mode='categorical',
subset='validation',
shuffle=False,
seed=42
)

test_generator = test_datagen.flow_from_directory(
    TEST_DIR,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    shuffle=False
)

print(f"\n☑ Data Generators Created:")
print(f" - Training samples: {train_generator.samples}")
print(f" - Validation samples: {validation_generator.samples}")
print(f" - Test samples: {test_generator.samples}")
print(f"\n Class indices: {train_generator.class_indices}")

print("\n" + "="*80)
print("PHASE 4: VGG16 MODEL ARCHITECTURE")
print("="*80)

# Load pre-trained VGG16 model
base_model = VGG16(
    weights='imagenet',
    include_top=False,

```

```

        input_shape=(IMG_HEIGHT, IMG_WIDTH, 3)
    )

    # Freeze base model layers
    for layer in base_model.layers:
        layer.trainable = False

    print(f"\n  VGG16 Base Model Loaded:")
    print(f" - Total layers: {len(base_model.layers)}")
    print(f" - Trainable: {sum([layer.trainable for layer in base_model.layers])} layers")
    print(f" - Frozen: {sum([not layer.trainable for layer in base_model.layers])} layers")

    # Build complete model
    model = Sequential([
        base_model,
        GlobalAveragePooling2D(),
        BatchNormalization(),
        Dense(512, activation='relu'),
        Dropout(0.5),
        BatchNormalization(),
        Dense(256, activation='relu'),
        Dropout(0.3),
        Dense(NUM_CLASSES, activation='softmax')
    ])

    # Compile model
    model.compile(

```

```

optimizer=Adam(learning_rate=0.0001),
loss='categorical_crossentropy',
metrics=['accuracy', tf.keras.metrics.Precision(), tf.keras.metrics.Recall()]
)

```

```

print("\n🏗 Complete Model Architecture:")
model.summary()

```

```

# Calculate parameters
total_params = model.count_params()
trainable_params = sum([tf.size(w).numpy() for w in model.trainable_weights])
non_trainable_params = total_params - trainable_params

```

```

print(f"\n📊 Model Parameters:")
print(f" - Total parameters: {total_params:,}")
print(f" - Trainable parameters: {trainable_params:,}")
print(f" - Non-trainable parameters: {non_trainable_params:,}")

```

```

print("\n" + "="*80)
print("PHASE 5: TRAINING CONFIGURATION")
print("="*80)

```

```

# Define callbacks
checkpoint = ModelCheckpoint(
    'best_vgg16_model.keras',
    monitor='val_accuracy',
    save_best_only=True,
    mode='max',

```

```

        verbose=1
    )

    early_stopping = EarlyStopping(
        monitor='val_loss',
        patience=10,
        restore_best_weights=True,
        verbose=1
    )

    reduce_lr = ReduceLROnPlateau(
        monitor='val_loss',
        factor=0.5,
        patience=5,
        min_lr=1e-7,
        verbose=1
    )

    callbacks = [checkpoint, early_stopping, reduce_lr]

    print("\n⚙️ Training Callbacks:")
    print(" ✓ ModelCheckpoint - Saves best model based on validation accuracy")
    print(" ✓ EarlyStopping - Stops training if no improvement for 10 epochs")
    print(" ✓ ReduceLROnPlateau - Reduces learning rate if validation loss plateaus")

    print("\n" + "="*80)
    print("PHASE 6: MODEL TRAINING")
    print("="*80)

```

EPOCHS = 50

```
print(f"\n🚀 Starting Training:")
print(f" - Epochs: {EPOCHS}")
print(f" - Steps per epoch: {train_generator.samples // BATCH_SIZE}")
print(f" - Validation steps: {validation_generator.samples // BATCH_SIZE}")
print("\n")
```

```
history = model.fit(
    train_generator,
    epochs=EPOCHS,
    validation_data=validation_generator,
    callbacks=callbacks,
    verbose=1
)
```

```
print("\n✅ Training Completed!")
```

```
print("\n" + "="*80)
print("PHASE 7: TRAINING PERFORMANCE ANALYSIS")
print("="*80)
```

Plot training history

