

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка
рівень вищої освіти – другий (магістерський)

Дослідження методів класифікації геморагії на томографічних зображеннях мозку
(тема)

Виконав: студент 2 курсу, групи ПЗМ-18-1

Панченко Д.В.
(прізвище, ініціали)

спеціальності 121– Інженерія програмного забезпечення
(код і повна назва спеціальності)

Освітньо-наукової програми
(тип програми)

Інженерія програмного забезпечення
(тип програми)

Керівник к.т.н., доц. Турута О.П.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Дудар З.В.
(прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти – другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

Студентові Панченку Дмитру Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи (проекту): Дослідження методів класифікації геморагії на томографічних зображеннях мозку

затверджена наказом по університету від "27" березня 2020 р. №473 Ст

2. Термін подання студентом роботи (проекту)

15 травня 2020 р.

3. Вихідні дані до роботи (проекту) скрипти для аналізу набору даних, для тренування нейромережі, візуалізації вивчених вікон та отримання прогнозу з нейромережі, пояснювальна записка. Використовувати ОС Linux, мову програмування Python.

4. Перелік питань, що потрібно опрацювати в роботі: мета роботи, аналіз проблемної галузі і постановка задачі, огляд існуючих методів з розпізнавання томографічних знімків, застосування методів для поставленої задачі, аналіз якості моделей генерації опису, пошук способів підвищення якості та аналіз їх ефективності.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка*
1.	Аналіз предметної галузі	03 квітня 2020 р.	
2.	Огляд існуючих методів	10 квітня 2020 р.	
3.	Дослідження методів класифікації томографічних зображень	17 квітня 2020 р.	
4.	Підготовка пояснювальної записки	24 квітня 2020 р.	
5.	Спецчастина	29 квітня 2020 р.	
6.	Підготовка презентації та доповіді	30 квітня 2020 р.	
7.	Попередній захист	05 травня 2020 р.	
8.	Нормоконтроль, рецензування	06 травня 2020 р.	
9.	Занесення диплома в електронний архів	13 травня 2020 р.	
10.	Допуск до захисту у зав. кафедри	14 травня 2020 р.	
* заповнюється вручну після виконання чергового пункту			

Дата видачі завдання 27 березня 2020 р.

Студент _____
(підпис)

Керівник роботи (проекту) _____ к.т.н., доцент Турута О.П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Атестаційна робота магістра містить: 87 с., 23 рис., 4 табл., 12 формул, 45 джерел.

ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, КОМП'ЮТЕРНА ТОМОГРАФІЯ, КОМП'ЮТЕРНИЙ ЗІР, PYTORCH.

Метою роботи є дослідження, аналіз та реалізація методів класифікації томографічних зображень черепа з метою пошуку крововиливів у мозок. Ці методи надають можливість автоматизувати системи аналізу медичних даних та прискорити діагностування при травмах голови та підозрах на інсульт

Методи розробки базуються на інструментах аналізу даних та машинного навчання. Зокрема, використовується мова програмування Python, а також екосистема фреймворку PyTorch та fast.ai.

В результаті роботи пропонується метод класифікації томографічних зображень, що досягає найвищої якості розпізнавання для задачі виявлення геморагії.

COMPUTED TOMOGRAPHY, COMPUTER VISION, CONVOLUTIONAL NEURAL NETWORK, IMAGE CLASSIFICATION, PYTORCH.

The aim of this work is to research, analysis and implementation of classification methods for cranial CT scans in order to detect brain hemorrhage. Such methods will allow us to automate medical data analysis systems and speed up the diagnostics of head traumas or strokes.

Implementation methods are based on data analysis and machine learning tools. Python programming language with PyTorch and fast.ai frameworks is used.

As a result, method of CT scans classification is proposed that achieve state-of-the-art recognition performance for hemorrhage classification task.

ЗМІСТ

Вступ.....	6
1 Аналіз проблемної галузі.....	8
1.1 Застосування машинного навчання для дослідження медичних зображень.....	8
1.2 Огляд задачі аналізу томографічних зображень.....	10
2 Постановка задачі.....	17
2.1 Загальні методи класифікації зображень.....	17
2.2 Методи класифікації томографічних зображень	23
2.3 Формалізація задачі	26
2.4 Метрика якості моделі.....	28
3 Методика досліджень	29
3.1 Архітектури згорткової частини мережі.....	29
3.2 Налаштування гіперпараметрів	31
3.3 Налаштування вікон.....	37
3.4 Технології та засоби.....	42
4 Аналіз результатів	44
4.1 Експерименти з архітектурами	44
4.2 Експерименти з пайплайном навчання.....	46
4.3 Експерименти з вікнами	47
4.4 Результати та подальші дослідження.....	50
Висновки	52
Перелік джерел посилання	53
Додаток А Наукові публікації.....	57
Додаток Б Лістинг коду	67
Додаток В Відгук.....	76
Додаток Г Слайди презентації	78

ВСТУП

В останній час однією з областей, що розвиваються найшвидше, став штучний інтелект. Розвиток штучного інтелекту проявляється в багатьох областях, але, мабуть, найбільшого прогресу за попередні десять років наука досягла саме в сфері комп'ютерного зору.

Наукова революція машинного зору почалась у 2007 році, коли вченим вдалося досягти дуже великого приросту у швидкості обчислень завдяки використуванню графічних прискорювачів замість звичайних процесорів для високонавантажених алгоритмів. Наступний крок у еволюції розпізнавання зображень було зроблено командою Джофрі Хінтона, Алекса Крижевського та Іллі Сутскевера, які у 2012 році запропонували підхід до класифікації зображень на основі нейронних мереж. Їхній алгоритм класифікації, згортова нейронна мережа AlexNet [1], майже вдвічі поліпшив найсучасніші на той момент моделі комп'ютерного зору, влаштувавши прорив в науці про дані і зокрема в семантичному аналізі зображень.

Зараз, через вісім років після так званої нейромережевої революції, згортові мережі зайняли ведучі позиції в майже усіх завданнях розпізнавання образів та машинного зору, таких як класифікація зображень та детекція об'єктів.

В останні кілька років однією з галузей, що починає адаптувати машинне навчання до своїх задач, стала медицина [2]. Найпопулярнішою задачею машинного навчання в медицині є аналіз медичних зображень.

Задача аналізу томографічних зображень мозку є однією з найактуальніших, та у той час вона несе багато специфіки, що пов'язана здебільшого з форматом медичних зображень та складністю анотації достатньо великих датасетів.

Метою цієї роботи є покращення існуючих методів автоматизованого аналізу томограм та виявлення геморагій мозку. Серед задач роботи, що планується виконати, дослідження методів класифікації томографічних зображень, їх тестування для задачі класифікації видів геморагії, а також створення новітнього

алгоритму для покращення розпізнавання томографічних знімків на предмет знаходження крововиливів та оцінка точності отриманої моделі.

Об'єктом дослідження є розпізнавання томографічних зображень засобами комп'ютерного зору.

Предметом дослідження є максимально точна класифікація внутрішньої кровотечі на томографічних зображеннях мозку людини.

Методи дослідження включають аналіз існуючих методів класифікації зображень та власне специфіку томографічних знімків, через яку класичні алгоритми для неї не працюють, порівняння різних архітектур нейронних мереж, налаштування гіперпараметрів, налаштування томографічних вікон та порівняльний аналіз результатів з використанням формальних математичних метрик якості.

В рамках роботи удосконалено існуючі методи автоматичної оптимізації томографічних вікон та отримано кращі значення метрики на задачі класифікації геморагії, а також продемонстрована переносимість результатів для різних архітектур нейронних мереж.

Описані методи використано в рамках бенчмарку алгоритмів машинного навчання RSNA Intracranial Hemorrhage Detection, що проводився за запитом Північно-Американського суспільства радіологів.

Робота базується на попередніх результатах автору, опублікованих у науковому журналі Nature Methods в статті «Analysis of the Human Protein Atlas Image Classification competition».

Результати роботи допомагають покращити розпізнавання кровотеч в мозок, що потенціально може бути використано для створення систем автоматизованої діагностики, тому робота націлена на досягнення суспільно-корисних гуманістичних цілей з використанням методів комп'ютерного зору та штучного інтелекту.

1 АНАЛІЗ ПРОБЛЕМНОЇ ГАЛУЗІ

1.1 Застосування машинного навчання для дослідження медичних зображень

Існує ціла низка алгоритмів та інструментів для аналізу зображень, але в останні роки майже в усіх задачах комп'ютерного зору найкращі результати демонструють саме алгоритми машинного навчання, а саме моделі глибокого навчання – згорткові та капсульні нейронні мережі.

Всі задачі машинного навчання розподіляються на три основні категорії [3]:

а) задача навчання з учителем – це клас проблем машинного навчання, в якому алгоритм має побудувати модель, що приймає на вхід сигнал, представлений векторами (у загальному випадку – тензорами) у якомусь багатовимірному просторі, та зіставляє йому вихідний сигнал (у загальному випадку – також тензор). Навчання з учителем завжди потребує анотованого набору даних, де вхідним точкам зіставляється вихідні мітки – так звана цільова змінна [4];

1) класифікація – це найтипівіший підвид задачі навчання з учителем, в якому цільова змінна представлена дискретною величиною. Ця змінна зазвичай називається класом. Результат побудування класифікаційної моделі – алгоритм, що може приймати на вхід нові спостереження та розподіляти їх по вивченим категоріям. Класифікація поділяється на бінарну (коли цільова змінна приймає значення 0 чи 1), багатокласову чи мультикласову (коли цільова змінна відноситься до однієї з багатьох категорій) та мультилейбельну (коли цільова змінна може відноситися до кількох з багатьох класів). Також, вона може поділятися на «м'яку» та «жорстку» класифікацію. В першому випадку, модель має прогнозувати вірогідність приналежності до кожного класу, а у другому – тільки саму мітку класу. Це досить важлива різниця, тому що деякі моделі (наприклад, метод опорних векторів) не здатні

будувати ймовірнісну функцію приналежності, а іншим моделям навпаки потрібно мануально додавати поріг класифікації, щоб перейти від ймовірнісного виходу до лейбл класу;

2) регресія – це задача побудови функції перетворення вхідного простору у неперервну цільову змінну, на відміну від дискретних класів, що використовуються в класифікації. В комп'ютерному зорі регресія найчастіше з'являється у якості підзадачі – регресії координат точок інтересу або кутів обмежуючих коробок для задачі детекції об'єктів;

б) навчання без учителя – завдання машинного навчання, яке складається з виведення функції для опису прихованої структури з нерозмічених даних (отже, цільова змінна не включена до спостережень, як це було у задачах навчання з учителем);

в) навчання з підкріпленням – це тип машинного навчання, який дозволяє інтелектуальному агенту навчатися в інтерактивному середовищі шляхом реакції на результати своїх дій, використовуючи функцію підкріплення як зворотний зв'язок до виборів, які він здійснив.

У межах цієї роботи здебільшого розглядається навчання з учителем, а саме задача мультилейбельної ймовірнісної класифікації зображень.

Саме така задача встає під час аналізу медичних знімків томографії мозку.

Внутрішньочерепна геморагія – це кровотеча, яка виникає всередині черепа. Це серйозна загроза здоров'ю людини, яка вимагає швидкого та часто інтенсивного медикаментозного лікування. Наприклад, внутрішньочерепні крововиливи становлять приблизно 10% інсультів у США [5], де інсульт є п'ятою по кількості випадків причиною смерті [6]. Визначення місця та типу будь-яких наявних крововиливів є критичним кроком у лікуванні пацієнта.

Діагностика такої травми має бути терміновою процедурою. Коли у пацієнта виявляються гострі неврологічні симптоми, такі як сильний головний біль або втрата свідомості, висококваліфіковані фахівці перевіряють медичні знімки черепа

пацієнта, щоб виявити наявність, місце розташування та тип крововиливу. Процес складний і часто забирає багато часу.

Отже, створення автоматизованої системи, що може спростити та пришвидшити процес аналізу томографічних знімків – це актуальна задача в сфері медичної діагностики.

1.2 Огляд задачі аналізу томографічних зображень

Задача класифікації зображень – це задача відображення тривимірних тензорів, що відповідають зображенням, на множину деяких категорій. У випадку ймовірнісної мультитейбельної класифікації, наприклад, така задача формалізується як:

$$f: \{x \in R^{H \times W \times C}\} \rightarrow \{y \in R^n | 0 \leq y_i \leq 1, \forall i: 1 \leq i \leq n, i \in N\} \quad (1.1)$$

де x – вхідне зображення;

y – вихідний вектор, кожен елемент котрого відповідає ймовірності одного з класів;

H – висота зображення;

W – ширина зображення;

C – кількість каналів (або глибина) зображення (наприклад, для RGB кількість каналів дорівнює 3);

n – кількість класів.

Зазвичай кількість каналів зображення дорівнює 1 (для чорно-білих зображень) чи 3 (для RGB-зображень, де кількість каналів відповідає кольорам). Таким чином, модель машинного навчання, що класифікує звичайні кольорові зображення, насправді вивчає функцію наступного вигляду:

$$\begin{aligned}
 & f: \{x \in R^{H \times W \times 3} \mid \forall i, j, k: x_{ijk} \in [0, 255]\} \\
 & \rightarrow \{y \in R^n \mid 0 \leq y_i \leq 1, \forall i: 1 \leq i \leq n, i \in N\}
 \end{aligned}
 \tag{1.2}$$

де x – вхідне зображення;

y – вихідний вектор;

H – висота зображення;

W – ширина зображення;

n – кількість класів.

Але томографічні зображення мають іншу природу.

Комп'ютерна томографія (КТ) – це метод дослідження внутрішньої структури людського тіла за допомогою рентгенівського сканування, при якому пучок рентгенівського випромінювання проходить через тонкий шар тіла в різних напрямках. Вимірювання інтенсивності рентгенівського променя на виході з тіла при подальшій обробці дозволяє оцінити щільності тканин, що розташовані на шляху променів. Результати комп'ютерної томографії можуть бути представлені як набір 2D-зображень («перерізів» тіла) або як 3D-модель, що будується на основі цих зображень [7]. Сучасні комп'ютерні томографи зазвичай генерують від 64 до 640 зображень протягом однієї сесії томографічного сканування. Зображення найчастіше матиме розміри 512x512, 1024x1024 або 2048x2048.

Матриця зображення будується за допомогою спеціальної шкали – шкали Гаунсфілда. Ця шкала надає зручні кількісні значення для ступеня послаблення рентгенівського випромінювання структурами організму відносно дистильованої води. Таким чином нульовий показник шкали Гаунсфілда відповідає щільності води, від'ємні числа – тканинам менш щільним, ніж вода, додатні – більш щільним. Одиниця вимірювання – HU (Hounsfield unit, або одиниця Гаунсфілда).

