

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

Інститут прикладного системного аналізу  
Кафедра математичних методів системного аналізу

На правах рукопису  
УДК 004.896

До захисту допущено  
В. о. завідувача кафедри ММСА  
О.Л.Тимошук

«\_\_\_» \_\_\_\_\_ 2021 р.

Магістерська дисертація

на здобуття ступеня магістра  
зі спеціальності 122 Комп'ютерні науки та інформаційні технології  
на тему: «Система автоматичної діагностики раку головного мозку із  
використанням штучних нейромереж»

Виконав:  
студент II курсу, групи КА-93мн  
Срібний Андрій Євгенович \_\_\_\_\_

Керівник:  
професор кафедри ММСА, д.т.н., доц.  
Зайченко Олена Юріївна \_\_\_\_\_

Рецензент:  
доцент кафедри автоматизації проектування  
енергетичних процесів і систем  
КПІ ім. Ігоря Сікорського, к.т.н., доц.  
Шаповалова Світлана Ігорівна \_\_\_\_\_

Засвідчую, що в цій магістерській  
Дисертації немає запозичень із праць  
інших авторів без відповідних посилань  
Студент \_\_\_\_\_

Київ  
2021

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

Інститут прикладного системного аналізу  
Кафедра математичних методів системного аналізу

Рівень вищої освіти — другий (магістерський)  
Спеціальність — 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри ММСА

О. Л. Тимошук

«\_\_\_» \_\_\_\_\_ 2021 р.

**ЗАВДАННЯ**

на магістерську дисертацію студента Срібного Андрія Євгеновича

**1. Тема дисертації:** «Система автоматичної діагностики раку головного мозку із використанням штучних нейромереж», науковий керівник дисертації Зайченко Олена Юріївна, професор кафедри ММСА, д.т.н., затверджені наказом по університету від 17 березня 2021 р № 851-с.

**2. Термін подання студентом дисертації:** 14 травня 2021 р.

**3. Об'єкт дослідження:** зображення МРТ головного мозку.

**4. Предмет дослідження:** методи обробки зображень та штучні нейронні мережі для класифікації зображень.

**5. Перелік завдань, які потрібно розробити:**

- 1) дослідити сучасні методи класифікації медичних зображень;
- 2) пошук та обробка даних для подальшого використання;
- 3) генерація додаткових даних для тренування нейронної мережі;

- 4) реалізувати нейронну мережу, яка на вхід зможе приймати зображення магнітно-резонансної томографії головного мозку
- 5) виконати експерименти, навчити модель на згенерованих даних та проаналізувати отриманий результат
- 6) розробити концептуальні висновки за результатами наукового дослідження;
- 7) оформити наукову статтю.

## 6. Орієнтовний перелік графічного (ілюстративного) матеріалу

1. Рисунки пихулини головного мозку.
2. Запропонована архітектура нейронної мережі та моделей-аналогів.
3. Таблиці порівняння результатів роботи.
4. Рисунки та графіки на яких зображено результати роботи

## 7. Орієнтовний перелік публікацій:

- 1) Зайченко О. Ю., Срібний А. Є., Система автоматичної діагностики раку головного мозку із застосуванням нейронних мереж, *IX Міжнародна науково-практична конференція «THE WORLD OF SCIENCE AND INNOVATION»*, 7-9 квітня 2021 року Лондон, Великобританія, С. 310-314.

URL: <https://sci-conf.com.ua/wp-content/uploads/2021/04/THE-WORLD-OF-SCIENCE-AND-INNOVATION-7-9.04.21.pdf>

**8. Дата видачі завдання:** 10 березня 2021 р.

## Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації
1	Концептуальний вступ дисертації. Формулювання об'єкта, предмета, цілі, завдань, новизни, практичної значущості результатів	10.03.2021—20.03.2021
2	Перший розділ. Огляд літературно-інформаційних джерел, формування нормативної бази. Характеристика об'єкта.	21.03.2021—01.04.2021
3	Другий розділ. Опис теоретичних основ методів дослідження.	02.04.2021—07.04.2021

4	Третій розділ. Обробка вхідних даних. Реалізація та застосування обраної моделі. Збір та аналіз результатів.	08.04.2021—25.04.2021
5	Концептуальні висновки. Перспективи розвитку отриманих рішень	26.04.2021—30.04.2021
6	Оформлення наукової статті	01.05.2021—05.05.2021

Студент

А. Є. Срібний

Науковий керівник дисертації

О. Ю. Зайченко

## РЕФЕРАТ

Магістерська дисертація: 95 с., 20 рис., 24 табл., 1 додатки, 45 джерел.

Об'єктом дослідження є МРТ зображення головного мозку людини.

Предметом дослідження є методи обробки зображень та штучні нейронні мережі для класифікації зображень.

Мета роботи – проаналізувати МРТ зображення на наявність раку головного мозку, дослідження існуючих підходів та розробка програмного забезпечення.

Виконано аналіз методів автоматичного розпізнавання і знаходження раку головного мозку. Було використано техніку збільшення даних та розроблено модель нейронної мережі для класифікації зображень МРТ й порівняно з аналогами.

В результаті було розроблено програмний продукт, який дозволяє класифікувати зображення МРТ головного мозку з раком.