```
fig, axes = plt.subplots(2, 2, figsize=(16, 12))
```

Accuracy

```
axes[0, 0].plot(history.history['accuracy'], label='Training Accuracy', linewidth=2)
axes[0, 0].plot(history.history['val_accuracy'], label='Validation Accuracy',
linewidth=2)
axes[0, 0].set_title('Model Accuracy', fontsize=14, fontweight='bold')
axes[0, 0].set_xlabel('Epoch', fontsize=12)
axes[0, 0].set_ylabel('Accuracy', fontsize=12)
axes[0, 0].legend(loc='lower right')
axes[0, 0].grid(alpha=0.3)
```

Loss

```
axes[0, 1].plot(history.history['loss'], label='Training Loss', linewidth=2)
axes[0, 1].plot(history.history['val_loss'], label='Validation Loss', linewidth=2)
axes[0, 1].set_title('Model Loss', fontsize=14, fontweight='bold')
axes[0, 1].set_xlabel('Epoch', fontsize=12)
axes[0, 1].set_ylabel('Loss', fontsize=12)
axes[0, 1].legend(loc='upper right')
axes[0, 1].grid(alpha=0.3)
```

Precision

```
axes[1, 0].plot(history.history['precision'], label='Training Precision', linewidth=2)
axes[1, 0].plot(history.history['val_precision'], label='Validation Precision',
linewidth=2)
axes[1, 0].set_title('Model Precision', fontsize=14, fontweight='bold')
axes[1, 0].set_xlabel('Epoch', fontsize=12)
axes[1, 0].set_ylabel('Precision', fontsize=12)
axes[1, 0].legend(loc='lower right')
axes[1, 0].grid(alpha=0.3)
```

Recall

```

axes[1, 1].plot(history.history['recall'], label='Training Recall', linewidth=2)
axes[1, 1].plot(history.history['val_recall'], label='Validation Recall', linewidth=2)
axes[1, 1].set_title('Model Recall', fontsize=14, fontweight='bold')
axes[1, 1].set_xlabel('Epoch', fontsize=12)
axes[1, 1].set_ylabel('Recall', fontsize=12)
axes[1, 1].legend(loc='lower right')
axes[1, 1].grid(alpha=0.3)

```

```

plt.suptitle('Training Performance Metrics', fontsize=16, fontweight='bold')
plt.tight_layout()
plt.show()

```

Print best metrics

```

best_epoch = np.argmax(history.history['val_accuracy'])
print(f"\n 🏆 Best Model Performance (Epoch {best_epoch + 1}):")
print(f" - Training Accuracy: {history.history['accuracy'][best_epoch]:.4f}")
print(f" - Validation Accuracy: {history.history['val_accuracy'][best_epoch]:.4f}")
print(f" - Training Loss: {history.history['loss'][best_epoch]:.4f}")
print(f" - Validation Loss: {history.history['val_loss'][best_epoch]:.4f}")

```

Evaluate on test set

```

test_loss, test_accuracy, test_precision, test_recall = model.evaluate(
    test_generator,
    verbose=1
)

```

```

print(f"\n 📊 Test Set Performance:")
print(f" - Test Accuracy: {test_accuracy:.4f} ({test_accuracy*100:.2f}%)")

```

```

print(f' - Test Loss: {test_loss:.4f}')
print(f' - Test Precision: {test_precision:.4f}')
print(f' - Test Recall: {test_recall:.4f}')
print(f' - Test F1-Score: {2 * (test_precision * test_recall) / (test_precision +
test_recall):.4f}')

# Get predictions
test_generator.reset()
y_pred_probs = model.predict(test_generator, verbose=1)
y_pred = np.argmax(y_pred_probs, axis=1)
y_true = test_generator.classes