Зазвичай, в тілі людини HU вимірюється в межах від -1024 до $+3071$, тобто усього 4096 рівнів послаблення [8] (якщо немає сторонніх тіл). Величину HU можна вирахувати за формулою:

$$HU = \frac{\mu_x - \mu_{\text{води}}}{\mu_{\text{води}} - \mu_{\text{повітря}}} \times 1000 \quad (1.3)$$

де μ_x – коефіцієнт послаблення матеріалу;

$\mu_{\text{води}}$ – коефіцієнт послаблення для води;

$\mu_{\text{повітря}}$ – коефіцієнт послаблення для повітря.

Для покращення розмежування органів один від одного, а також нормальних структур від патологічних, можуть використовуватися додаткові методики контрастного посилення. Спеціально введена контрастна речовина (часто йодовмісний контрастний препарат) дозволяє виділити обстежувану ділянку, зробити її більш «видимою» для КТ.

Приклади значень шкали Гаунсфілда [9] для різних речовин та тканин тіла (без застосування контрастного посилення):

- повітря: -1000 HU;
- легені: -600 HU;
- жир: від -120 до -90 HU;
- вода – за дефініцією рівно 0 HU;
- кров: від +30 до +45 HU;
- м'язи: від +10 до +40 HU;
- печінка: від +40 до +60 HU;
- кістки: від +700 (губчаста речовина) до +3000 (кісткова речовина) HU.

На жаль, при візуалізації знімків КТ не має сенсу приписувати весь діапазон шкали Гаунсфілда доступному діапазону яскравості (або чорно-білому спектру). Причиною цьому служить те, що звичайний монітор комп'ютера, який використовується на томографічному апараті, здатний відображати лише до 256 відтінків сірого кольору, а неозброєне людське око насправді здатне відрізнити між собою ще менше відтінків.

Тому у зв'язку зі значною шириною шкали Гаунсфілда і неможливістю відобразити весь її діапазон в одному спектрі, використовується програмний перерахунок сірого градієнта залежно від вікна – інтервалу шкали (рис. 1.1).

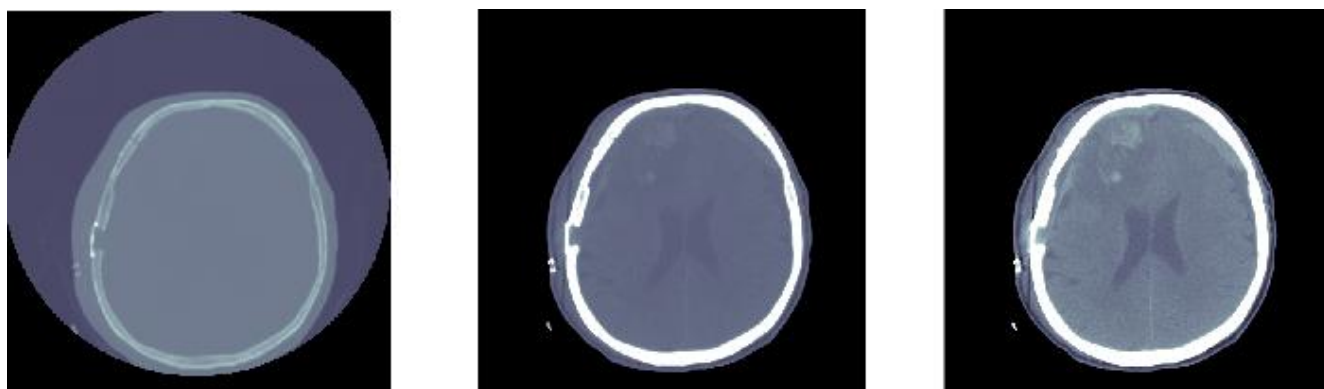


Рисунок 1.1 – Зображення одного перерізу з різними вікнами

Вікно може бути обране широке – тоді візуалізуються структури багатьох щільностей, проте неможливо розрізнити структури, близькі по щільності (як на другій частині рисунку 1.1). Інший варіант – вузьке вікно із заданими рівнем та шириною («легеневе вікно», «вікно для м'яких тканин», як на третьому зображенні рисунку 1.1, тощо). В цьому випадку втрачається інформація про структури, щільність яких виходить за межі обраного вікна, проте добре помітні структури, що є близькими по щільності. Іншими словами, зміну рівня вікна та його ширини можна порівняти зі зміною яскравості і контрастності зображення відповідно. Зменшення ширини вікна підвищує контраст зображення, зниження рівня вікна робить зображення більш яскравим, а підвищення рівня – більш темним. Зображення взагалі без вікна (перша частина рисунку 1.1) виглядає зовсім неконтрастно.

Через ці причини томографічні зображення найчастіше зберігаються у спеціальному форматі DICOM [10] (скорочено від The Digital Imaging and Communications in Medicine), що дозволяє зберігати разом метадані щодо томографічних знімків (наприклад, фізичну роздільність, глибину та орієнтацію знімку, вік та стать пацієнта тощо) та власне результати комп'ютерної томографії у вигляді бітмапів (бітових зображень без компресії), які зберігають шари зображення, репрезентуючи кожен піксель як число шістнадцятибітного цілочисленного типу [11].

Тож, модель машинного навчання, натренована класифікувати томографічні зображення, насправді вивчає іншу функцію:

$$f: \{x \in R^{H \times W \times C} \mid \forall i, j, k: x_{ijk} \in [-32768, 32767]\} \rightarrow \{y \in R^n \mid 0 \leq y_i \leq 1, \forall i: 1 \leq i \leq n, i \in N\} \quad (1.4)$$

де x – вхідне зображення;

y – вихідний вектор;

H – висота зображення;

W – ширина зображення;

C – кількість шарів томографії (зазвичай, велике число, близько 50-100);

n – кількість класів.

Отже, встає питання щодо того, як опрацьовувати такі зображення, адже звичайні двовимірні згорткові нейронні мережі націлені на роботу з даними іншого формату. Також, стає неможливою звичайна процедура переносу навчання (transfer learning), яка буде докладніше описана далі.

Ще однією важливою відмінністю обробки медичних зображень від звичайних задач комп'ютерного зору є велика складність в здобутті великих наборів даних [12]. Якщо в класичному комп'ютерному зорі існують набори даних широкого призначення розміром в кілька мільйонів анотованих зображень (навіть для складних для розмітки задач, таких як детекція об'єктів чи семантична й сутнісна сегментації), а також набори даних для вузьких задач розміром в сотні тисяч зображень, то в медичних задачах такі набори даних – це велика рідкість. Причина цього у тому, що анотацію медичних знімків може проводити тільки висококваліфікований спеціаліст – доктор відповідної спеціалізації. Отже, розмітка медичних знімків коштує дуже дорого та займає великі обсяги часу [13].

Через це є тенденція до спрощення типів задач у обробці медичних зображень. Наприклад, для діагностики травм голови та лікування геморагій мозку найкращим рішенням було б створити систему для тривимірної сутнісної сегментації (instance segmentation), де у тренувальному наборі даних кожний

крововитік розмічено як окрему одиницю, а також всі вокселі 3D-зображення віднесено до того чи іншого об'єкту або до фону. Але така анотація надто складна та потребує багато часу.

Одним із прикладів того, як подібну розмітку можна спростити, є зведення задачі сегментації до задачі класифікації. Адже для медиків при постановці діагнозу та призначенні лікування найважливішим є тільки те, в якій зоні внутрішньочерепного простору (конкретніше – між якими шарами мозку) з'явився крововилив. Тож, задачу пошуку геморагії зазвичай зводять до задачі класифікації зображення на кілька класів, що відповідають типам кровотечі (наприклад, епідуральні, субдуральні та інші).

Іншим прикладом спрощення розмітки є зведення задачі класифікації 3D-зображення, взятого з об'ємної комп'ютерної томографії, до класифікації низки 2D-зображень, кожне з котрих відповідає одному шару томографії (рис. 1.2).



Рисунок 1.2 – Двовимірний переріз, анотований для задачі семантичної сегментації

При такій формалізації задачі побудування та оцінювання моделі може зводитися до звичайної класифікації чорно-білих зображень, а результати

класифікації окремих зображень можна агрегувати у загальний результат оцінки томографічного знімку.

Обмеженням такого спрощення є те, що алгоритм має навчатися тільки на одній конкретній проекції знімку й не може бачити тривимірну картину (рис. 1.3).

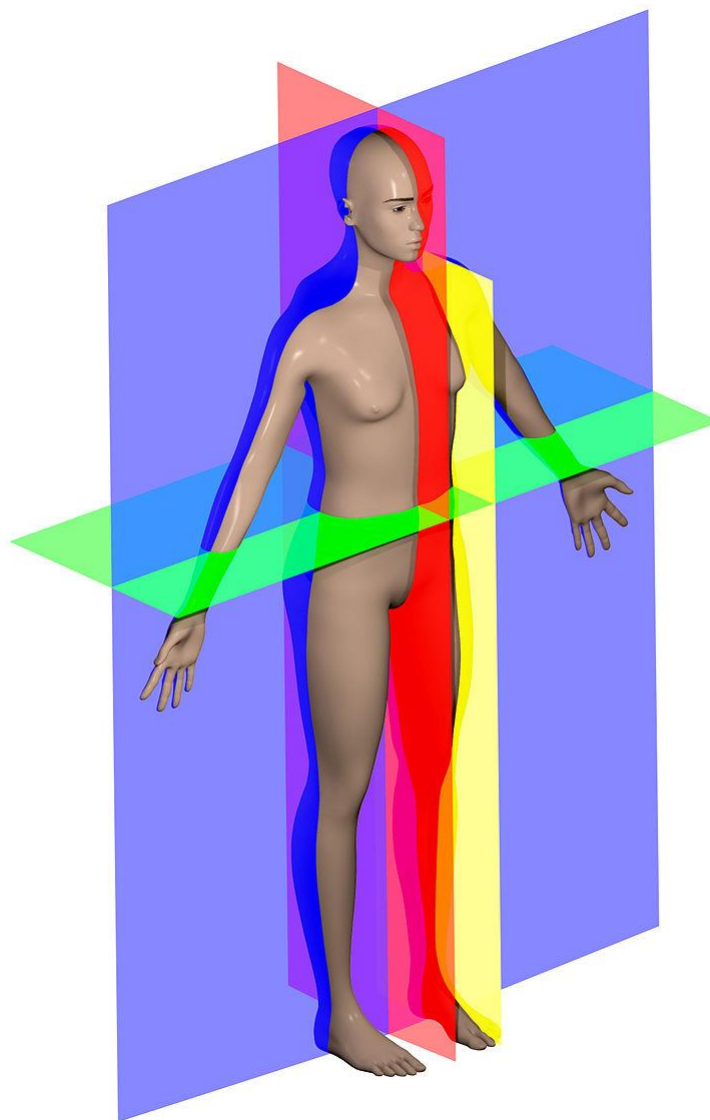


Рисунок 1.3 – Різні площини тіла: медіальна (червоним), парасагітальна (жовтим), фронтальна (синім) і аксіальна (зеленим) [14]

Для вирішення задачі пошуку геморагій на томографічних знімках мозку найчастіше використовується проекція саме на аксіальну площину тіла, що зображена на рисунку 1.3 зеленим кольором.

2 ПОСТАНОВКА ЗАДАЧІ

2.1 Загальні методи класифікації зображень

Через те, що нейронні мережі стали найпровіднішими алгоритмами в усіх задачах комп'ютерного зору, суттєво зсунувши межі того, що вважалося можливими, багато промислових застосувань машинного бачення починають утилізувати такий підхід.

В основному, подібний прорив став можливим завдяки роботі дослідницької групи Принстонського університету під керівництвом Фей-Фей Лі, яка створила набір даних ImageNet, що складається з кількох мільйонів зображень різноманітних об'єктів реального світу [15]. Існує кілька версій ImageNet, що використовувалися у різних змаганнях в рамках ILSVRC, але найчастіше у наукових роботах використовується варіант датасету, що анотовано під задачу класифікації зображень на тисячу категорій.

Цей набір даних уперше став достатньо великим, щоб продемонструвати увесь потенціал глибинного навчання та згорткових нейронних мереж.

Базова архітектура згорткової мережі складається з двох частин: згорткова та повнозв'язна [16].

Спочатку зображення подається на вхід саме до згорткової частини, яка також часто називається бекбоуном. Згорткова частина виконує роль вилучення ознак та фактично представляє собою комплексне нелінійне перетворення з простору зображень у простір ембедингів, в якому кожне зображення постає вектором фіксованої довжини. Особливість цього простору під час вирішення задачі класифікації у тому, що задача оптимізації ставиться таким чином, щоб у ньому цільові категорії були лінійно-роздільними, тобто виконувалося:

$$\exists w_1, \dots, w_n, k, \forall x \in X_0: \sum_{i=0}^n w_i x_i > k, \quad (2.1)$$

$$\forall x \in X_1: \sum_{i=0}^n w_i x_i < k$$

де X_0 – негативний клас;

X_1 – позитивний клас.

Вихідний шар цієї частини часто називають бутлнек-шаром або шаром «вузького місця»;

Друга частина згорткової мережі – це повнозв’язна голова, що безпосередньо виконує класифікацію зображення на основі ембедингів зі згорткової частини.

Класифікаційна частина зазвичай має досить просту структуру. В найпростішому випадку вона складається з шару батч-нормалізації [17] (пакетної нормалізації), шару дропауту [18] та повнозв’язного шару.

Після повнозв’язного шару йде функція активації, яка в залежності від задачі приймає вигляд софтмакс-активації для мультикласової класифікації чи сигмоїдальної активації для бінарної класифікації.

Мультилейбельна класифікація у випадку нейронної мережі зазвичай зводиться до бінарної класифікації по всім класам, тож на останньому шарі також використовується сигмоїда (формула 2.2).

$$\delta(x)_i = \frac{e^{x_i}}{1 + e^{x_i}} \quad (2.2)$$

де x – сума вхідних сигналів відповідного нейрону;

n – кількість класів;

i – номер класу.

Згорткова частина мережі складається зі згорткових блоків, які в свою чергу зазвичай складаються з кількох (2-8) згорткових шарів та шару субдискретизації на кінці [19].

Частіше за все згорткові шари реалізовані як операція кроскореляції з одиничним кроком та падінгом вхідного тензору нулями (це робиться для того,

щоб згортковий шар не зменшував просторову розмірність зображення). Найчастіше використовуються згортки 3×3 або більш складні варіації, такі як глибинна роздільна згортка [20].

Шар субдискретизації (pooling) зазвичай представляє собою операцію, яка з кожних чотирьох сусідніх пікселів зображення обирає лише один, який має максимальне значення активації з попереднього шару.

Зазвичай, кожен наступний згортковий блок зменшує просторові розміри вхідного тензора вдвічі, але збільшує глибину тензору (тобто кількість вихідних каналів) також вдвічі. Єдиним виключенням зазвичай є перший згортковий шар, який приймає на вхід похідне зображення з одним чи трьома каналами та генерує тензор з 8 чи 16 каналами.