ПУХЛИНА МОЗКУ, МРТ, ГЛИБОКЕ НАВЧАННЯ, ТРАНСФЕРНЕ НАВЧАННЯ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, МАШИННЕ НАВЧАННЯ

## ABSTRACT

Master's thesis: 95 pages, 20 figures, 24 tables, 1 appendixes, 45 sources.

The object of the study is an MRI image of the human brain.

The subject of the research is image processing methods and artificial neural networks for image classification.

The aim of the work is to analyze MRI images on the provision of brain cancer, study existing approaches and develop software.

The analysis of methods of automatic detection and detection of brain cancer is performed. The technique of using messages was used and a neural network model was developed for the classification of MRI images and compared with analogues.

As a result, a software product was developed that allows the classification of MRI images of the brain with cancer.

BRAIN TUMOR, MRI, DEEP LEARNING, TRANSFER LEARNING,  
CONVOLUTIONAL NEURAL NETWORKS, CLASSIFICATION OF IMAGES,  
MACHINE LEARNING

## ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 АНАЛІЗ ЗАДАЧІ. ОСНОВНІ ПОНЯТТЯ РАКУ МОЗКУ, ДІАГНОСТИКА.....	11
1.1    Концепт та термінологія.....	11
1.2    Медична візуалізація та методи діагностики пухлини головного мозку	13
1.3    МРТ Характеристика зображення пухлини головного мозку .....	15
1.4    Епідеміологія .....	16
1.5    Фактори ризику .....	17
1.6    Генетика .....	18
1.7    Клінічна презентація.....	19
1.8    Висновки до розділу 1 .....	20
РОЗДІЛ 2 ОСНОВИ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ.....	21
2.1    Вступ.....	21
2.2    Згорткові нейронні мережі .....	22
2.3    Переваги використання ЗНМ .....	24
2.4    Шари ЗНМ.....	24
2.5    Функції втрат .....	30
2.6    Регуляризація ЗНМ.....	32
2.7    Вибір оптимізатора.....	34
2.8    Розробка алгоритмів (зворотне розповсюдження помилки) .....	37
2.9    Аналіз існуючих робіт.....	40
2.10   Висновки до розділу 2.....	42
РОЗДІЛ 3 ОПИС АРХІТЕКТУРИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ.....	43
3.1    Запропонований підхід.....	43
3.2    Обробка зображень.....	44
3.3    Аугментація даних.....	44
3.4    Опис запропонованої моделі згорткової нейронної мережі.....	45

	8
3.5 Функція втрат.....	47
3.6 Оптимізація .....	48
3.7 Трансферне навчання .....	49
3.7.1 VGG-16 .....	49
3.7.2 Inception-v3.....	50
3.7.3 ResNet50 .....	51
3.8 Аналіз результатів .....	52
3.9 Висновки до розділу 3 .....	56
РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ .....	57
4.1 Опис ідеї проекту.....	58
4.2 Технологічний аудит ідеї проекту .....	60
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	61
4.4 Аналіз ринкової стратегії проекту .....	69
4.5 Розроблення маркетингової програми стартап-проекту.....	72
4.6 Висновки до розділу 4.....	77
ВИСНОВКИ .....	78
ПЕРЕЛІК ПОСИЛАНЬ.....	80



## ВСТУП

За даними Всесвітньої організації охорони здоров'я, рак є другою провідною причиною смерті у всьому світі. Раннє виявлення раку може запобігти смерті, але не завжди. На відміну від раку, пухлина може бути доброякісною або злоякісною. Доброякісні пухлини відрізняються від злоякісних тим, що доброякісні, як правило, не поширюються на інші органи та тканини і можуть бути видалені хірургічним шляхом.

Деякі з первинних пухлин головного мозку - це гліоми, менінгіоми та пухлини гіпофіза. Гліоми - загальний термін для пухлин, що виникають із тканин головного мозку, крім нервових клітин та судин. З іншого боку, менінгіоми виникають із оболонок, які покривають головний мозок і оточують центральну нервову систему, тоді як пухлини гіпофіза - це грудочки, які знаходяться усередині черепа. Найважливіша відмінність між цими трьома типами пухлин полягає в тому, що менінгіоми, як правило, доброякісні, а гліоми найчастіше злоякісні. Пухлини гіпофіза, навіть доброякісні, можуть спричинити інші медичні пошкодження, на відміну від менінгіом, які є повільно зростаючими пухлинами. Через вищезазначену інформацію точна диференціація між цими трьома типами пухлин є дуже важливим етапом клінічного діагностичного процесу та пізніше ефективної оцінки пацієнтів.

Пухлина головного мозку діагностується за допомогою декількох методів, таких як КТ, ЕЕГ, але зображення магнітних ресурсів (МРТ) є найбільш ефективним і широко застосовуваним методом. МРТ використовує потужні та ефективні магнітні поля та радіохвилі для створення внутрішніх зображень органів всередині тіла. МРТ надає більш детальну інформацію про внутрішні органи і, отже, є більш ефективним, ніж сканування КТ або ЕЕГ. Раннє виявлення раку головного мозку та пухлини здебільшого залежить від досвіду рентгенолога. Діагностика пухлини не могла бути вірною на 100% до

встановлення доброякісної чи злоякісної пухлини. Для того, щоб дослідити, доброякісна чи злоякісна тканина, зазвичай проводять біопсію. На відміну від пухлин в інших місцях тіла, біопсія пухлини головного мозку зазвичай не отримується до остаточної хірургічної операції на мозку. Для отримання точної діагностики та уникнення хірургічного втручання та суб'єктивності, важливо розробити ефективний інструмент класифікації пухлин за зображеннями МРТ.