# Confusion Matrix
cm = confusion_matrix(y_true, y_pred)
class_names = list(test_generator.class_indices.keys())

# Plot confusion matrix
plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=class_names, yticklabels=class_names,
            cbar_kws={'label': 'Count'})
plt.title('Confusion Matrix', fontsize=16, fontweight='bold', pad=20)
plt.ylabel('True Label', fontsize=12)
plt.xlabel('Predicted Label', fontsize=12)
plt.tight_layout()
plt.show()

# Normalized Confusion Matrix
cm_normalized = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

```

```

plt.figure(figsize=(10, 8))
sns.heatmap(cm_normalized, annot=True, fmt='.2%', cmap='Greens',
            xticklabels=class_names, yticklabels=class_names,
            cbar_kws={'label': 'Percentage'})
plt.title('Normalized Confusion Matrix (%)', fontsize=16, fontweight='bold',
          pad=20)
plt.ylabel('True Label', fontsize=12)
plt.xlabel('Predicted Label', fontsize=12)
plt.tight_layout()
plt.show()

# Classification Report
print("\n" + "="*60)
print("CLASSIFICATION REPORT")
print("="*60)
print(classification_report(y_true, y_pred, target_names=class_names, digits=4))

# Per-class accuracy
class_accuracy = cm.diagonal() / cm.sum(axis=1)
print("\n📊 Per-Class Accuracy:")
for i, cls in enumerate(class_names):
    print(f" {cls:<15}: {class_accuracy[i]:.4f} ({class_accuracy[i]*100:.2f}%)")

# Binarize labels for ROC curve
y_true_bin = label_binarize(y_true, classes=range(NUM_CLASSES))

# Compute ROC curve and AUC for each class
fpr = dict()
tpr = dict()

```

```

roc_auc = dict()

for i in range(NUM_CLASSES):
    fpr[i], tpr[i], _ = roc_curve(y_true_bin[:, i], y_pred_probs[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Plot ROC curves
plt.figure(figsize=(12, 8))
colors = ['blue', 'red', 'green', 'orange']

for i, color in enumerate(colors):
    plt.plot(fpr[i], tpr[i], color=color, lw=2,
             label=f'{class_names[i]} (AUC = {roc_auc[i]:.4f})')

plt.plot([0, 1], [0, 1], 'k--', lw=2, label='Random Classifier')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate', fontsize=12)
plt.ylabel('True Positive Rate', fontsize=12)
plt.title('ROC Curves - Multi-Class Classification', fontsize=14, fontweight='bold')
plt.legend(loc='lower right', fontsize=10)
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()

print("\n 🌀 AUC Scores:")
for i, cls in enumerate(class_names):
    print(f' {cls:<15}: {roc_auc[i]:.4f}')

```

```

print(f'  {'Mean AUC':<15}: {np.mean(list(roc_auc.values())):.4f}')

def visualize_predictions(generator, model, num_samples=16):
    """Visualize model predictions on sample images"""
    generator.reset()

    # Get a batch
    x_batch, y_batch = next(generator)
    predictions = model.predict(x_batch[:num_samples], verbose=0)

    fig, axes = plt.subplots(4, 4, figsize=(16, 16))
    axes = axes.flatten()

    for i in range(min(num_samples, len(x_batch))):
        # Reverse preprocessing for visualization
        img = x_batch[i].copy()
        img += [103.939, 116.779, 123.68] # Reverse VGG16 preprocessing
        img = img[:, :, ::-1] # BGR to RGB
        img = np.clip(img / 255.0, 0, 1)

        true_label = class_names[np.argmax(y_batch[i])]
        pred_label = class_names[np.argmax(predictions[i])]
        confidence = np.max(predictions[i]) * 100

        axes[i].imshow(img)

        # Color code: green for correct, red for incorrect
        color = 'green' if true_label == pred_label else 'red'

```

```
axes[i].set_title(f'True: {true_label}\nPred: {pred_label}'  
({confidence:.1f}%'),
```