Після кожного згорткового шару йде функція активації, призначення якої – додавати нелінійність до перетворення, яке вивчає нейронна мережа. Існує багато функцій активації, але найпопулярнішою в даний час є функція ReLU та сімейство споріднених функцій (Leaky ReLU та інші), а також гладкі функції, що мають графік, схожий на ReLU (такі, як swish-функція [21], наприклад).

Існує кілька модифікацій стандартної згорткової архітектури, які стали загальноновживаними та адаптовані у більшості найкращих рішень.

Першою такою модифікацією є «залишкові зв'язки» [22], які передають тензори з попередніх шарів на кілька кроків вперед, де ці тензори додаються до активацій поточного шару (рис. 2.1).

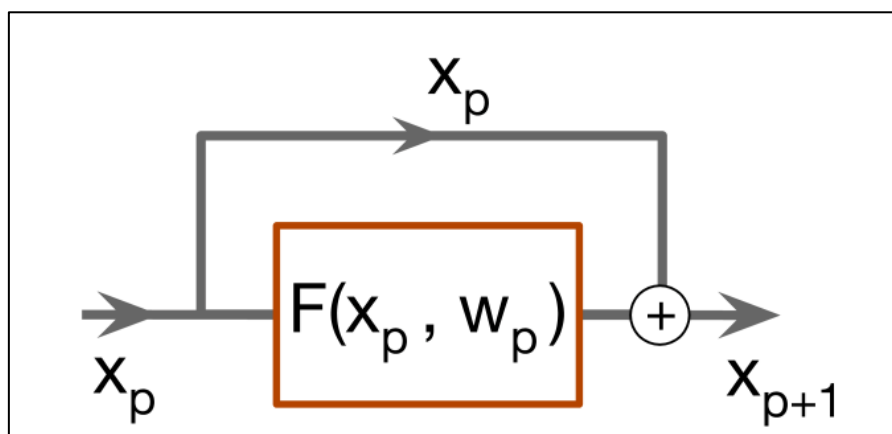


Рисунок 2.1 – Залишкові зв'язки у нейронній мережі

Наступний еволюційний крок «залишкових зв'язків» – це так звані блоки стискання-та-збудження [23] (squeeze-and-excitation). Фактично, це спроба застосувати механізм, схожий на механізм уваги (широко розповсюджений в прогнозуванні часових рядів та в обробці мови) до виходів згорткових шарів. Для цього попередній шар субдискретизується до розміру 1×1 глобальним усереднюючим пулінгом, проходить через повнозв'язний шар (який саме і вивчає так звану «увагу»), а результат цієї операції використовується як ваги, щоб переважити вихід початкового згорткового шару (рис. 2.2).

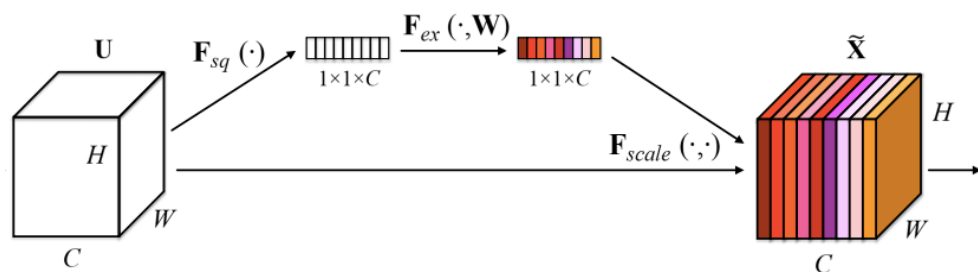


Рисунок 2.2 – Блок стискання-та-збудження

Останньою розробкою у галузі архітектур згорткових мереж є автоматичний пошук архітектур. Однією з найвідоміших розробок у цьому напрямку є EfficientNet – архітектура, розроблена корпорацією Google та представлена у нещодавньому дослідженні «EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks» [24]. Ця архітектура, знайдена за допомогою фреймворку AutoML (автоматичне машинне навчання) мобільного пошуку нейромережевих архітектур, має одне з найкращих співвідношень розміру мережі до точності на ImageNet, а також досить унікальну здатність масштабуватися від маленького розміру (EfficientNet-b0) до великого (EfficientNet-b7) з повільним нарощуванням кількості вагів мережі та її якості розпізнавання. Архітектура мережі EfficientNet-B0 зображена на рисунку 2.3.

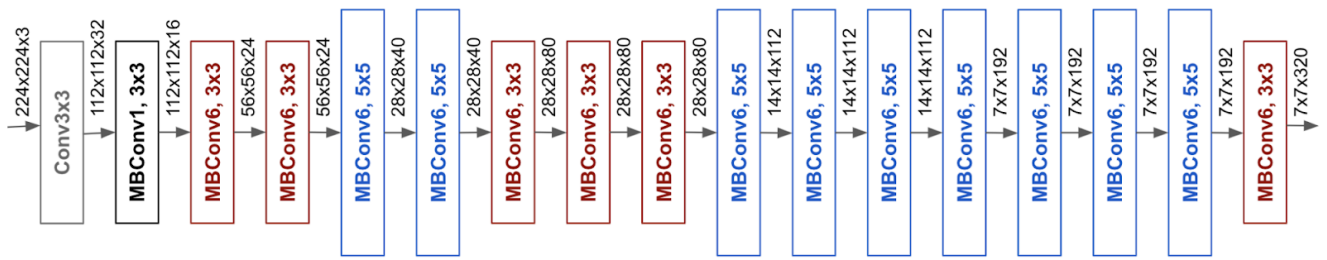


Рисунок 2.3 – Архітектура EfficientNet-B0

Всі ці архітектури можна використовувати для задачі класифікації томографічних знімків.

Для цього використовується підхід переносу навчання (transfer learning), суть якого у наступному [25].

Береться нейронна мережа, яку вже заздалегідь претреновано на достатньо великому наборі даних. Класифікаційна голова цієї мережі замінюється на нову, яка відповідає задачі, що вирішується. Після цього результуюча мережа навчається за звичайним алгоритмом. Таким чином, ваги мережі, що вже адаптовані під інший датасет, виступають точкою ініціалізації для відповідних шарів мережі при навчанні на новому наборі даних.

Перенос навчання пришвидшує сходимость мережі, а також у багатьох випадках призводить до більш високої фінальної точності моделі.

Чому це взагалі працює? Припустимо, що ми маємо згорткову нейронну мережу, що побудовано на наборі даних для класифікації меблів, та хочемо вирішити задачу класифікацію геморагій. Для переносу навчання ми беремо згорткову частину з претренованої мережі та випадково ініціалізуємо її повнозв'язну голову.

Кожний згортковий шар у згортковій частині вивчив якусь ознаку, яка сприяє класифікації меблів. Наприклад, згортки останнього шару перед ботлнеком можуть вивчати такі високорівневі ознаки, як текстура поверхні чи наявність ніжок у об'єкта в кадрі. Звичайно, ці ознаки мають мало спільного з ознаками, які можуть використовуватися для класифікації кровотеч. Але як Зейлер та Фергюс демонструють в статті «Visualizing and Understanding Convolutional Networks //

Computer Vision» [26], чим ближче до входу згортковий шар знаходиться в нейронній мережі, тим простіше ознаки, які він вивчає.

Тобто, будь ця мережа натренована на класифікацію меблів чи на класифікацію геморагій, найперші шари вивчать прості геометричні форми, такі як прямі лінії, криві, плями та інше (рис. 2.4). Звичайно, якщо ці ознаки вже є в мережі, натренованій на класифікацію стільців, наявність таких згорток пришвидшить навчання мережі під нове завдання.

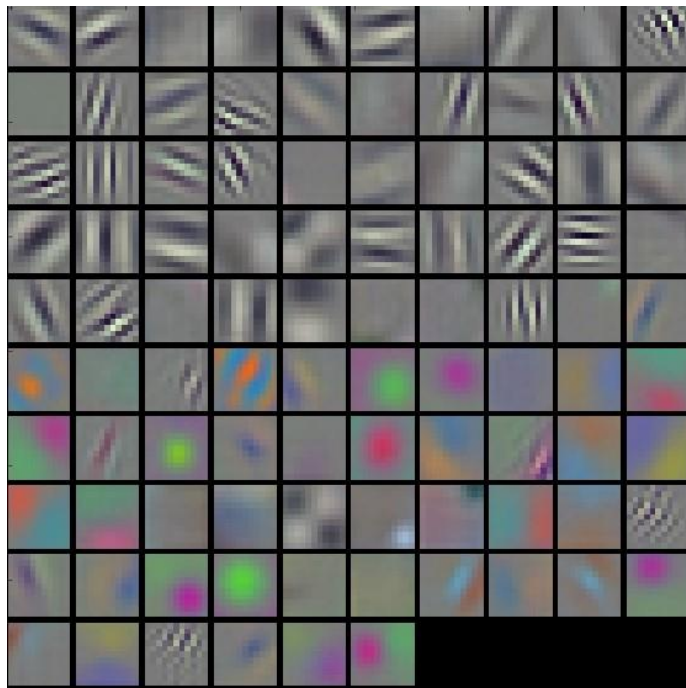


Рисунок 2.4 – Візуалізація ознак, вивчених першими шарами згорткової мережі, натренованої на ImageNet [26]

В якості набору даних для претренування зазвичай використовується саме ImageNet, тому що він достатньо великий, щоб нейронні мережі з великою кількістю вагів не були схильні до перенавчання, та в той самий час має як і велике різноманіття класів (від вимерлих динозаврів до марок авто), так і групи дуже схожих класів (наприклад, шістдесят порід собак), і через це згортки, які вивчаються на ньому, покривають велику кількість доменів.

Як демонструють роботи з аналізу медичних знімків, такі як «Analysis of the Human Protein Atlas Image Classification competition» [27], навіть для задач, домен

яких зовсім не схожий на природні зображення з ImageNet, перенос навчання дає істотній приріст з точності, і робити класифікацію томографічних знімків без використання претренованих вагів також недоцільно.

2.2 Методи класифікації томографічних зображень

Методи обробки томографічних зображень с використанням глибокого навчання вивчаються все досить давно. Але саме для задачі розпізнавання кровотечі мозку тільки в 2018 році було створено перші достатньо великі публічні набори даних для цієї задачі: Qure25k, що складається з 21905 сканів з комп'ютерної томографії, які анотовано з допомогою алгоритму автоматичної екстрації діагнозів з медичних карт пацієнтів, та CQ500, що складається з 491 зображення мозку, які були мануально проанотовані трьома незалежними експертами-радіологами [28].

На цих наборах даних було уперше протестовано алгоритми глибинного навчання для задач класифікації та сегментації геморагій. Для розпізнавання зображення подавалися на вхід до нейронної мережі зжатыми до розмірів 224x224. Оскільки претренована нейронна мережа потребує трьохканальний вхід (бо перенос навчання робиться з набору даних ImageNet, зображення в якому представлені в RGB), на вхід їй подавалися три різних вікна:

- стандартне вікно для тканин мозку: $w=80, l=40$;
- вікно для костей: $w=500, l=3000$;
- субдуральне : $w=175, l=50$.

Тут і далі рентгенологічні вікна позначаються двома параметрами:

- ширина вікна w позначає відстань у HU між пікселем зі значенням 0 (чорний колір) та значенням 1 (білий колір);
- рівень вікна l позначає щільність HU, що відповідає середині вікна.

Тобто, наприклад, стандартне вікно для мозку $w=80$, $l=40$ позначає, що всі значення томографічного знімку, що лежать нижче нуля, будуть відображені в чорний колір, а всі значення, що лежать вище 80, будуть відображені в білий колір. Значення між 0 та 80 (це як раз значення щільності більшості тканин мозку) будуть відповідати відтінкам сірого з лінійним переходом.

З таким препроцесінгом на наборах даних CQ500 та Qure25k для задачі виявлення геморагії авторами статті були отримані середні значення ROC-AUC, що дорівнюють 0.9419 та 0.9194 відповідно.

Інші цікаві результати демонструються в роботі «Practical Window Setting Optimization for Medical Image Deep Learning» [29]. По-перше, автори вводять поняття віконної функції. Класичні вікна, що використовуються радіологами для візуалізації, вони називають лінійними вікнами та задають наступною формулою:

$$F_{lin} = \min(\max(Wx + b, U), 0),$$

$$W = \frac{U}{WW},$$

$$b = -\frac{U}{WW} (WL - \frac{WW}{2})$$
(2.3)

де F_{lin} – функція лінійного вікна;

x – вхідне зображення;

W – ваги віконного шару;

b – вільний коефіцієнт віконного шару;

WW – ширина вікна;

WL – рівень вікна;

U – максимальне значення пікселя.

Максимальне допустиме значення пікселя, до якого робиться масштабування, залежить від репрезентації зображення. В наших експериментах U дорівнює одиниці, тому що зображення подається у float16-репрезентації.

Також, автори вводять поняття сигмоїдального вікна:

$$F_{sig} = \frac{U}{1 + e^{-(Wx+b)}},$$

$$W = \frac{2}{WW} \ln\left(\frac{U}{\epsilon} - 1\right),$$

$$b = -\frac{2WL}{WW} \ln\left(\frac{U}{\epsilon} - 1\right)$$
(2.4)

де F_{sig} – функція сигмоїдального вікна;

x – вхідне зображення;

W – ваги віконного шару;

b – вільний коефіцієнт віконного шару;

WW – ширина вікна;

WL – рівень вікна;

U – максимальне значення пікселя;

ϵ – «температура» сигмоїди, що відповідає за швидкість насичення.

Таким чином, графіки відображень, що задаються цими формулами, зображено на рисунку 2.5.

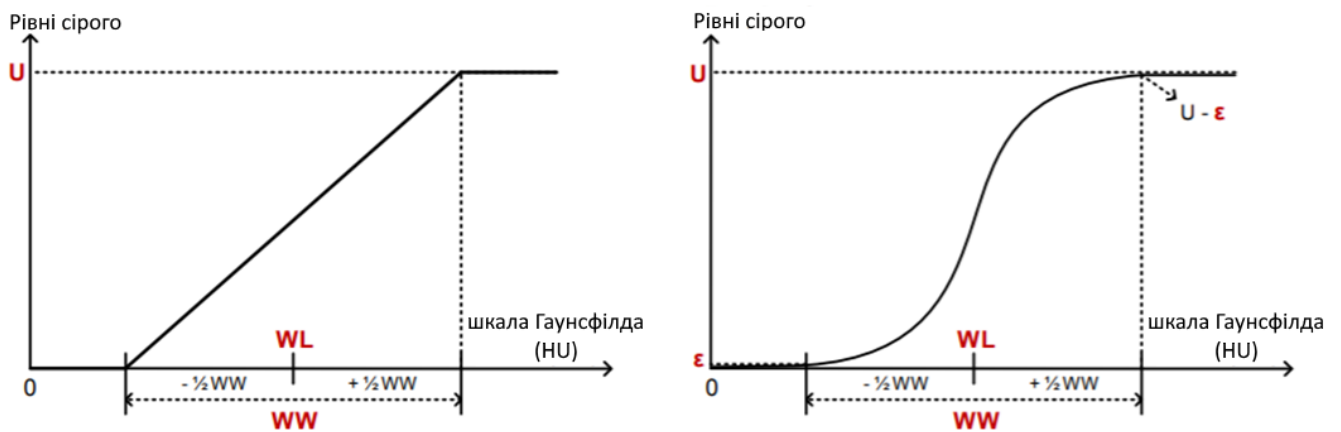


Рисунок 2.5 – Лінійне та сигмоїдальне вікна

Далі автори пропонують представити функцію перетворення вікна, що відображає чисті НУ в обмежений діапазон, на якому здатна навчатися нейронна мережа, у якості шару, що складається зі згортки 1x1 та функції активації –

жорсткої сигмоїди для лінійного вікна та звичайної сигмоїди для сигмоїдального вікна.