Протягом останніх кількох років завдяки штучному інтелекту та глибокому навчанню було досягнуто значного прогресу в медичній науці, такій як техніка обробки медичних зображень, яка допомагає лікарям діагностувати захворювання швидко і легко, до цього це було трудомістко.

Найбільша проблема з класифікацією та сегментуванням МРТ-зображень за допомогою деяких нейронних мереж полягає в кількості зображень. Крім того, МРТ-зображення отримують у різних площинах, тому можливість використання всіх доступних площин може розширити базу даних. Оскільки це, як правило, може вплинути на вихід класифікації через переобладнання, необхідна попередня обробка перед подачею зображень у нейронну мережу.

Для вирішення такого роду завдань вкрай необхідна автоматизована технологія, оскільки медична сфера потребує ефективних та надійних методів діагностики таких загрозливих для життя захворювань, як рак, який є провідною причиною смертності у всьому світі для пацієнтів.

Метою цього дослідження є зробити автоматизовану систему класифікації зображень. Отже, за допомогою МРТ-зображень головного мозку було запропоновано метод класифікації пухлин головного мозку на ракові та неракові, використовуючи техніку збільшення даних та нову модель згорткової нейронної мережі.

## РОЗДІЛ 1 АНАЛІЗ ЗАДАЧІ. ОСНОВНІ ПОНЯТТЯ РАКУ МОЗКУ, ДІАГНОСТИКА

### 1.1 Концепт та термінологія

Пухлина - це аномальна маса, яка може існувати всередині або на мозку. Для цієї аномальної та ненормальної частини мозку використовуються два різні терміни:

- пухлина;
- рак.

Пухлина і рак не мають однакових характеристик. Пухлина - це тверда або наповнена рідиною маса аномальних тканин. Пухлина також називається новоутворенням. Пухлина може бути категоризована на первинну та вторинну пухлину [1]. Первинна пухлина складається з клітин того органу, де локалізується пухлина. Здебільшого первинна пухлина підтримується нервовою системою для зростання, і ріст пухлини відбувається дуже повільно. Цей тип пухлини, який пов'язаний з нервовою системою, називається гліомами, а клітини глії мозку - це будівельний матеріал.

Рак - це швидке і неконтрольоване зростання аномальних тканин, що пошкоджує сусідні тканини здоров'я мозку.

Пухлина класифікується на доброякісну, злоякісну та дозлосякісну. Доброякісні містять неракові властивості. Злоякісний містить ракову характеристику. Предозлосякісність має передракову характеристику.

Вторинна пухлина складається з клітин, які належать до різних та інших частин тіла. Іншими словами, можна сказати, що ракові клітини є причиною вторинної пухлини. Отже, робиться висновок, що всі пухлини не є раком, але всі пухлини є пухлинами. Пухлина може бути класифікована на основі різних критеріїв, наведених нижче:

- локалізація пухлини в черепі;
- локалізація пухлини в мозку;

-локалізація в відділі.

Крім цього, пухлина також класифікується на основі клітин, які складають пухлину, як

- пухлина складається з нейронових клітин;
- пухлина складається з клітин глії;
- пухлина складається із статевих клітин;
- пухлина складається з мозкових оболонок.

Категорія домінуючої основи патології наведена нижче

- доброякісна;
- злоякісна.

На рисунку 1.1 показано зображення пухлини.



Рисунок 1.1 – Приклад мозку з пухлиною

## 1.2 Медична візуалізація та методи діагностики пухлини головного мозку

Своєчасна діагностика допомагає в лікувальній процедурі. Для діагностики пухлини та причин та наслідків цієї хвороби використовуються різні методи, такі як біопсія мозку та система візуалізації мозку. Біопсія головного мозку - це процедура, при якій на черепі проколюють отвір і видаляють шматочок тканини та пухлини, щоб дослідити пухлину, тип пухлини, її склад та причину пухлини під мікроскопом. Рисунок 1.2 показує процес біопсії [2].

Цей прийом дуже ризикований для життя людини. Метод візуалізації також використовується при біопсії для виявлення пухлини та отримання частини тканини.

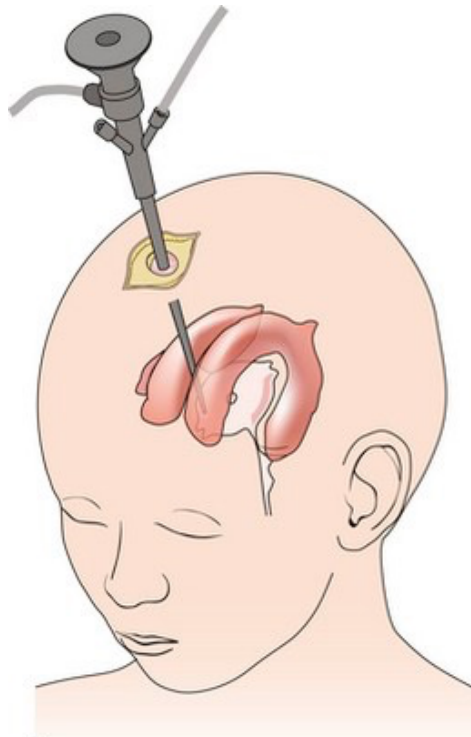


Рисунок 1.2 – біопсія головного мозку

Для отримання зображень головного мозку використовуються різні методи візуалізації, щоб пухлина могла діагностуватися за місцем розташування та розміром пухлини, як рентген, КТ та МРТ [3].