```
        fontsize=10, color=color, fontweight='bold')
```

```
axes[i].axis('off')
```

```
plt.suptitle('Sample Predictions (Green=Correct, Red=Incorrect)',
```

```
        fontsize=16, fontweight='bold')
```

```
plt.tight_layout()
```

```
plt.show()
```

```
visualize_predictions(test_generator, model, num_samples=16)
```

```
summary_data = {
```

```
    'Metric': ['Accuracy', 'Precision', 'Recall', 'F1-Score', 'Loss'],
```

```
    'Training': [  
        f'{history.history['accuracy'][-1]:.4f}',  
        f'{history.history['precision'][-1]:.4f}',  
        f'{history.history['recall'][-1]:.4f}',  
        f'{2 * history.history['precision'][-1] * history.history['recall'][-1] /  
(history.history['precision'][-1] + history.history['recall'][-1]):.4f}',  
        f'{history.history['loss'][-1]:.4f}'  
    ],
```

```
    'Validation': [  
        f'{history.history['val_accuracy'][-1]:.4f}',  
        f'{history.history['val_precision'][-1]:.4f}',  
        f'{history.history['val_recall'][-1]:.4f}',  
        f'{2 * history.history['val_precision'][-1] * history.history['val_recall'][-1] /  
(history.history['val_precision'][-1] + history.history['val_recall'][-1]):.4f}',  
        f'{history.history['val_loss'][-1]:.4f}'  
    ]
```

```

],
'Test': [
    f'{test_accuracy:.4f}',
    f'{test_precision:.4f}',
    f'{test_recall:.4f}',
    f'{2 * test_precision * test_recall / (test_precision + test_recall):.4f}',
    f'{test_loss:.4f}'
]
}

```

```

summary_df = pd.DataFrame(summary_data)
print("\n")
print(summary_df.to_string(index=False))

```

Додаток Б

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
import random
from pathlib import Path
import warnings
warnings.filterwarnings('ignore')

# Deep Learning Libraries
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Model, Sequential

```

```

from tensorflow.keras.layers import Dense, Dropout, GlobalAveragePooling2D,
BatchNormalization
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping,
ReduceLROnPlateau
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.applications.resnet50 import preprocess_input

# Metrics and Evaluation
from sklearn.metrics import classification_report, confusion_matrix, roc_curve,
auc
from sklearn.preprocessing import label_binarize

# Set random seeds for reproducibility
SEED = 42
np.random.seed(SEED)
tf.random.set_seed(SEED)
random.seed(SEED)

# Конфігурація
BASE_DIR = 'mri-images'
TRAIN_DIR = os.path.join(BASE_DIR, 'Training')
TEST_DIR = os.path.join(BASE_DIR, 'Testing')

IMG_HEIGHT = 224
IMG_WIDTH = 224
BATCH_SIZE = 32
VALIDATION_SPLIT = 0.2
EPOCHS = 50

```

```
NUM_CLASSES = 4
```

```
# Аугментація даних для навчання
```

```
train_datagen = ImageDataGenerator(  
    preprocessing_function=preprocess_input,  
    validation_split=VALIDATION_SPLIT,  
    rotation_range=20,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    horizontal_flip=True,  
    zoom_range=0.2,  
    shear_range=0.15,  
    fill_mode='nearest'  
)
```

```
# Генератор для валідації (без аугментації)
```

```
val_datagen = ImageDataGenerator(  
    preprocessing_function=preprocess_input,  
    validation_split=VALIDATION_SPLIT  
)
```

```
# Генератор для тестування (без аугментації)
```

```
test_datagen = ImageDataGenerator(  
    preprocessing_function=preprocess_input  
)
```

```
# Створення генераторів даних
```

```
train_generator = train_datagen.flow_from_directory(  
    TRAIN_DIR,  
    target_size=(IMG_HEIGHT, IMG_WIDTH),
```

```

        batch_size=BATCH_SIZE,
        class_mode='categorical',
        subset='training',
        shuffle=True,
        seed=SEED
    )

validation_generator = val_datagen.flow_from_directory(
    TRAIN_DIR,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation',
    shuffle=False,
    seed=SEED
)

test_generator = test_datagen.flow_from_directory(
    TEST_DIR,
    target_size=(IMG_HEIGHT, IMG_WIDTH),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    shuffle=False
)

# Побудова моделі ResNet50
base_model = ResNet50(
    weights='imagenet',
    include_top=False,
    input_shape=(IMG_HEIGHT, IMG_WIDTH, 3)