Таким чином, вікна перетворюються на диференційовані функції, які можуть навчатися сумісно з усією мережею алгоритмом зворотного поширення помилки.

2.3 Формалізація задачі

Для виконання роботи обрано набір даних з конкурсу RSNA Intracranial Hemorrhage Detection [30].

Чотири науково-дослідні установи надали великі обсяги анонімізованих томографічних досліджень, які було зібрано для створення єдиного набору даних. Американське товариство нейрорадіології (ASNR) кадр створило команду з більш ніж 60 докторів-добровольців, які анотували понад 25000 томографічних досліджень, що загалом склали близько 670000 зображень, розмічених на шість класів (рис. 2.6). Сервіс MD.ai надав інструменти та підтримку процесу анотації даних.

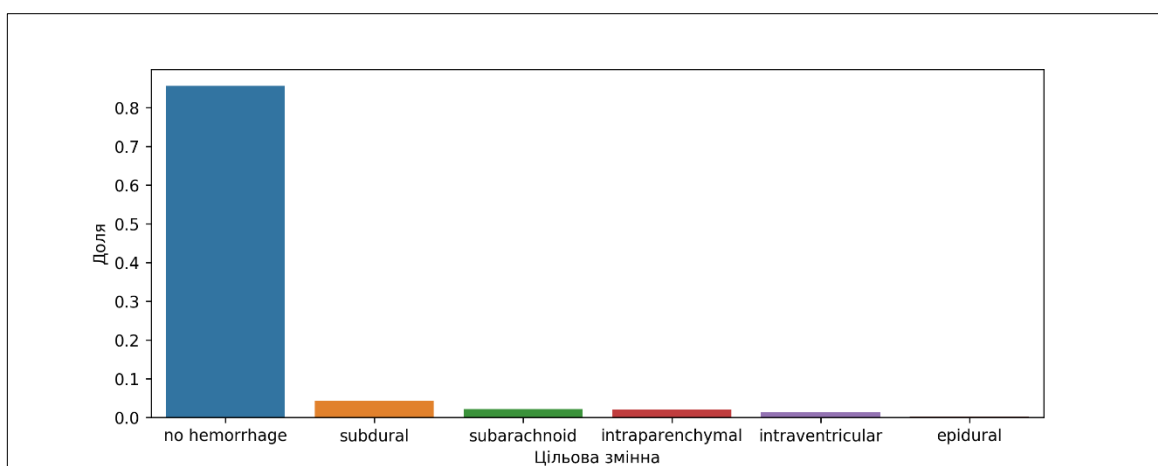


Рисунок 2.6 – Розподіл класів у наборі даних

Формальна постановка задачі на цьому наборі даних – мультилейбельна ймовірнісна класифікація геморагії мозку на шість класів: наявність геморагії, субдуральна, субарахноїдальна, інтрапаренхіматозна та інтравентрикулярна геморагії.

Внутрішньочерепна геморагія – відносно поширений стан, який має безліч причин, починаючи від травми, інсульту, аневризми, хвороб судин, високого артеріального тиску, наркотиків та порушень згортання крові. Неврологічні наслідки також сильно різняться залежно від розміру, типу крововиливу та місця розташування, починаючи від головного болю до смерті. Роль рентгенолога полягає у виявленні крововиливу, характеристиці підтипу крововиливу, його розмірах та визначенні, чи може крововилив ставити під загрозу критичні ділянки мозку, які можуть вимагати негайної операції.

Хоча всі гострі (тобто нові) крововиливи на комп'ютерній томографії (КТ) виявляються щільними (тобто білими), основними ознаками візуалізації, які допомагають радіологам визначити підтип крововиливу, є розташування, форма та близькість до інших структур.

Інтрапаренхіматозний крововилив – це геморагія, яка розташована повністю всередині самого мозку.

Внутрішньошлуночковий (інтравентрикулярний) та субарахноїдальний крововиливи – це кров, яка просочилася в простори мозку, які в нормі містять спинномозкову рідину (шлуночки або субарахноїдальні цистерни).

Екстрааксіальні крововиливи – це кров, яка збирається в тканинних покривах, що оточують мозок (наприклад, субдуральні або епідуральні підтипи). Пацієнти можуть демонструвати більше одного типу крововиливів у мозок, які можуть з'являтися на одному зображенні. Незважаючи на те, що невеликі крововиливи менш небезпечні, ніж великі крововиливи, як правило, навіть невеликі крововиливи можуть призвести до смерті, оскільки зазвичай це симптом іншого типу серйозних порушень (наприклад, аневризми).

2.4 Метрика якості моделі

Для оцінки задачі в якості метрики буде використовуватися зважена кросентропія:

$$\logloss = \sum_{i=0}^n \sum_{j=0}^m w_j (y_{ij} p_{clip_{ij}} + (1 - y_{ij})(1 - p_{clip_{ij}})), \quad (2.6)$$

$$p_{clip_{ij}} = \max(\min(p_{ij}, 1 - 10^{-15}), 10^{-15})$$

де m – кількість класів;

n – кількість зображень;

w – ваги класів (дорівнюють 1 для кожного типу геморагії та 2 для класу наявності крововиливу в цілому);

p – ймовірність класу на виході моделі.

Кросентропія у якості метрики досить сильно карає модель за надто впевнені невірні результати. Через це вона часто використовується в задачах класифікації медичних зображень, бо, на відміну від таких метрик як точність та повнота, наприклад, демонструє якість ймовірнісного виходу. Тож, впевненість моделі з низькою кросентропією можна вважати достовірною. Через це саме ця метрика (яку також іноді називають «негативним логлоссом») популярна у галузях, де результати використання моделі мають бути інтерпретованими, і медична сфера якраз відноситься до таких галузей.

3 МЕТОДИКА ДОСЛІДЖЕНЬ

3.1 Архітектури згорткової частини мережі

Як вже було зазначено, згорткові мережі складаються з двох половин. Конфігурація власне класифікаційної половини в усіх експериментах буде однаковою: адаптивний конкатенаційний шар дискретизації (конкатенація максимальної та середньої субдискретизацій розміром $N \times N$), батч-нормалізація, шар виключення (дропаут) та повноз'єднаний шар з шістьма вихідними нейронами з сигмоїдальною активацією. Рівень дропауту буде налаштовуватися окремо для різних експериментів.

Конфігурація згорткової частини мережі впливає на результати значно більше. Як зазначено вище, зазвичай згорткові частини мережі розробляються та тестуються на датасеті ImageNet (приклад компромісу між точністю та розміром мережі зі статті [24] наведено на рисунку 3.1).

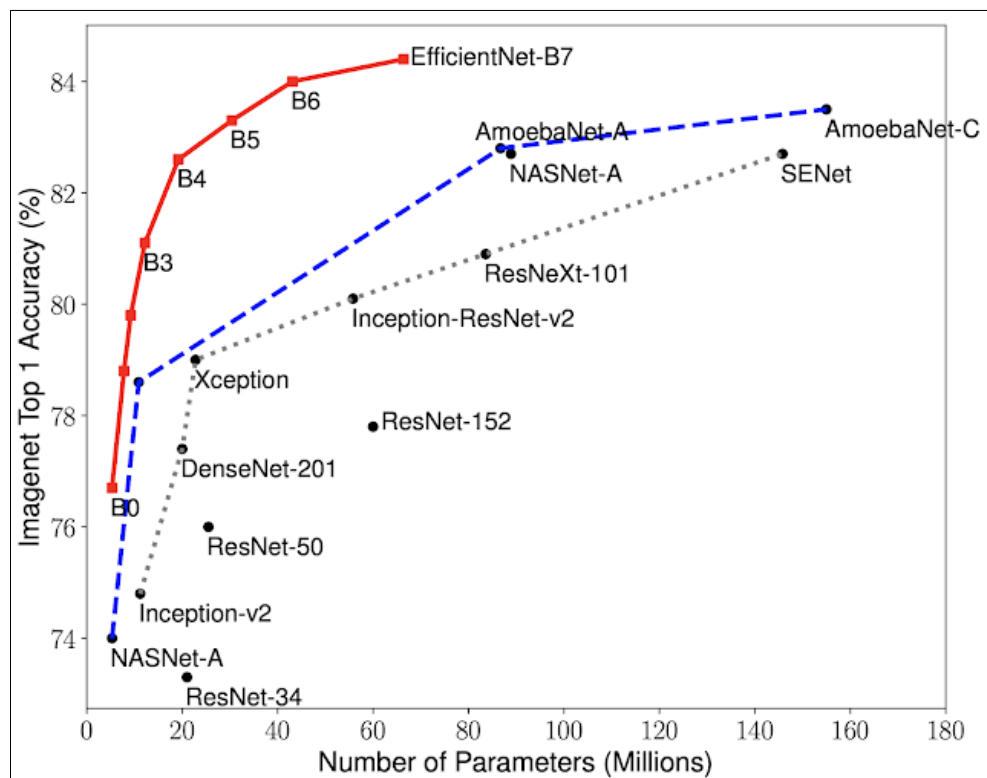


Рисунок 3.1 – Розмір та точність різних мереж за результатами ImageNet

Існують архітектури, створені спеціально для конкретних задач на інших датасетах (наприклад, архітектура GapNet [31] створена спеціально для задач аналізу медичних досліджень), але частіше за все такі архітектури не використовуються за межами окремих задач (тобто, вони надто вузько заточені) та мають результати, кращі за навчання з нуля, без претренованих вагів, але гірше за навчання з transfer learning.

Загалом є дві основні кореляції.

По-перше, чим більша мережа (тобто, чим більше в неї вагів), тим більшу точність вона демонструє у порівнянні з іншими архітектурами того ж сімейства. Цю залежність найкраще видно на рисунку вище на прикладі родини мереж EfficientNet, де послідовне збільшення згорткових блоків призводить до збільшення мережі та нелінійного підвищення точності. Але ця залежність може бути дещо оманливою: треба мати на увазі, що на ImageNet збільшення числа параметрів працює на користь моделі через те, що ImageNet складається з мільйонів зображень, тож на ньому майже неможливий сильний оверфітинг (перенавчання моделі). Але на наборах даних з меншою кількістю даних чи з більшим рівнем шуму збільшення мережі може призводити до погіршення навчання.

По-друге, взагалі використання мережі з більш високою точністю на ImageNet на іншій задачі частіше за все також призводить до більш високої точності на ній. Наприклад, використання EfficientNet як енкодера для детекції об'єктів дозволяє досягнути більш високих результатів, ніж використання якогось класичного бекбоуну, такого як ResNet-50, наприклад [32]. Але і ця залежність не завжди зберігається, через те, що деякі архітектури (особливо архітектури, що побудовано з використанням нейронного пошуку архітектур) можуть бути надто заточені саме під ImageNet та погано працювати для задач з інших галузей.

Задля перевірки цих двох гіпотез на задачі класифікації томографічних знімків, було перевірено наступні бекбоуни:

- SE-ResNeXt-50_32x4d (79.1% top-1 точність, 27.5М параметрів) [33];
- SE-ResNext-101_32x4d (80.2% top-1 точність, 48.8М параметрів) [33];
- EfficientNet-B3 (81.1% top-1 точність, 12.0М параметрів) [24];

– EfficientNet-B5 (83.3% top-1 точність, 30.0М параметрів) [24].

Архітектури порівняно на пайплайн навчання, конкретні параметри якого будуть детальніше розглянуті в наступному підрозділі.

Для вибору архітектур, які прийматимуть участь у подальших експериментах, скористаємося методом головного критерію. В якості критеріїв візьмемо:

- кросентропія мережі на датасеті RSNA (головний критерій для оптимізації при прийнятті рішення);
- час навчання до сходження (обмеження – не більше 16 годин).

За цими критеріями буде обрано дві найбільш придатні архітектури для подальших досліджень.

3.2 Налаштування гіперпараметрів

Нейронні мережі мають цілу низку гіперпараметрів, налаштування котрих відіграє важливу роль в отриманні високої якості розпізнавання. Перелічимо основні параметри, які будуть налаштовуватися в ході роботи, та спосіб їх вибору або експерименти з їх налаштування, які планується проводити.

Перший параметр – це темп навчання (learning rate). Цей параметр відповідає за масштабування часткових похідних функції витрат під час зворотного поширення градієнту. Темп навчання зазвичай налаштовується сумісно з розміром батчу за формулою:

$$lr_{new} = \sqrt{\frac{bs_{new}}{bs_{old}}} * lr_{old} \quad (3.1)$$

де lr_{new} – новий темп навчання (після зміни розміру батча);

lr_{old} – старий темп навчання;

bs_{new} – новий розмір батча;

bs_{old} – старий розмір батча.

Під час наших експериментів це дозволило масштабувати темп навчання при перенесенні коду між різними GPU або при переході з обчислень в 32-бітному представленні мережі до 16-бітного.

Для обрання початкового значення темпу навчання та його скедулінгу використовуються прийоми з роботи «Cyclical Learning Rates for Training Neural Networks» [33]. А саме, використовується циклічний темп навчання. Для визначення стартової верхньої границі циклу використано тест меж темпу навчання, також відомий як learning rate finder. Суть методу складається у наступному: нейронну мережу навчають протягом двох-восьми епох (зважаючи на розмір набору даних), поступово збільшуючи темп навчання з заздалегідь обраного замалого значення (наприклад, 10^{-7}) до надто великого значення (наприклад, 1) за логарифмічною шкалою. Протягом цього навчання записується функція втрат на тренувальному наборі даних. Якщо побудувати графік функції втрат від темпу навчання, отримаємо увігнуту криву (рис. 3.2).