КТ є важливим методом візуалізації в медичній галузі та надає інформацію за лічені секунди, і зазвичай тривалість мінімізується до її частки. Це допомагає забезпечити більш чітку інформацію, ніж рентгенівські промені, але ризик радіаційного опромінення дуже низький

ПЕТ - позитронно-емісійна томографія, при якій в кров вводять радіоактивний матеріал, і сканер виявляє цей матеріал для отримання зображення. Ця методика дає уявлення про діяльність та функції мозку. Цей метод є економічно вигідним шкідливим матеріалом.

Рентген - це техніка візуалізації, яка не дає детальної інформації про орган. Рентген може спричинити рак шкіри, якщо його використовувати кілька разів на одному тілі та місці. Але ця техніка менш затратна і проста у використанні.

МРТ - це ще одна методика, яка використовує радіочастотні сигнали для отримання зображення мозку. Ця техніка візуалізації є нашою технікою фокусування. Загальні методи візуалізації зображені на рисунку 1.3.

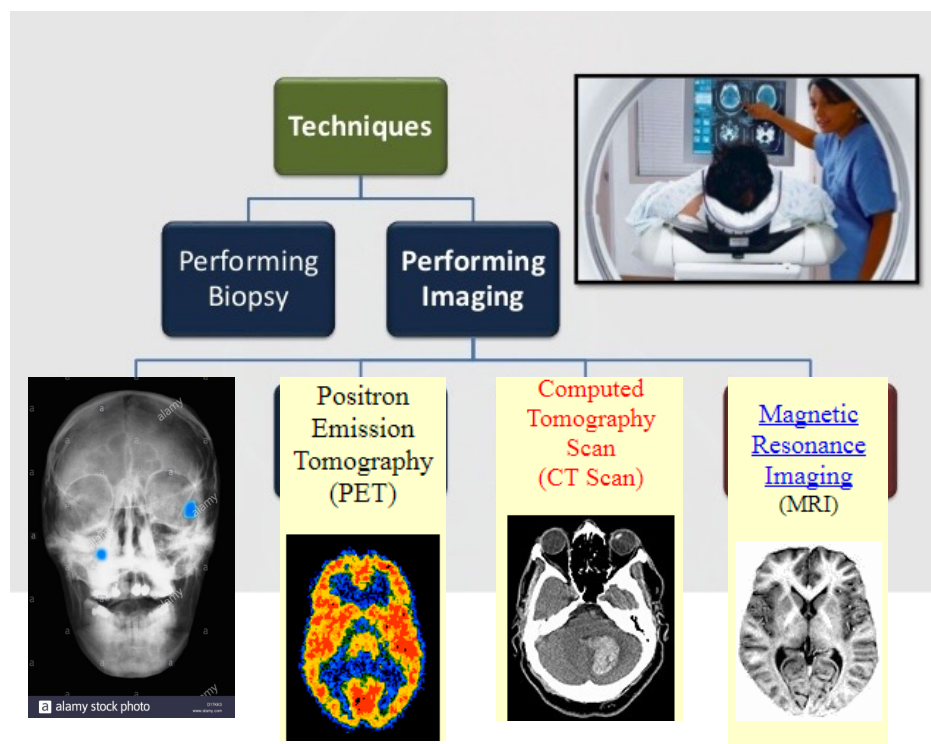


Рисунок 1.3 – Різні види візуалізації головного мозку

### 1.3 МРТ Характеристика зображення пухлини головного мозку

МРТ є методом візуалізації, який є більш корисним, ніж рентген [4]. МРТ-зображення не використовують шкідливих випромінювань і забезпечують достатньо інформації для діагностики та прийняття рішень лікарями. Зображення використовують у попередній обробці виявлення та діагностики пухлини головного мозку. У цій процедурі використовуються різні типи МРТ відповідно до вимог [5]. Тип послідовностей, що використовуються в МРТ, надаються як вхідні дані на етапі попередньої обробки, такі як T1, T2 та FLAIR.

Щоб зрозуміти концепцію різних типів МРТ-зображень, необхідно чітко зрозуміти концепцію TE та TR. TE - це (час ехо) різниця в часі між подачею радіочастотного імпульсу та прийманням ехосигналу. TR - це (час повторення) час прийому між двома безперервними імпульсами, поданими в одній послідовності.

T1-зважені зображення: містять темний вигляд ліквору та рідини [6]. Сіра речовина (ГМ) темніше білої речовини (ЗМ). T1 дає кращий результат у випадку зображень структури мозку, і жир у цього типу виглядає яскравішим. Час TE та TR ( $TR \approx 500$  мсек,  $TE \approx 14$  мсек) є коротким для отримання зображень (використовується позовжня релаксація).