```

)

Заморожуємо базову модель

base_model.trainable = False

Побудова повної моделі

```
model = Sequential([
    base_model,
    GlobalAveragePooling2D(),
    BatchNormalization(),
    Dense(512, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)),
    Dropout(0.5),
    BatchNormalization(),
    Dense(256, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001)),
    Dropout(0.4),
    BatchNormalization(),
    Dense(128, activation='relu'),
    Dropout(0.3),
    Dense(NUM_CLASSES, activation='softmax')
])
```

Компіляція моделі

```
model.compile(
    optimizer=Adam(learning_rate=0.0001),
    loss='categorical_crossentropy',
    metrics=['accuracy', tf.keras.metrics.Precision(), tf.keras.metrics.Recall()]
)
```

print("\n🔍 Архітектура моделі:")

model.summary()

```

# Callbacks
os.makedirs('models', exist_ok=True)

checkpoint = ModelCheckpoint(
    'models/best_resnet50_model.keras',
    monitor='val_accuracy',
    save_best_only=True,
    mode='max',
    verbose=1
)

early_stopping = EarlyStopping(
    monitor='val_loss',
    patience=15,
    restore_best_weights=True,
    verbose=1,
    min_delta=0.001
)

reduce_lr = ReduceLROnPlateau(
    monitor='val_loss',
    factor=0.5,
    patience=7,
    min_lr=1e-7,
    verbose=1,
    min_delta=0.001
)

callbacks = [checkpoint, early_stopping, reduce_lr]

```

```
# Навчання моделі
```

```
print("\n🚀 Початок навчання...")
```

```
steps_per_epoch = train_generator.samples // BATCH_SIZE
```

```
validation_steps = validation_generator.samples // BATCH_SIZE
```

```
history = model.fit(  
    train_generator,  
    steps_per_epoch=steps_per_epoch,  
    epochs=EPOCHS,  
    validation_data=validation_generator,  
    validation_steps=validation_steps,  
    callbacks=callbacks,  
    verbose=1  
)
```

```
print("\n✅ Навчання завершено!")
```

```
# Візуалізація результатів навчання
```

```
fig, axes = plt.subplots(2, 2, figsize=(16, 12))
```

```
# Точність
```

```
axes[0, 0].plot(history.history['accuracy'], label='Training Accuracy', linewidth=2)
```

```
axes[0, 0].plot(history.history['val_accuracy'], label='Validation Accuracy',  
linewidth=2)
```

```
axes[0, 0].set_title('Model Accuracy', fontsize=14, fontweight='bold')
```

```
axes[0, 0].set_xlabel('Epoch')
```

```
axes[0, 0].set_ylabel('Accuracy')
```

```

axes[0, 0].legend()
axes[0, 0].grid(alpha=0.3)

# Втрети
axes[0, 1].plot(history.history['loss'], label='Training Loss', linewidth=2)
axes[0, 1].plot(history.history['val_loss'], label='Validation Loss', linewidth=2)
axes[0, 1].set_title('Model Loss', fontsize=14, fontweight='bold')
axes[0, 1].set_xlabel('Epoch')
axes[0, 1].set_ylabel('Loss')
axes[0, 1].legend()
axes[0, 1].grid(alpha=0.3)

# Precision
axes[1, 0].plot(history.history['precision'], label='Training Precision', linewidth=2)
axes[1, 0].plot(history.history['val_precision'], label='Validation Precision',
linewidth=2)
axes[1, 0].set_title('Model Precision', fontsize=14, fontweight='bold')
axes[1, 0].set_xlabel('Epoch')
axes[1, 0].set_ylabel('Precision')
axes[1, 0].legend()
axes[1, 0].grid(alpha=0.3)

# Recall
axes[1, 1].plot(history.history['recall'], label='Training Recall', linewidth=2)
axes[1, 1].plot(history.history['val_recall'], label='Validation Recall', linewidth=2)
axes[1, 1].set_title('Model Recall', fontsize=14, fontweight='bold')
axes[1, 1].set_xlabel('Epoch')
axes[1, 1].set_ylabel('Recall')
axes[1, 1].legend()
axes[1, 1].grid(alpha=0.3)