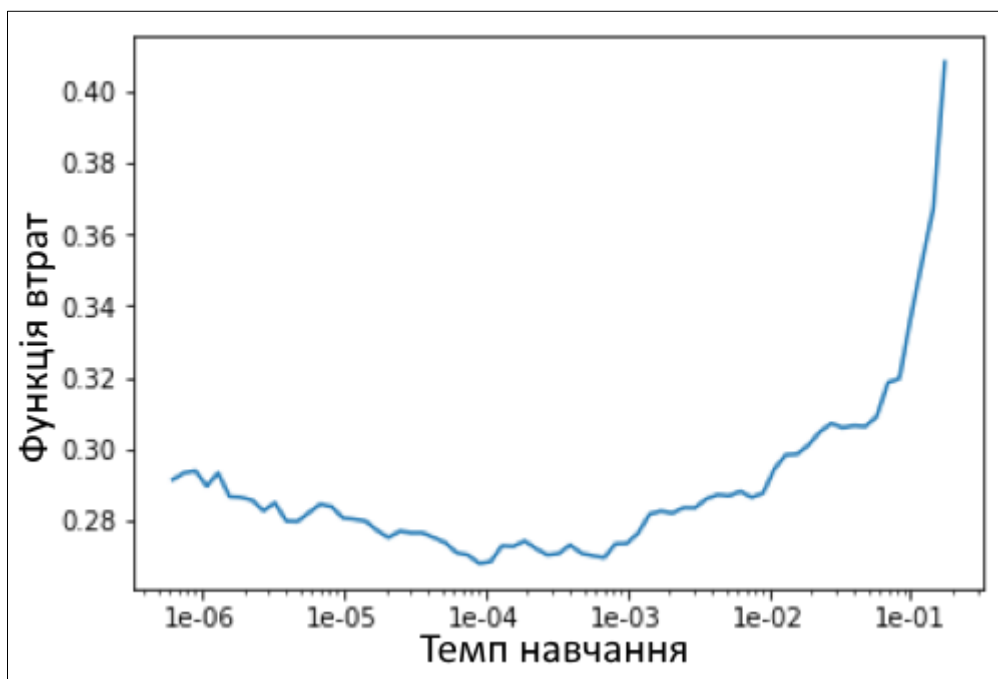


Рисунок 3.2 – Пошук оптимального темпу навчання

Бачимо, що функція втрат досягає глобального мінімуму в районі темпу навчання близько 10^{-3} . Після цього мережа починає розходитися, отже якщо довго навчати мережу з таким темпом навчання, функція витрат почне коливатися. Тож оптимальним темпом навчання для мережі є значення десь в 5-10 разів менше (в нашому прикладі таким значенням може бути $2 \cdot 10^{-4}$).

Саме цей алгоритм буде використовуватися під час усіх експериментів для мануального вибору оптимальної швидкості навчання (верхньої межі циклу у випадку циклічного темпу навчання).

Базовий скедулінг для експериментів з архітектурами складається з трьох стадій.

Спочатку протягом двох епох навчається тільки голова мережі з константним темпом навчання, знайденим з використанням learning rate finder. Це призначено для того, щоб знизити помилку мережі та адаптувати ваги голови, що спочатку ініційовані випадково, до активацій ботлнек-шару згорткової частини мережі.

На другій стадії мережа навчається один цикл циклічного темпу навчання на розмірі зображень 256×256 пікселів. Навчання на зменшеному розмірі зображень (так зване прогресивне навчання) використовується для прискорення пайплайну. Темп навчання також обирається з використанням learning rate finder.

На третій стадії мережа навчається протягом одного циклу на зображеннях 512×512 пікселів, що відповідає оригінальному розміру зображень в dicom-файлах. При цьому темп навчання (його верхня межа у випадку циклічного темпу навчання) виставляється вдвічі меншим за відповідний показник під час першого циклу.

Таким чином, ми проводимо оптимальний перенос навчання та тренуємо нашу мережу до сходження протягом 10 епох. З кожного циклу обирається епоха з найкращою кросентропією на валідаційній частині набору даних для подальшого використання.

Темп навчання змінюється за трикутним скедулінгом згідно статті [34]. Графік темпу навчання зображено на рисунку 3.3.

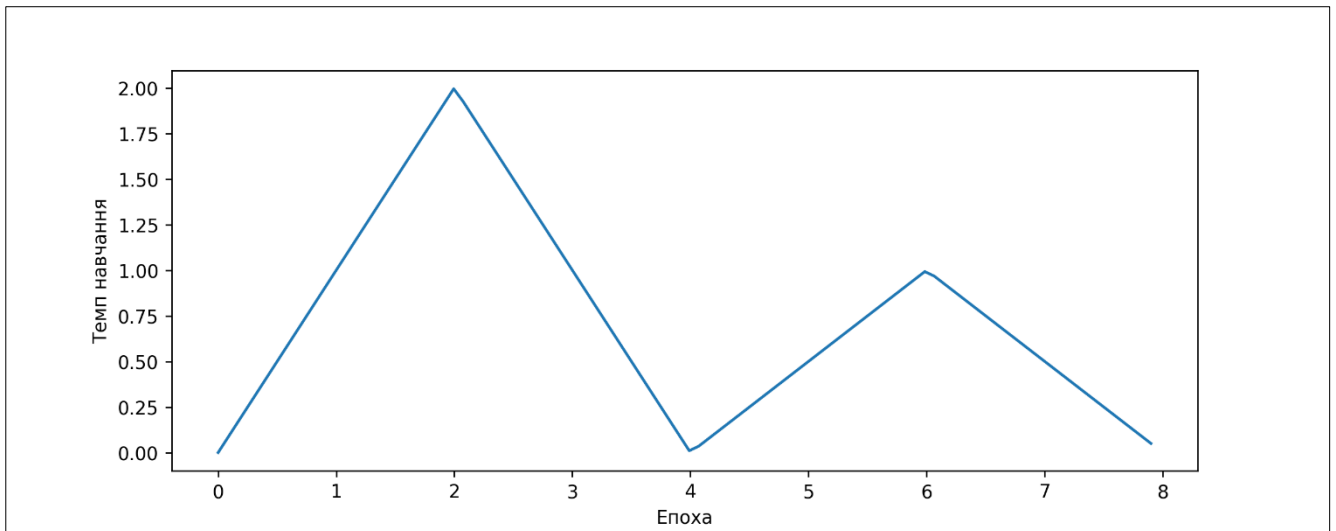


Рисунок 3.3 – Темп навчання за трикутною схемою

Після обрання архітектур для подальшого тестування також перевірено використання косинусоїдальних циклів згідно зі статтею «SGDR: Stochastic Gradient Descent With Warm Restarts» [35]. Графік темпу навчання зображено на рисунку 3.4.

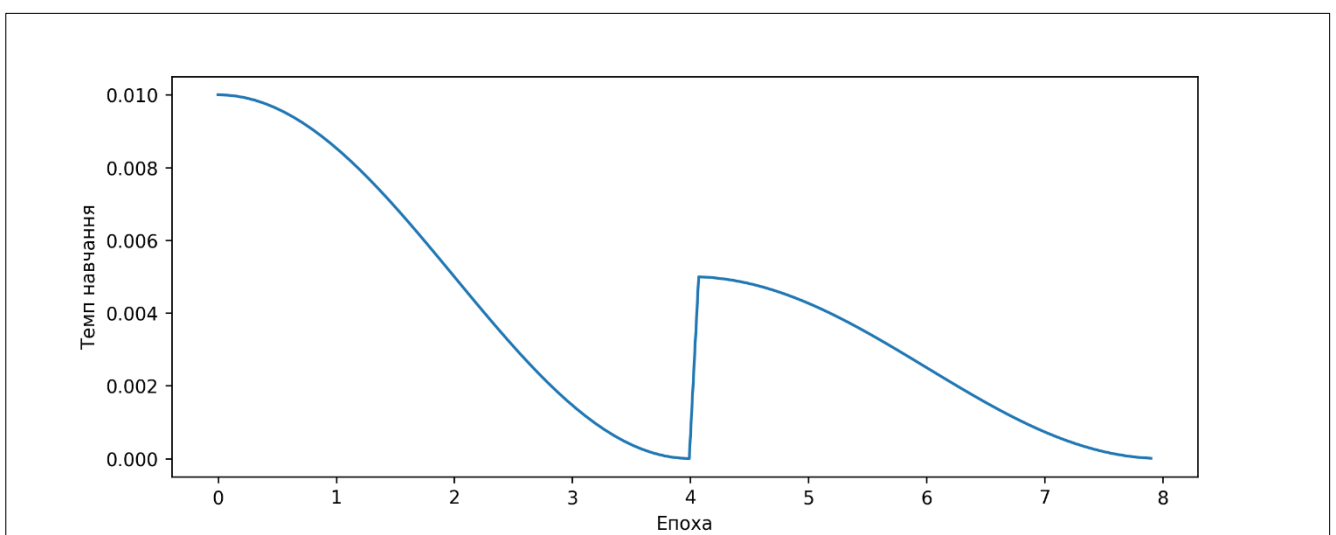


Рисунок 3.4 – Темп навчання за косинусоїдальною схемою

В якості оптимізатору використано найсучаснішу модифікацію алгоритму градієнтного спуску – RangerLars, що поєднує в собі ректифіковану адаптивну оцінку моментів (в оригіналі – Radam [36]) з евристикою LoodAhead [37],

оптимізовану для обробки великих батчів даних з допомогою алгоритму пошарового адаптивного масштабування темпу навчання (в оригіналі – LARS [38]).

Важливою проблемою майже всіх алгоритмів машинного навчання (й особливо нейронних мереж) є оверфітінг, також відомий як перенавчання. Перенавчена нейронна мережа демонструє високу точність (у нашому випадку – низьку кросентропію) на тренувальному наборі даних, але значно нижчу точність та більшу дисперсію передбачень на тестовому наборі.

Перенавчення виникає тоді, коли структура мережі надто складна та дозволяє «запам'ятати» елементи навчальної вибірки, підлагодивши ваги спеціально для кожного зображення таким чином, щоб для нього мережа видавала вірну мітку класу.

Існує багато механізмів боротьби з перенавчанням. До них відносяться і типовий механізм регуляризації, що використовувався ще у алгоритмі лінійної регресії – лассо- та рідж-регуляризації, і шари дропауту та батч-нормалізації. Також, для алгоритмів комп'ютерного зору існує ще один, досить специфічний спосіб регуляризувати модель – так звані аугментації.

Аугментації – це процес довільного спотворення зображень, що подаються на вхід до мережі, з метою підвищення різноманіття вхідних даних. Ідея аугментацій для регуляризації мережі полягає в наступному: одне й те ж томографічне зображення мозку могло бути зроблене дещо по-іншому, але мітка класу від цього не може змінитися.

Аугментації використовуються під час навчання моделі для збільшення набору даних. Для цього зображення трансформуються в онлайні під час генерації батчу перед поданням до мережі.

Також, аугментації можна використовувати під час інференсу мережі. Цей алгоритм називається *test-time augmentations*. Суть його полягає в тому, щоб зробити прямий прохід нейронної мережі на оригінальному зображенні, а також на кількох його трансформованих версіях. Після цього результати передбачення усереднюються. В нашому пайплайні використовується вісім трансформованих

версій: кожна з них відповідає одному з можливих поворотів з групи симетрії D4. Результати передбачень усереднюються на рівні логітів геометричним середнім.

В якості прикладу візьмемо томографічний знімок з інтрапаренхімальною кровотечею (рис. 3.5).

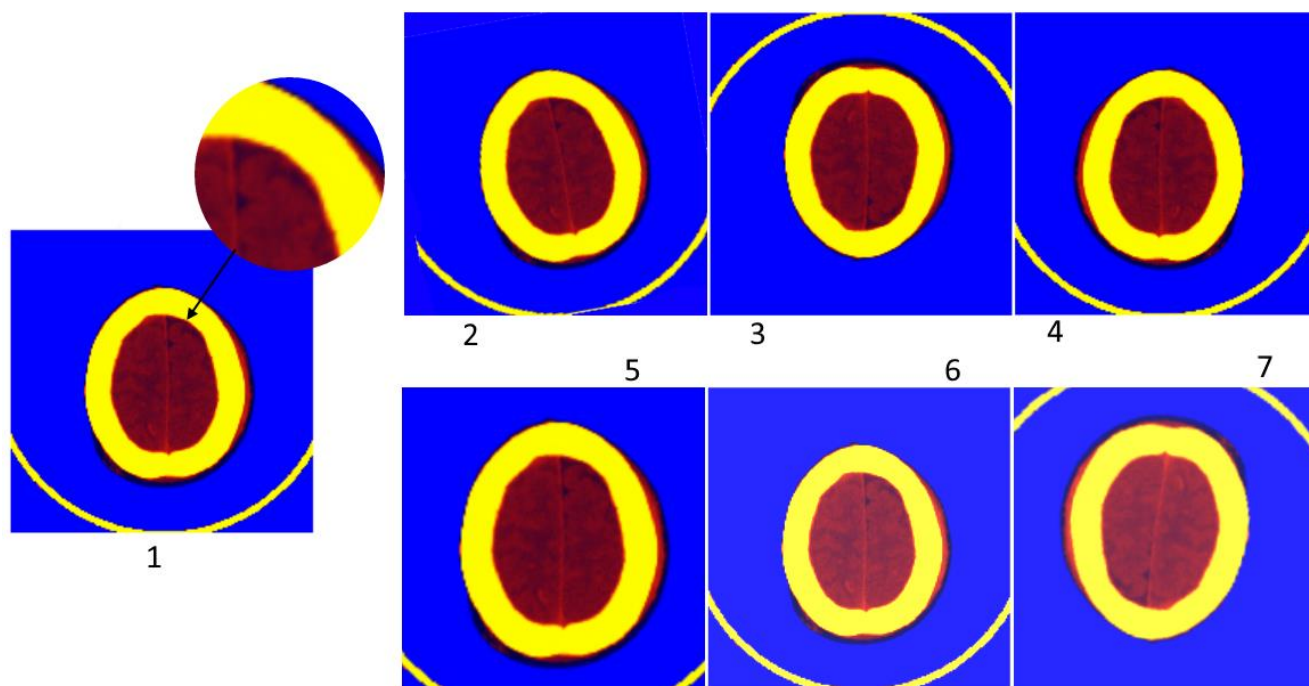


Рисунок 3.5 – Приклад аугментацій

На цьому рисунку під номером 1 зображено оригінал знімку. На збільшеній версії видно власне кровотечу – вона виглядає як більш темна зона під паренхімальним покровом. Саме вона задає клас зображення. Отже, будь-яке перетворення, яке залишає цю пляму розпізнаваною, є валідною аугментацією. В нашому пайплайні використано наступні аугментації, що зображені на рисунку 3.5:

- випадкові повороти до 15° (номер 2);
- вертикальне віддзеркалення (номер 3);
- горизонтальне віддзеркалення (номер 4);
- збільшення до 20% (номер 5);
- зміни яскравості до 20% (номер 6).

Кожне з афінних перетворень (повороти, віддзеркалення, збільшення) має 75% ймовірності використання. Зміни яскравості використовуються для 80%

зображень. Приклад кумулятивного використання всіх аугментацій відразу намальовано на сьомому прикладі на зображенні.