T2-зважені зображення: які містять вищу інтенсивність сигналу ліквору та рідини порівняно з тканиною, і з цієї причини вони виглядають яскравими [7]. T2 використовував тривалий час ( $TR \approx 4000$  msec,  $TE \approx 19$  msec) для TE і TR для створення зображень (поперечна релаксація). T2 яскравіший для води та рідини, ідеально підходить для набрякової тканини

FLAIR подібно до T2, але він ослабив рідину ліквору, але відхилення залишаються яскравими [8]. Це добре для візуалізації набряку мозку. Він використовує дуже тривалий час TE та TR ( $TR \approx 9000$  msec,  $TE \approx 114$  msec) для

створення зображень. Рисунок 1.4 представляє різницю між цими типами послідовностей на зображенні МРТ.

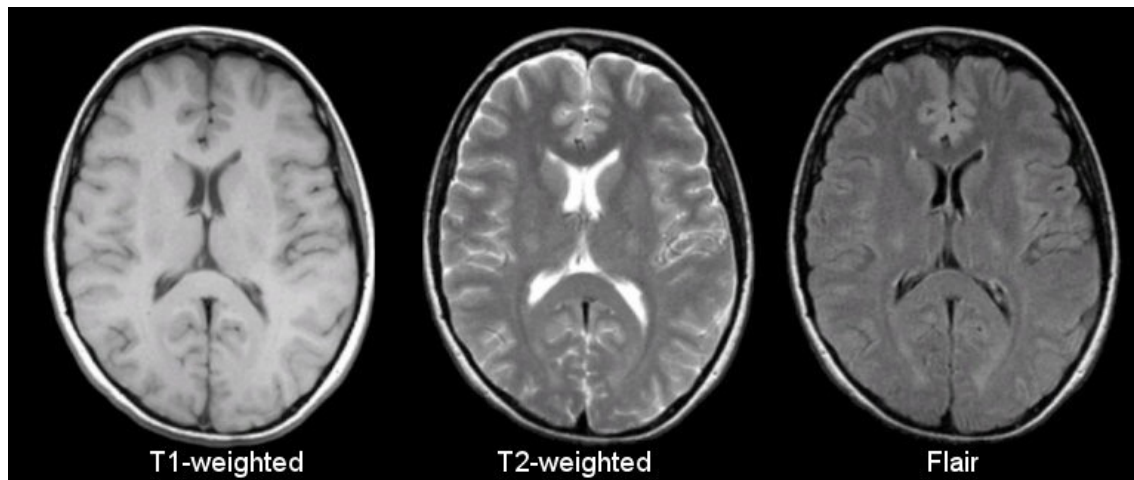


Рисунок 1.4 – Типи МРТ

#### 1.4 Епідеміологія

Цього року в США буде діагностовано близько 69 720 нових випадків первинних пухлин ЦНС (Центральної нервової системи). З цих уражень близько 24 620 будуть злоякісними. Хоча частота виникнення первинних пухлин головного мозку порівняно низька в порівнянні з іншими типами раку, первинні пухлини головного мозку спринцюють непропорційну кількість захворюваності та змісту, часто спричиняючи виснажувальні порушення руху та мови людей. Хоча первинні пухлини ЦНС складають лише 1,4% від усіх видів раку, вони є одними з найбільших агресивних пухлин і призводять до загальної смертності близько у 60% випадків [9].

Більшість пухлин головного мозку підключають за схемою класифікації ВООЗ пухлин нейроепітеліальної тканини. Злоякісні гліоми - найпоширеніша первинна пухлина головного мозку, яка складає понад 80 відсотків усіх первинних новоутворень головного мозку. Гліоми можна розділити на астроцити, олігодендрогліоми, епендимому та олігоастроцити (змішані гліоми). Якщо нейрогліальні пухлини можна



розділити на основі ступеня. Астроцитоми підрозділяються на I-IV ступенях наступним чином: пілоцитарні, I ступеня; дифузний, II ступінь; анапластичний, III ступінь; та мультиформна гліобластома (GBM), IV ступінь. Епендимомы поділяються на I-III ступені. Олігодендрогліоми, як правило, мають II ступінь, а олігоастроцитомами потрібно III ступінь. олігоастроцитоми (III ступінь ВООЗ) разом складають 10%. Останні 5-20% представляють рідкісні типи пухлин, включаючи анапластичні епендимомы та анапластичні гангліогліоми.

### 1.5 Фактори ризику

Кілька досліджуваних факторів ризику пухлин головного мозку, але наші знання про їх етіологію залишаються обмеженими. Хоча єдиним явним фактором ризику, який був виявлений для гліальних та менінгеальних новоутворень, є іонізуюче випромінювання, деякі дослідники спостерігали асоціації для підтримки потенційних факторів ризику первинних пухлин головного мозку [10]. Було проведено багато професійних досліджень, щоб визначити відносний ризик пухлин головного мозку, не маючи остаточної зв'язку з певними хімічними речовинами чи експозицією. Докази радіаційного опромінення (наприклад, електромагнітного випромінювання та стільникових телефонів та радіочастотного випромінювання), крім іонізуючого, що збільшує ризик розвитку первинної пухлини мозку, є безперечними [11]. Оцінка додаткових факторів ризику, включаючи травму голови, алергію, дієту, тютюн та алкоголь, також дала суперечливі результати з підвищеним ризиком розвитку первинних пухлин головного мозку. Паралельно з цими дослідженнями зростає вивчення вірусних та генетичних причин.