```

```
plt.suptitle('Training Performance Metrics', fontsize=16, fontweight='bold')
plt.tight_layout()
plt.show()
```

```
# Оцінка на тестовому наборі
```

```
print("\n🔍 Оцінка на тестовому наборі...")
```

```
test_steps = test_generator.samples // BATCH_SIZE + 1
```

```
test_results = model.evaluate(test_generator, steps=test_steps, verbose=1)
```

```
test_loss = test_results[0]
```

```
test_accuracy = test_results[1]
```

```
test_precision = test_results[2]
```

```
test_recall = test_results[3]
```

```
test_f1 = 2 * (test_precision * test_recall) / (test_precision + test_recall)
```

```
print(f"\n📊 Результати на тестовому наборі:")
```

```
print(f" Test Accuracy: {test_accuracy:.4f} ({test_accuracy*100:.2f}%)")
```

```
print(f" Test Loss: {test_loss:.4f}")
```

```
print(f" Test Precision: {test_precision:.4f}")
```

```
print(f" Test Recall: {test_recall:.4f}")
```

```
print(f" Test F1-Score: {test_f1:.4f}")
```

```
# Передбачення
```

```
test_generator.reset()
```

```
y_pred_probs = model.predict(test_generator, steps=test_steps, verbose=1)
```

```
y_pred = np.argmax(y_pred_probs, axis=1)
```

```
y_true = test_generator.classes[:len(y_pred)]
```

```

# Матриця плутанини
class_names = list(train_generator.class_indices.keys())
cm = confusion_matrix(y_true, y_pred)

plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=class_names, yticklabels=class_names,
            cbar_kws={'label': 'Count'})
plt.title('Confusion Matrix', fontsize=16, fontweight='bold')
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.tight_layout()
plt.show()

# Нормалізована матриця плутанини
cm_normalized = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
plt.figure(figsize=(10, 8))
sns.heatmap(cm_normalized, annot=True, fmt='.2%', cmap='Greens',
            xticklabels=class_names, yticklabels=class_names,
            cbar_kws={'label': 'Percentage'})
plt.title('Normalized Confusion Matrix (%)', fontsize=16, fontweight='bold')
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.tight_layout()
plt.show()

# Звіт класифікації
print("\n" + "="*60)
print("CLASSIFICATION REPORT")

```

```

print("="*60)
print(classification_report(y_true, y_pred, target_names=class_names, digits=4))

# ROC-криві
y_true_bin = label_binarize(y_true, classes=range(NUM_CLASSES))

fpr = dict()
tpr = dict()
roc_auc = dict()

for i in range(NUM_CLASSES):
    fpr[i], tpr[i], _ = roc_curve(y_true_bin[:, i], y_pred_probs[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

plt.figure(figsize=(12, 8))
colors = ['blue', 'red', 'green', 'orange']

for i, color in enumerate(colors):
    plt.plot(fpr[i], tpr[i], color=color, lw=2,
             label=f'{class_names[i]} (AUC = {roc_auc[i]:.4f})')

plt.plot([0, 1], [0, 1], 'k--', lw=2, label='Random Classifier')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curves - Multi-Class Classification', fontsize=14, fontweight='bold')
plt.legend(loc='lower right')
plt.grid(alpha=0.3)
plt.tight_layout()

```

```
plt.show()
```

```
print("\n🌀 AUC Scores:")
```

```
for i, cls in enumerate(class_names):
```

```
    print(f" {cls:<15}: {roc_auc[i]:.4f}")
```

```
print(f" {'Mean AUC':<15}: {np.mean(list(roc_auc.values())):.4f}")
```