3.3 Налаштування вікон

Вдалий вибір вікна зображення є ключовим для розпізнавання аномалій на томографічних знімках. Існує кілька підходів для роботи з вікнами, які потрібно протестувати. Також, у цій роботі ми пропонуємо новітній підхід для правильного вибору вікон з урахуванням переносу навчання.

Здається можливим навчити мережу зовсім без вікон, адже насправді немає ніяких обмежень на розподіли вхідних даних для нейромережі. Однак, по-перше, великий інтервал вхідних даних (приблизно від -2000 до 3000 HU) призводить до великої погрішності обчислень з плаваючою точкою, а по-друге, зображення без вікон виходять зовсім не контрастними, тому що інтервал, який відповідає мозку, складає лише 2-3% від загального інтервалу (рис. 3.6).

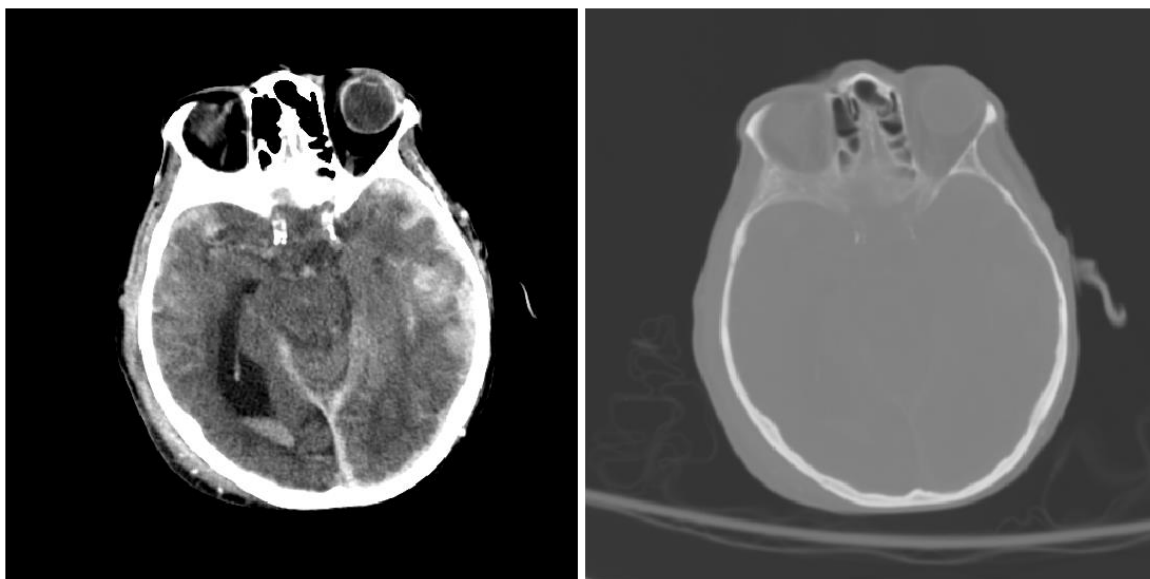


Рисунок 3.6 – Порівняння стандартного вікна та повного зображення

Базові експерименти з архітектурами та з налаштуванням гіперпараметрів буде виконано на вікнах з центром $l=40$ та шириною $w=80$. Таке вікно відповідає сірій та білій речовині мозку та найчастіше використовується радіологами для візуального дослідження томограми мозку.

Після того, як у експериментах з архітектурами та у експериментах з налаштуванням гіперпараметрів було обрано дві найкращі архітектури, для однієї з них буде досліджено наступні підходи:

- стандартне вікно з $w=80$, $l=40$;
- більш широке вікно з $w=256$, $l=80$;
- три вікна: $w=80$, $l=40$, $w=128$, $l=60$ та $w=256$, $l=80$;
- автоматична оптимізація вікна (WSO) за статтею «Practical Window Setting Optimization for Medical Image Deep Learning» [29] з випадковою ініціалізацією вікон;
- WSO з вікнами, мануально ініційованими в значення $w=80$, $l=40$, $w=128$, $l=60$ та $w=256$, $l=80$;
- модифікована оптимізація вікна, що бере до уваги перенос навчання.

Стандартне вікно використовується у якості бейзлайну. Вікно з шириною 256 береться як приклад того, що у мережу можна надавати більше інформації (ціною зменшення контрастності), задаючи більш широке вікно (рис. 3.7).

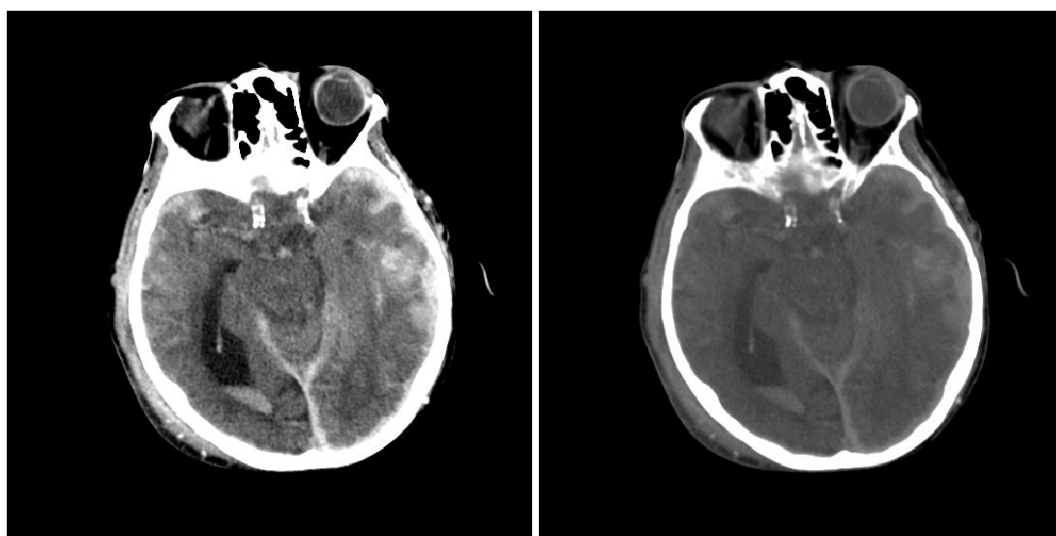


Рисунок 3.7 – Порівняння стандартного вікна та широкого

Саме ширина 256 здається оптимальною, тому що таким чином зберігається оригінальна роздільність пікселів (адже зображення після формування вікон зберігаються в восьмибітному формат перед початком навчання мережі), і не потрібно робити додаткову інтерполяцію, яка дещо спотворює зображення.

Використання трьох вікон – це спосіб утилізувати той факт, що претренована мережа з вагами з ImageNet має три окремих канали з різними наборами параметрів у першій згортці. В якості трьох вікон використовуються стандартне вікно, яке має високий контраст, але зберігає мало інформації, та два більш широких (та, як результат, менш контрастних) вікна (рис. 3.8).

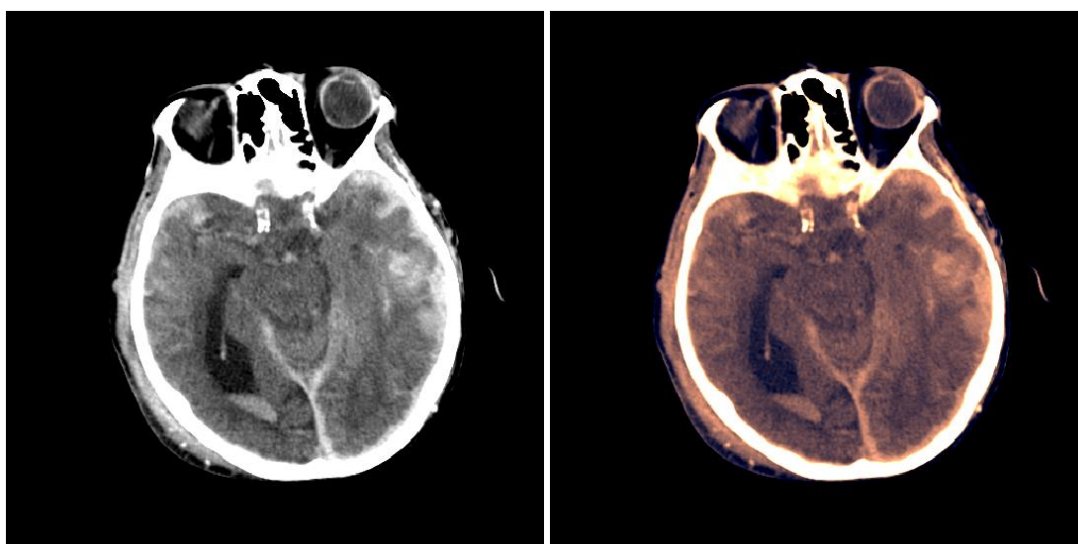


Рисунок 3.8 – Порівняння стандартного вікна та трьох вікон

Бачимо, що з трьома вікнами зберігається більше деталей (а також, деякі зони, що були повністю білими або чорними, добувають окрасу через ширші границі вікон – верхня змінюється з 80 до 208, а нижня – з 0 до -48).

Автоматична оптимізація вікон (від оригінального WSO – window setting optimization) має працювати краще вікон, що підібрані власноруч, тому що вона надає можливість більш тонко налаштувати набір вікон, що передає всю необхідну інформацію. Крім того, при автоматичній оптимізації вікон автори статті пропонують використовувати не лінійні, а сигмоїдальні вікна (рис. 3.9).

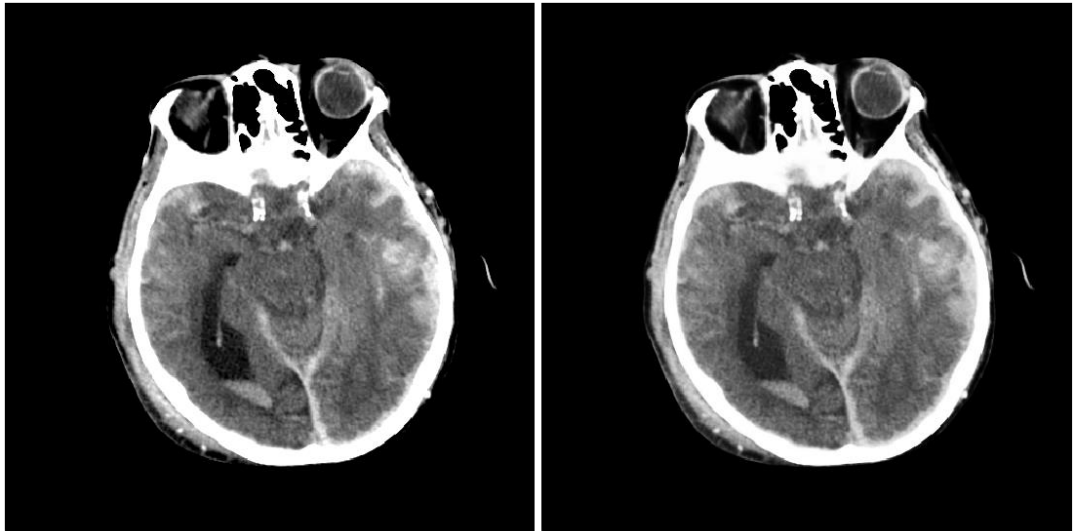


Рисунок 3.9 – Порівняння лінійного та сигмоїдального вікна

З рисунка бачимо, що візуально лінійне та сигмоїдальне вікна для одного й того ж зображення та налаштування вікна (в даному випадку $w=80$, $l=40$) майже не відрізняються, тому в попередніх експериментах для спрощення використовуються лінійні вікна.

Але у модулі автоматичної оптимізації використання сигмоїдальних вікон покращує градієнти (тому що сигмоїда має гладку похідну та диференційована всюди, на відміну від порогової функції, що використовується в лінійному вікні) та призводить до більш високих результатів, згідно з оригінальною статтею.

Недоліком цього алгоритму є те, що автори оригінальної статті експериментують з мережею, що тренується з нуля. Однак, коли модуль автоматичної оптимізації вікон додається до вже претренованої мережі, виникає проблема того, що цей модуль на початку свого навчання видає активації з розподілу, що не відповідає розподілу, на якому мережа навчалася (тому що шари цього модулю ініційовані випадково).

Задля вирішення цієї проблеми пропонується підхід нормалізації вікна (N-WSO) – штучної зміни розподілу вихідних активацій блоку автоматичної оптимізації вікна таким чином, щоб він максимально нагадував ImageNet.

Мережа, що претренована під ImageNet, вивчає наступний вхідний розподіл зображень по каналам:

$$\begin{aligned}
 X_{red} &\sim \mathcal{N}(0.456, 0.224^2) \\
 X_{green} &\sim \mathcal{N}(0.485, 0.229^2) \\
 X_{blue} &\sim \mathcal{N}(0.406, 0.225^2)
 \end{aligned}
 \tag{3.2}$$

де X_{red} – канал красного кольору (перший канал),

X_{green} – канал зеленого кольору (другий канал),

X_{blue} – канал блакитного кольору (третій канал).

Тож, для зберігання та максимально ефективного використання параметрів, що вивчено на ImageNet, потрібно, щоб шар автоматичної оптимізації вікон при ініціалізації видавав результати вікон, нормалізовані в цьому діапазоні. Для цього використовується наступна схема блоку (рис. 3.10).

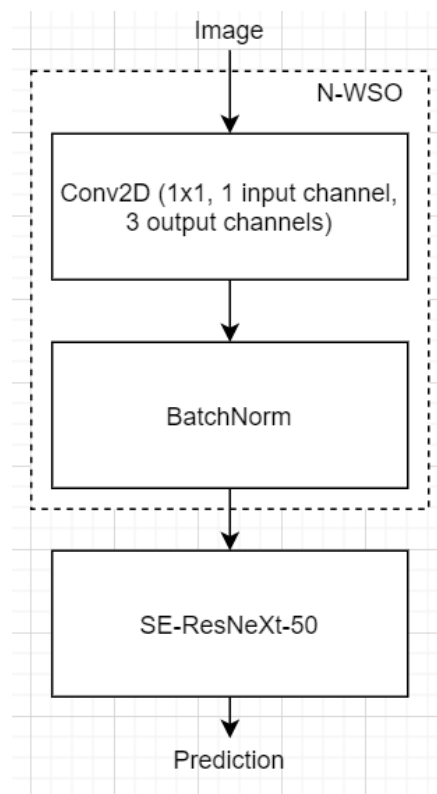


Рисунок 3.10 – Схема блоку N-WSO

Ваги двовимірної згортки ініціалізуються згідно з формулою 2.4. Для ініціалізації використовуються ті ж самі вікна, що і у експерименті з трьома вікнами: $w=80, l=40, w=128, l=60$ та $w=256, l=80$.

Шар батч-нормалізації ініціалізується наступним чином: параметр γ для кожного каналу дорівнює $1/\sigma$ відповідного нормального розподілу, а параметр β дорівнює $-E[x]/\sigma$. Таким чином, блок починає вивчати вікна зі стану, де розподіл активацій блоку співпадає з розподілом вхідних зображень на ImageNet, що теоретично дозволяє зробити максимально плавний перенос навчання.