## 1.6 Генетика

Кілька спадкових генетичних синдромів були пов'язані з розвитком первинної пухлини головного мозку. Згідно з оглядом, проведеним Бонді та його колегами, кілька спадкових синдромів, включаючи нейрофіброматоз типів 1 і 2 (NF1 і NF2), туберкульозний склероз, синдром базальної клітини, Синдром Турко, синдром Лі-Фраумені та синдром фон-Гіппеля-Ліндау (VHL) всі вони становлять генетичну схильність до розвитку пухлини мозку [12]. Бульбовий склероз є генетичним захворюванням, що вражає множинні системи і асоціюється з доброякісними пухлинами мозку та інших життєво важливих органів. Синдром уникнення базаліоми також впливає на безліч систем в організмі людини і пов'язаний із підвищеним ризиком розвитку медуллобластоми [13]. Синдром Туркота, також відомий як синдром відновлення невідповідності через асоціацію біалельних мутацій відновлення невідповідності ДНК, пов'язаний з декількома симптомами, включаючи пухлини мозку (медуллобластоми та гліоми) та поліпоз товстої кишки. Синдром Лі-Фраумені - це аутосомно-домінантна ознака, яка зазвичай пов'язана із спадковою мутацією гена TP53. У пацієнтів із синдромом Лі-Фраумені підвищений ризик розвитку раку, включаючи пухлини головного мозку. Мутації зародкових ліній TP53 спостерігались частіше у пацієнтів з мультифокальними гліомами, гліомою та іншим первинним злоякісним захворюванням або гліомою, пов'язаною із сімейною історією раку, ніж у пацієнтів з іншими пухлинами головного мозку. Синдром VHL - це аутосомно-домінантне розлад, пов'язане з мутацією супресора пухлини VHL. Отримана мутація призводить до гемангіобластоми, кіст підшлункової залози та нейроендокринних пухлин, пухлин нирок та феохромоцитом. Нарешті, збільшення частоти гліом у сім'ях було задокументовано в декількох звітах. За оцінками, 5% усіх випадків гліоми можуть бути сімейними. Однак

характер розвитку гліоми у багатьох сім'ях вказує на екологічні причини, а не на схильне спадкове захворювання.

### 1.7 Клінічна презентація

Пацієнти з первинними пухлинами головного мозку можуть мати загальні або вогнищеві ознаки та симптоми [14]. Як правило, генералізовані симптоми виникають пізніше в патогенезі захворювання, коли пухлина розростається і спричинює підвищений внутрішньочерепний тиск, що призводить до головного болю, судом, нудоти, блювоти та зміни психічної функції. Такі вогнищеві симптоми, як вогнищевий неврологічний дефіцит (наприклад, геміпарез та афазія), відносяться до пухлин низького або високого ступеня та відображають внутрішньочерепне розташування пухлини. Приблизно 77% всіх пацієнтів з первинними пухлинами головного мозку відзначають тупий головний біль напруженого типу, який може зберігатися більше 50 місяців у 50% пацієнтів [15]. Хоча головний біль є найпоширенішим початковим симптомом, інші симптоми часто асоціюються з головними болями, включаючи напади у 50% пацієнтів, порушення зору у 40% пацієнтів та нудоту та блювоту у 38% пацієнтів.

Судоми часто зустрічаються у пацієнтів з пухлинами головного мозку, у яких 15-95% пацієнтів мають принаймні один судом під час хвороби [16]. У пацієнтів із гліомами низького ступеня випадки частіше спостерігаються судоми (65-95%), ніж у пацієнтів із ГБМ (15-25%). В одному великому дослідженні 23% пацієнтів перенесли принаймні один напад до діагностики пухлини. Вісімнадцять відсотків пацієнтів з ГБМ спочатку мали напади. Вік пацієнта, локалізація пухлини та гістологія пов'язані з появою судом. Пацієнти у віці 30-50 років частіше відчували напади. Ураження пухлини лобової, лобово-тім'яної, скроневої та лобно-скроневої часток частіше

супроводжувалось судомами [17]. Нарешті, виходячи з гістологічного діагнозу, у пацієнтів із змішаними гліомами, олігодендрогліомами та астроцитомами судоми спостерігались відповідно у 62%, 53% та 42%.

## 1.8 Висновки до розділу 1

У даному розділі було наведено всі необхідні теоретичні відомості щодо видів пухлин, раку мозку. Детально описано генетику захворювань, фактори ризику та розглянуті різні методи отримання зображень мозку людини.

## РОЗДІЛ 2 ОСНОВИ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

### 2.1 Вступ

Штучні нейронні мережі мають перевагу перед статистичними методами класифікації. Вони підходять у випадках, коли традиційні методи класифікації не вдаються через зашумлені або неповні дані. Нейронні мережі також мають перевагу у варіативних задачах класифікації з високим ступенем кореляції. Діагностика захворювань є гарним прикладом таких складних проблем класифікації. На основі цієї узагальненої моделі можна класифікувати вхідні дані, що представляють різні симптоми захворювань.

Весь процес діагностики хвороб можна розділити на навчально-діагностичну частину. Загалом навчальний процес починається з вибору цільових захворювань, для яких буде пов'язана проблема класифікації. Після відповідного вибору захворювання необхідно визначити конкретні параметри, симптоми та лабораторні результати, які детально описують характер цієї хвороби. На наступному кроці на основі цих даних створюється база даних, яку потрібно перевірити, а крайні значення поза діапазоном - відкинути. Нейромережа проходить навчання за допомогою цієї бази даних, а потім результати, отримані в цьому процесі, перевіряються. Якщо результати навченої нейронної мережі є правильними, то нейронну модель можна використовувати в медичній практиці. З цього кроку починається процес діагностики. Дані пацієнта обробляються нейронною мережею, яка визначає ймовірний діагноз. Потім цей результат підтверджує лікуючий лікар. Остаточний діагноз є результатом рішення лікаря, який на основі власного досвіду оцінює всі аспекти захворювання та результат класифікації нейронних мереж.