3.4 Технології та засоби

Алгоритми машинного навчання повністю будуть реалізовані на мові Python, яка в останні роки стала де-факто стандартом у сфері науки про дані та штучного інтелекту завдяки неявній динамічній типізації, зручного інтерфейсу для роботи з багатовимірними масивами з використанням SIMD-паралелізму (бroadкастингу), простоті створювання обгортки над швидкими бібліотеками для обчислень, написаними на мовах C/C++ та Fortran, а також кросплатформності цієї мови. Для розробки алгоритмів буде використано ряд бібліотек та фреймворків, які дозволяють створювати оптимальні та лаконічні реалізації нейромережових моделей:

- NumPy – фундаментальний пакет для наукових обчислень з Python [39]. Він містить, серед іншого потужну імплементацію тензорних обчислень (вектори, матриці, n-вимірні масиви), складні високопаралельні функції роботи з матрицями та високопродуктивні інструменти лінійної алгебри;
- scikit-image – це набір алгоритмів для обробки зображень [40]. До цього набору відносяться як базові функції (такі як читання зображень з файлу), так і досить складні операції (наприклад, матричні перетворення, що необхідні для аугментації даних при навчанні нейронної мережі);
- pydicom – пакет для обробки DICOM-файлів на Python [41];
- PyTorch – фреймворк для тензорних обчислень [42], в якому реалізовані основні низькорівневі операції, потрібні для навчання та подальшого

використання нейронних мереж. PyTorch оптимізований для обчислень на графічних процесорах, тому його використання допоможе підтримувати необхідну швидкість роботи, що задана в нефункціональних вимогах. Крім того, на відміну від альтернатив (таких, як TensorFlow), PyTorch надає прозорий Python-інтерфейс для створення нових шарів;

- Fast.ai – високорівнева обгортка над PyTorch [43], що надає програмний інтерфейс для навчального циклу, а також реалізації багатьох складних операцій, таких як аугментації часу передбачення та циклічний темп навчання;
- Pytorch pretrained models – зоопарк претренованих згорткових мереж, реалізованих під PyTorch.

Розробка програмної системи проведена за допомогою середовищ Jupyter Notebook і PyCharm для Python. Для контролю вихідного коду використано розподілену систему контролю версій Git.

Розроблене програмне забезпечення включає в себе Jupyter Notebook [44] для тренування нейронних мереж в рамках експериментів, що описано вище, а також віджет для Jupyter для інтерактивної візуалізації вікон, що буде використовуватися для аналізу результатів.

Навчання нейронних мереж відбувається розподілено на двох відеокартах Nvidia GeForce 1080ti або на відеокарті V100 на Google Cloud Platform.

4 АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Експерименти з архітектурами

Перший проведений експеримент – вибір архітектури згорткового бекбоуну мережі. Для цього було перевірено чотири архітектури. Результати експерименту представлено в таблиці 4.1.

Таблиця 4.1 – Порівняння архітектур згорткових нейромереж

Назва архітектури	Кросентропія	Час навчання, годин
SE-ResNeXt-50	0.075	12
SE-ResNeXt-101	0.074	17
EfficientNet-B3	0.076	9
EfficientNet-B5	0.072	12.5

Однією з гіпотез, що було потрібно перевірити, була кореляція точності на ImageNet з метрикою на нашій задачі (рис. 4.1).

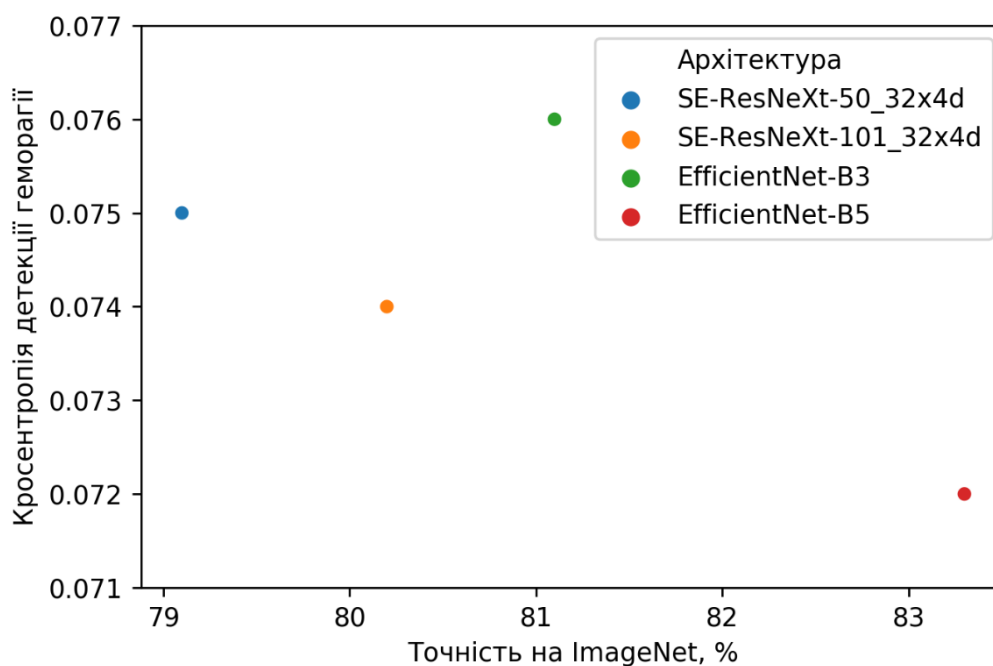


Рисунок 4.1 – Кореляція точності на ImageNet з кросентропією

Бачимо, що в рамках сімейств моделей є пряма залежність між точністю на ImageNet та результатами на наборі даних, що досліджується. Але між сімействами така кореляція не зберігається. Моделі з сімейства EfficientNet мають значно кращі результати на ImageNet, але порівняні результати на задачі розпізнавання геморагії. Як вже зазначалося вище, це може бути пов'язано з тим, що EfficientNet побудовані з допомогою нейронного пошуку архітектур та через це можуть буди надто заточені під ImageNet.

За методом головного критерія мережа SE-ResNeXt-101 не проходить по обмеженню на час навчання (рис. 4.2). Обмеження часу позначено чорною пунктирною лінією.

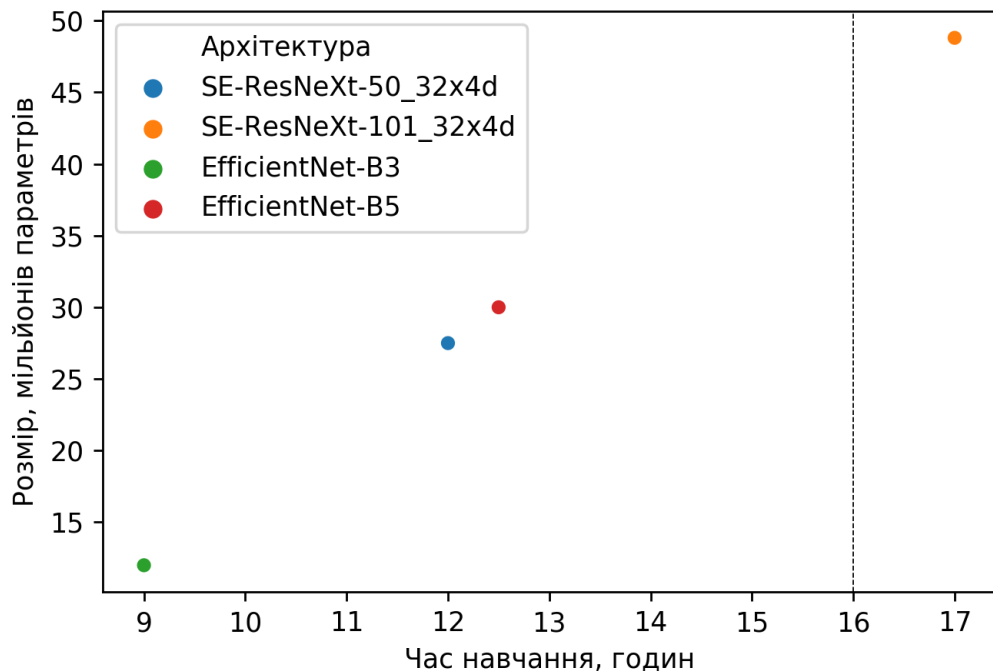


Рисунок 4.2 – Кореляція кількості параметрів та швидкості навчання

Бачимо з графіку, що час навчання майже лінійно корелює з кількістю параметрів мережі, як і очікувалося.

Отже, оскільки нам потрібно обрати дві архітектури для подальших експериментів, за методом головного критерію було обрано SE-ResNeXt-50 та EfficientNet-B5.

Подальші експерименти з пайплайном навчання та з вікнами будуть виконані на прикладі архітектури SE-ResNeXt-50. Після цього, вікна, що модуль автоматичної оптимізації вікон вивчить для цієї архітектури, будуть мануально перенесені на EfficientNet-B5 для перевірки того, що модуль вивчає осмислені вікна, які спрощують розпізнавання, а не перенавчається під конкретну мережу.

4.2 Експерименти з пайплайном навчання

В цих експериментах порівнюються два режими циклічного темпу навчання: трикутний та косинусоїдальний.

Верхня границя першого циклу, знайдена з використанням алгоритму пошуку оптимального темпу навчання – $2 \cdot 10^{-3}$. Для трикутного режиму нижня границя в десять разів нижче за верхню. Для косинусоїдального режиму нижня границя дорівнює нулю.

Результати наведено в таблиці 4.2.

Таблиця 4.2 – Порівняння скедулінгів темпу навчання

Режим циклів	Кросентропія
Трикутний	0.075
Косинусоїдальний	0.074

Бачимо, що косинусоїдальний циклічний темп навчання дає краще значення метрики.

Варто зазначити, що в обох випадках найкращий результат досягався на останню епоху останнього циклу навчання. Було проведено експерименти з подовшенням навчання до трьох циклів, але в обох випадках (для обох режимів) на третій цикл починалося сильне перенавчання: наприклад, для трикутного режиму

функція втрат на тренувальному наборі даних падала з 0.072 до 0.068, тоді як на валідаційному наборі даних вона підіймалася з 0.075 до 0.082.

Отже, для проведення подальших експериментів з вікнами було встановлено косинусоїдальний циклічний темп навчання.

4.3 Експерименти з вікнами

В експериментах з вікнами порівнюються п'ять режимів, зазначених вище. Попередні експерименти проводилися зі стандартним вікном мозку [45]. Результати наведено в таблиці 4.3.

Таблиця 4.3 – Порівняння віконних режимів

Режим	Кросентропія	Вікна
Стандартне вікно	0.074	$w=80, l=40$
Широке вікно	0.074	$w=256, l=80$
Три вікна	0.073	$w=80, l=40$ $w=128, l=60$ $w=256, l=80$
WSO (випадкова ініціалізація вікон)	0.077	$w=83, l=105$ $w=87, l=42$ $w=1393, l=457$
WSO (ініціалізація трьома вікнами)	0.072	$w=86, l=40$ $w=87, l=93$ $w=-114, l=-34$
Нормалізований WSO	0.071	$w=86, l=41$ $w=83, l=91$ $w=-122, l=-37$

По-перше, ми бачимо, що широке вікно не допомагає моделі, але використання трьох мануально відібраних вікон дозволяє мережі досягти більш високої точності.

По-друге, ми бачимо, що автоматична оптимізація вікна працює дещо краще, ніж використання трьох фіксованих вікон, а нормалізована автоматична оптимізація (N-WSO) працює ще краще, дозволяючи досягти мінімального значення кросентропії – 0.071. Важливо відмітити, що обидва алгоритми досягають таких значень метрики тільки за вимоги адекватної ініціалізації: випадкова ініціалізація вікон призводить до досить поганого значення кросентропії (причому по вивченим вікнам можна побачити, що модуль працює досить нестабільно та вивчає трансформації, які не мають сенсу з точки зору доменних експертів, адже третє вікно, наприклад, має низький контраст и демонструє здебільшого кістки).

Важливо дослідити, які саме вікна вивчає модуль WSO (рис. 4.3).

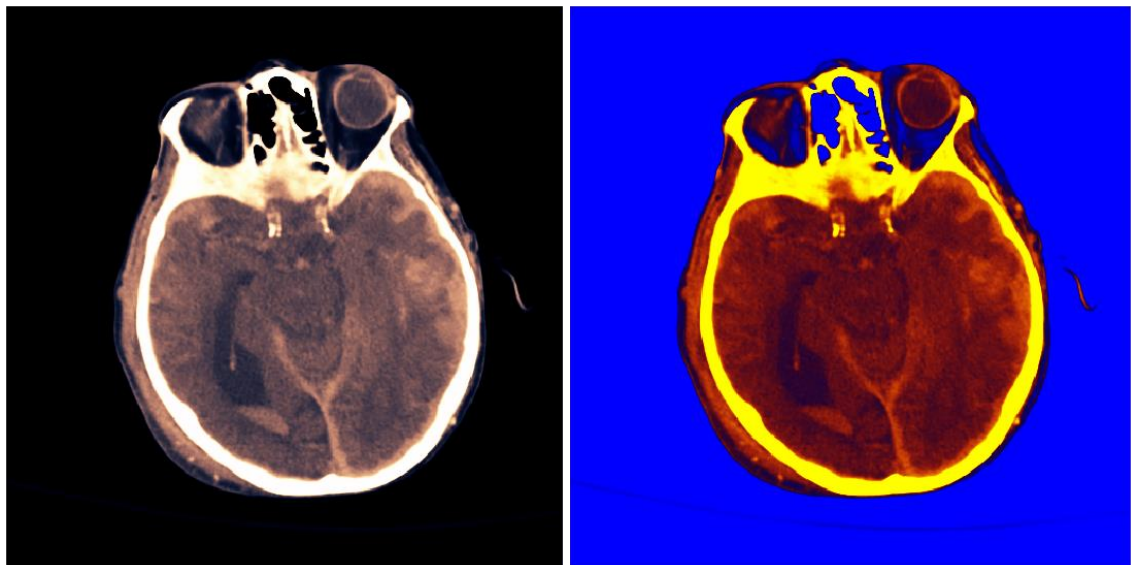


Рисунок 4.3 – Порівняння трьох вікон з вивченими вікнами

По-перше, ми бачимо, що і у WSO, і у нормалізованому WSO перші два вікна залишаються досить близько від тих значень, якими їх було ініційовано. Ширина обох вікон залишається близькою до 80, тобто ширини, яку використовують радіологи при реальних дослідженнях. Це означає, що модель вивчає значущі трансформації та зберігає гарні вікна. Також, бачимо, що навіть при випадковій

ініціалізації вікон вивчаються діапазони з шириною близько 80 (хоча рівні вікон й зовсім інші).

По-друге, цікаво, що WSO вивчає у якості третього вікна діапазон з негативною шириною. В реальних дослідженнях таке не використовується, але фізичний зміст легко інтерпретувати: це вікно, у якого більш насичений колір відповідає чорному на оригінальному зображенні, а менш яскравий – білому, тобто вікно інвертоване. Таке вікно призводить до підкреслення фону блакитним кольором та миттєвого збільшення контрастності зображення, що теоретично і є основною причиною покращення результатів у порівнянні з константними вікнами. З іншого боку, для людини-радіолога саме це інвертоване вікно робить зображення досить незвичним, тож для використання в мануальній діагностиці воно не дуже підходить.

По-третє, той факт, що WSO та нормалізований WSO сходяться до майже одних і тих же вікон, незважаючи на те, що нормалізований WSO дає дещо кращий результат, позначає, що саме значення вікон є оптимальними для інференсу (в протиставлення теорії, що модуль WSO покращує власне процес навчання). На відміну від цього, те, що нормалізований WSO демонструє дещо меншу кросентропію, найімовірніше пояснюється саме покращенням процесу переносу навчання.

Щоб в подальшому підтвердити теорію щодо доменної значущості вивчених вікон, навчимо EfficientNet-B5, подаючи на вхід три вікна та вікна, вивчені з використанням нормалізованого блоку автоматичного налаштування вікна. Результати експерименту наведено в таблиці 4.4.

Таблиця 4.4 – Порівняння віконних режимів

Режим	Кросентропія
Стандартне вікно	0.072
Три вікна	0.072
Вивчені вікна	0.069

Бачимо, що з вивченими вікнами навіть інша архітектура мережі отримує значне покращення точності в порівнянні з базовими конфігураціями.

4.4 Результати та подальші дослідження

Основним результатом роботи є створення нормалізованого модулю автоматичної оптимізації вікна (N-WSO), який значно покращує кросентропію. Порівняльні результати основних експериментів можна побачити на рисунку 4.4.

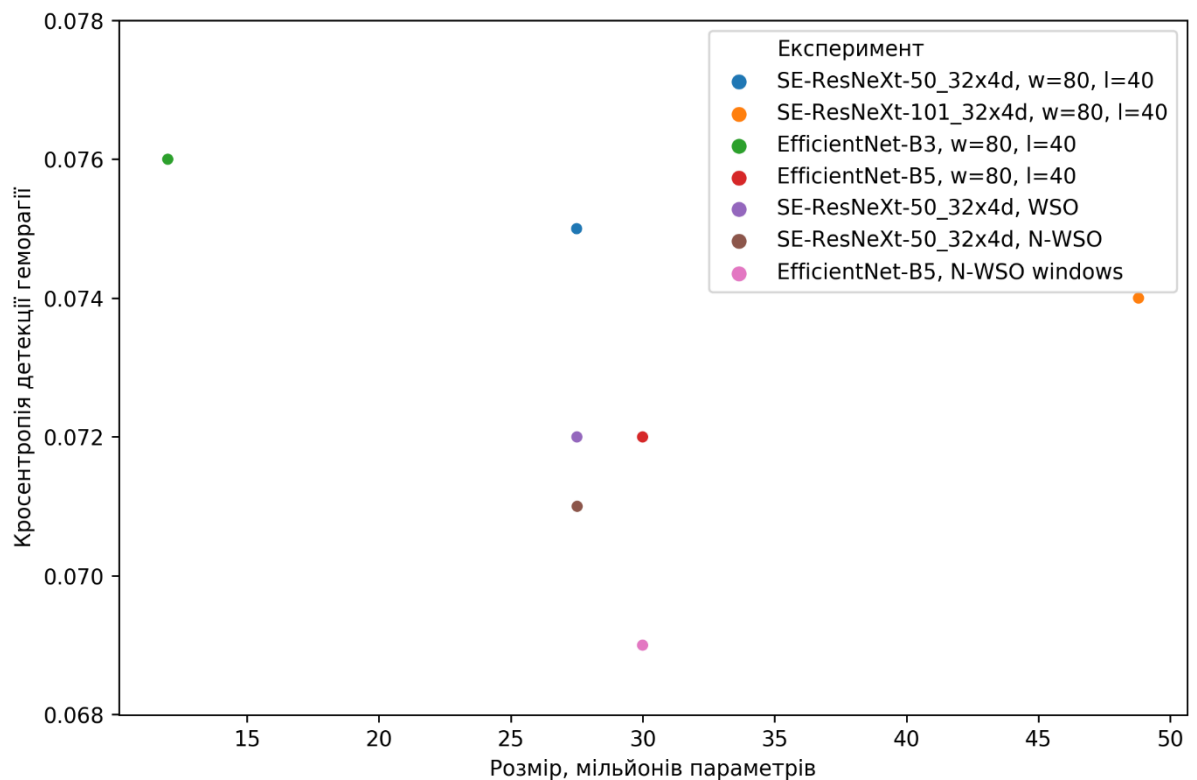


Рисунок 4.4 – Порівняння основних експериментів

Бачимо, що, завдяки правильно обраним гіперпараметрам та модулю автоматичної оптимізації вікна, мережа SE-ResNeXt-50, що демонструвала значно гірші результати ніж споріднена архітектура SE-ResNeXt-101, досягла істотно кращого значення метрики при майже вдвічі меншому розмірі (а отже вдвічі

більшій швидкості). Також, з модулем нормалізованої автоматичної оптимізації вікна вона навіть обігнала на один пункт по якості розпізнавання мозкової геморагії базову версію моделі EfficientNet-B5 на стандартному радіологічному вікні мозку, хоча на базових налаштуваннях ця архітектура залишалася непереможною в термінах кросентропії.

Варто зазначити, що хоча модуль N-WSO дещо впливає на час навчання, додаючи 12 параметрів, що тренуються, а також 3 параметра ковзаючих статистик у шарі батч-норму, але, по-перше, цей вклад майже непомітний на фоні десятків мільйонів параметрів мереж, а по-друге, під час передбачення цей модуль можна повністю заморозити і використовувати статичні вікна, взяті з нього.

В рамки цього дослідження не входила агрегація прогнозів для різних шарів томографічного дослідження. Потенціальним розвитком цієї роботи є стекинг вихідних результатів нейромережі з різних рівнів з використанням одномірної згорткової або рекурентної мережі або використання тривимірних згорток для подання кількох шарів до однієї згорткової мережі.

ВИСНОВКИ

В ході даної роботи було досліджено алгоритми класифікації томографічних результатів та запропоновано модифікацію алгоритму автоматичного пошуку вікна, що покращує результати розпізнавання на задачі класифікації геморагій мозку.

Перед початком розробки алгоритму було проаналізовано сучасний стан задачі класифікації зображень в цілому та зокрема томографічних знімків. Були зібрані базові наукові роботи, на які опиралося подальше дослідження. Було проведене ознайомлення з провідними підходами, та обрані найбільш ефективні нейромережеві архітектури та пайплайни.

Задача була формалізована як мультилейбельна ймовірнісна класифікація двовимірних зрізів томографії черепа на шість класів: наявність геморагії та п'ять її підвидів, що відрізняються локалізацією. Для дослідження було взято набір даних RSNA Intracranial Hemorrhage Detection з репозиторію наборів даних Kaggle, що складається з близько 670000 зображень. Якість класифікації на ньому оцінюється за метрикою бінарної кросентропії.

Був створений план експериментів, який включав в себе порівняння кількох архітектур нейромереж, серед яких за методом головного критерію було обрано EfficientNet-B5 та SE-ResNeXt-50. Після цього було проведено налаштування темпу навчання, де був обраний косинусоїдальний циклічний розклад. Також, якість розпізнавання покращилася після налаштування томографічних вікон. Було запропоновано покращення існуючого методу автоматичного налаштування вікон для випадку переносу навчання, яке дозволило досягти максимальної якості класифікації – 0.069 кросентропії з архітектурою EfficientNet-B5.

Для потенціального покращення результатів слід дивитися в бік переходу від розпізнавання двовимірних зрізів до безпосередньої роботи з тривимірними томографічними моделями шляхом застосування 1D-CNN або RNN поверх класифікаційної мережі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Krizhevsky A., Sutskever I., Hinton G.E. ImageNet classification with deep convolutional neural networks // Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012. С. 1097–1105.
2. Yerokhin A., Turuta O., Babii A., Nechyporenko A. Intelligent information system of heterogeneous medical data analysis // 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT). Lviv, 2017. С. 332–335.
3. Рашка С. Python и машинное обучение / під ред. А. Логунов. : ДМК-Пресс, 2017. 418 с. С. 26–31.
4. Николенко С., Кадурын А., Архангельская Е. Глубокое обучение. : Питер, 2017. 481 с. С. 17–19.
5. Go A. та ін. Heart disease and stroke statistics – 2013 update: a report from the American Heart Association // Circulation. 2013. Т. 127. № 1. С. e6–e245.
6. Heron M. Deaths: Leading Causes for 2017 // National Center for Health Statistics. 2019. Т. 68. № 6.
7. Прокоп М., Галански М. Спиральная и многослойная компьютерная томография. Москва: МЕДпресс-информ, 2011. 416 с.
8. Хофер М. Компьютерная томография. Базовое руководство. 2-ге вид., перероблене і доповнене. Москва : Медицинская литература, 2001. С. 16.
9. Burbridge B. Computed Tomography (CT) // Undergraduate Diagnostic Imaging Fundamentals. : University of Saskatchewan, Distance Education Unit, 2017.
10. Digital Imaging and Communications in Medicine (DICOM) Standard, National Electrical Manufacturers Association [Електронний ресурс]. URL: <https://www.dicomstandard.org/> (дата звернення: 22.03.2020).

11. Bidgood W.D. та ін. Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging // J Am Med Inform Assoc. 1997. Т. 4. № 3. С. 199–212.
12. Yerokhin A., Turuta O., Babii A., Nechyporenko A. A new intelligence-based approach for rhinomanometric data processing // 2016 IEEE 36th International Conference on Electronics and Nanotechnology (ELNANO). : IEEE, 2016. С. 198–201.
13. Yerokhin A., Turuta O., Babii A., Nechyporenko A. Usage of phase space diagram to finding significant features of rhinomanometric signals // 2016 XIth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT). : IEEE, 2016. С. 70–72.
14. Richfield D. Medical gallery of David Richfield // WikiJournal of Medicine. 2014. Т. 1. № 2. С. 9.
15. Deng J., Dong W., Socher R., Li L.-J., Li, K., Fei-Fei, L. Imagenet: A large-scale hierarchical image database // 2009 IEEE conference on computer vision and pattern recognition. : IEEE, 2009. С. 248–255.
16. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение. : ДМК Пресс, 2017. 652 с. С. 282–314.
17. Ioffe S., Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift // Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 ICML'15. Lille, France: JMLR.org, 2015. С. 448–456.
18. Srivastava N. та ін. Dropout: a simple way to prevent neural networks from overfitting // J. Mach. Learn. Res. 2014. Т. 15. № 1. С. 1929–1958.
19. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition // arXiv:1409.1556 [cs]. 2014.
20. Sandler M. та ін. MobileNetV2: Inverted Residuals and Linear Bottlenecks // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, 2018. С. 4510–4520.
21. Ramachandran P., Zoph B., Le Q.V. Searching for Activation Functions // arXiv:1710.05941 [cs]. 2017.

22. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA, 2016. С. 770–778.
23. Hu J., Shen L., Sun G. Squeeze-and-Excitation Networks // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, 2018. С. 7132–7141.
24. Tan M., Le Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks // arXiv:1905.11946 [cs, stat]. 2019.
25. Torrey L., Shavlik J. Transfer learning // Handbook of Research on Machine Learning Applications. 2009.
26. Zeiler M.D., Fergus R. Visualizing and Understanding Convolutional Networks // Computer Vision – ECCV 2014 Lecture Notes in Computer Science. / під ред. D. Fleet та ін. Cham: Springer International Publishing, 2014. С. 818–833.
27. Ouyang W., Winsnes C.F., Hjelmare M. та ін. Analysis of the Human Protein Atlas Image Classification competition // Nature Methods. 2019. Т. 16. № 12. С. 1254–1261.
28. Chilamkurthy S. та ін. Development and Validation of Deep Learning Algorithms for Detection of Critical Findings in Head CT Scans // arXiv:1803.05854 [cs]. 2018.
29. Lee H., Kim M., Do S. Practical Window Setting Optimization for Medical Image Deep Learning // arXiv:1812.00572 [cs]. 2018.
30. RSNA Intracranial Hemorrhage Detection [Електронний ресурс]. URL: <https://kaggle.com/c/rsna-intracranial-hemorrhage-detection> (дата звернення: 23.03.2020).
31. Rumetshofer E. та ін. Human-level Protein Localization with Convolutional Neural Networks // International Conference on Learning Representations. 2019.
32. Tan M., Pang R., Le Q.V. EfficientDet: Scalable and Efficient Object Detection // arXiv:1911.09070 [cs, eess]. 2019.
33. Hu J. та ін. Squeeze-and-Excitation Networks // arXiv:1709.01507 [cs]. 2017.

34. Smith L.N. Cyclical Learning Rates for Training Neural Networks // 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). 2017. С. 464–472.
35. Loshchilov I., Hutter F. SGDR: Stochastic Gradient Descent with Warm Restarts // arXiv:1608.03983 [cs, math]. 2016.
36. Liu L. та ін. On the Variance of the Adaptive Learning Rate and Beyond // arXiv:1908.03265 [cs, stat]. 2019.
37. Zhang M.R. та ін. Lookahead Optimizer: k steps forward, 1 step back // arXiv:1907.08610 [cs, stat]. 2019.
38. You Y., Gitman I., Ginsburg B. Large Batch Training of Convolutional Networks // arXiv:1708.03888 [cs]. 2017.
39. Oliphant T.E. A guide to NumPy. : Trelgol Publishing USA, 2006.
40. Van der Walt S. та ін. scikit-image: image processing in Python // PeerJ. 2014. Т. 2. С. e453.
41. Mason D. та ін. pydicom/pydicom: 1.1.0. : Zenodo, 2018.
42. Paszke A. та ін. PyTorch: An Imperative Style, High-Performance Deep Learning Library // Advances in Neural Information Processing Systems 32 / під ред. H. Wallach та ін. : Curran Associates, Inc., 2019. С. 8024–8035.
43. Howard J., Gugger S. Fastai: A Layered API for Deep Learning // Information. 2020. Т. 11. № 2. С. 108.
44. Kluyver T. та ін. Jupyter Notebooks – a publishing format for reproducible computational workflows // Positioning and Power in Academic Publishing: Players, Agents and Agendas / під ред. F. Loizides, B. Schmidt. : IOS Press, 2016. С. 87–90.
45. Murphy A. Windowing (CT). Radiology Reference Article – Radiopaedia.org [Електронний ресурс]. URL: <https://radiopaedia.org/articles/windowing-ct> (дата звернення: 03.04.2020).