## 2.2 Згорткові нейронні мережі

У галузі глибокого навчання згорткові нейронні мережі (ЗНМ) є найвідомішим та найпоширенішим алгоритмом. Основною перевагою ЗНМ у порівнянні з іншими є те, що вона автоматично визначає відповідні риси без будь-якого людського втручання. ЗНМ широко застосовуються в цілому ряді різних галузей, включаючи комп'ютерний зір, обробку мови, розпізнавання облич та ін. Структура ЗНМ натхнена нейронами в мозку людини та тварин, подібних до звичайної нейронної мережі. У мозку kota складна послідовність клітин утворює зорову кору; ця послідовність моделюється ЗНМ. Було визначено три ключові переваги ЗНМ: еквівалентні подання, розріджені взаємодії та спільний доступ до параметрів. На відміну від звичайних повністю з'єднаних мереж, спільні ваги та локальні з'єднання в ЗНМ використовуються для повного використання 2D структур вхідних даних, таких як сигнали зображення. Ця операція використовує надзвичайно малу кількість параметрів, що спрощує навчальний процес і пришвидшує роботу мережі. Це те саме, що і в клітинах зорової кори. Ці комірки сприймають лише невеликі ділянки сцени, а не всю сцену (тобто ці клітинки просторово витягують локальну кореляцію, доступну у вході, як локальні фільтри над входом). Найчастіше використовуваний тип ЗНМ, схожий на багат шаровий персептрон, складається з численних шарів згортки, що передують шарам субдискретизації (об'єднання), тоді як кінцеві шари – повнозв'язні. Приклад архітектури ЗНМ для класифікації зображень проілюстровано на рисунку 2.1. Вхід  $x$  кожного шару в моделі CNN організований у трьох вимірах: висота, ширина та глибина, або  $m \times m \times r$ , де висота ( $m$ ) дорівнює ширині. Глибина також називається номером каналу. Наприклад, на зображенні RGB глибина ( $r$ ) дорівнює трьом. Кілька ядер (filters), доступних у кожному згортковому шарі, позначаються  $k$  і також мають три виміри ( $n \times n \times q$ ), подібні до вхідного зображення; однак  $n$  має бути менше  $m$ , тоді як  $q$  або дорівнює  $r$ ,

або менше. Крім того, ядра є основою локальних з'єднань, які мають однакові параметри (зміщення  $b^k$  та вага  $W^k$ ) для генерації  $k$ -карт ознак  $h^k$  розміром  $(m - n - 1)$  кожна та конвертуються з вхідними даними, як зазначалося вище. Шар згортки обчислює точковий добуток між його входом та вагами, як у рівнянні 1. Далі, застосовуючи нелінійність або функцію активації до виходу рівня згортки, ми отримуємо наступне:

$$h^k = f(W^k * x + b^k) \quad (1)$$

Наступним кроком є видалення з вибірки для кожної карти ознак у шарах субдискретизації. Це призводить до зменшення параметрів мережі, що прискорює тренувальний процес і, в свою чергу, дозволяє вирішити проблему перенавчання. Для всіх карт ознак функція пулінгу (наприклад, максимальний або середній) застосовується до сусідньої області розміром  $p \times p$ , де  $p$  - розмір ядра. Повнозв'язні шари отримують ознаки середнього та низького рівня і створюють абстракцію високого рівня, яка представляє шари останньої стадії, як у типовій нейронній мережі. Бали класифікації генеруються за допомогою кінцевого шару.

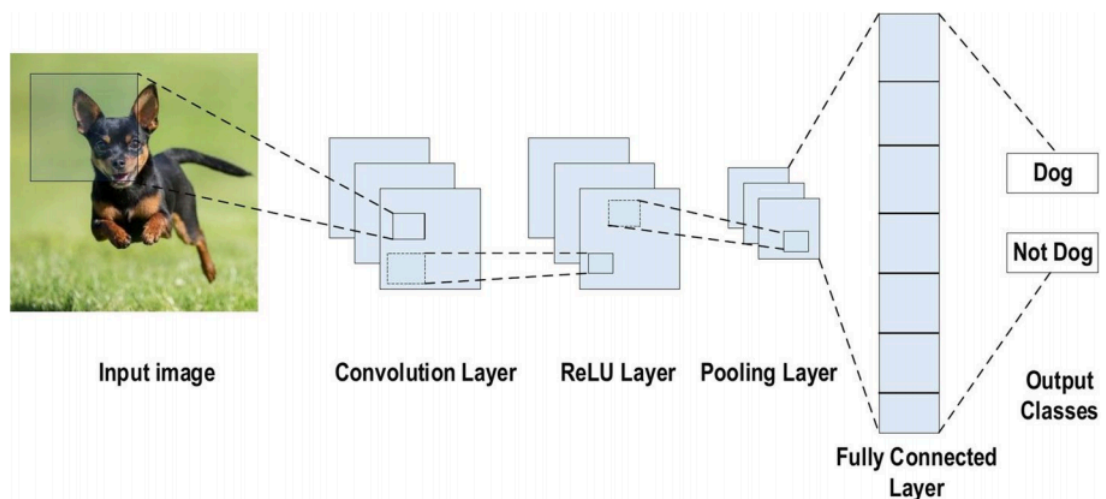


Рисунок 2.1. – Приклад архітектури ЗНМ для класифікації зображень

## 2.3 Переваги використання ЗНМ

Переваги використання ЗНМ над іншими традиційними нейронними мережами у галузі комп'ютерного зору:

1. Основною причиною розгляду ЗНМ є функція розподілу ваги, яка зменшує кількість параметрів мережі, що піддається навчанню, і, в свою чергу, допомагає мережі посилити генералізацію та уникнути перенавчання.
2. Одночасне вивчення шарів вилучення ознак та класифікаційних шарів призводить до того, що вихідні дані моделі є як високоорганізованими і сильно залежними від вилучених ознак.
3. Впровадження масштабної мережі набагато простіше з ЗНМ, ніж з іншими нейронними мережами.

## 2.4 Шари ЗНМ

Архітектура ЗНМ складається з ряду шарів.

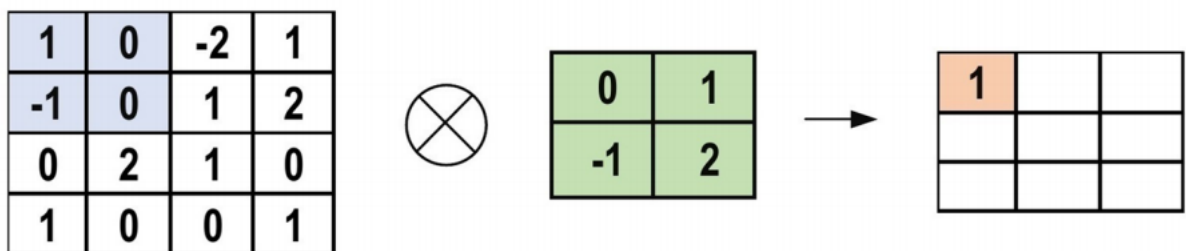
1. Згортковий шар: в архітектурі ЗНМ найбільш значущим компонентом є згортковий шар. Він складається з колекції згорткових фільтрів (ядер). Вхідне зображення, виражене у вигляді  $N$ -мірних метрик, згортається з цими фільтрами для створення вихідної карти ознак.

- ядро: сітка дискретних чисел або значень описує ядро. Кожне значення називається вагою ядра. Випадкові числа призначаються, щоб діяти як ваги ядра на початку навчального процесу ЗНМ. Крім того, існує кілька різних методів, що використовуються для ініціалізації ваг. Далі ці ваги коригуються в кожен епоху тренувань; таким чином, ядро вчиться виділяти важливі риси;



- згорткова операція: спочатку описаний формат вводу ЗНМ. Векторний формат є входом традиційної нейронної мережі, тоді як багатоканальне зображення - входом ЗНМ. Наприклад, одноканальний - це формат зображення в сірому масштабі, тоді як формат RGB-зображення є триканальним. Наприклад зображення  $4 \times 4$  в сірому масштабі з ядром, що ініціалізується випадковою вагою  $2 \times 2$ . Спочатку ядро ковзає по всьому зображенню по горизонталі та вертикалі. Крім того, визначається точковий добуток між вхідним зображенням і ядром, де їх відповідні значення множаться, а потім підсумовуються, створюючи єдине скалярне значення, обчислене одночасно. Потім весь процес повторюється, доки подальше ковзання неможливе. На рисунку 2.2 показано первинні розрахунки, виконані на кожному кроці. На цій представленні світло-зелений колір представляє ядро  $2 \times 2$ , тоді як світло-блакитний колір представляє область подібного розміру вхідного зображення. Обидва множаться; кінцевий результат після підсумовування отриманих значень (позначений світло-оранжевим кольором) являє собою значення входу на вихідну карту ознак.

### Step-1



### Step-2

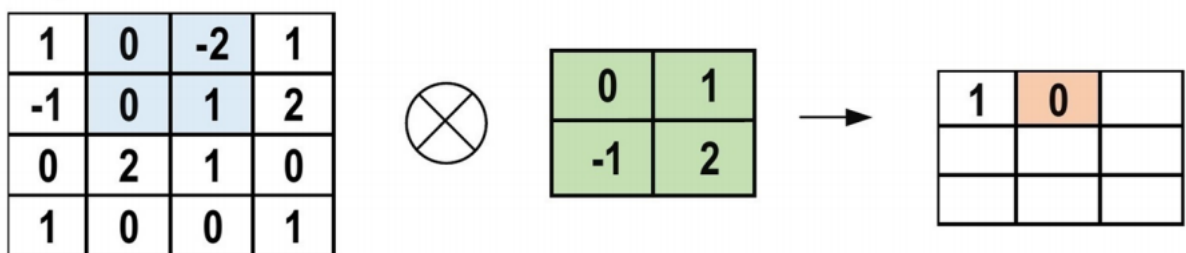


Рисунок 2.2 - Розрахунки, що виконуються на кожному кроці згорткового шару

Однак падінг не застосовується у попередньому прикладі, тоді як до ядра застосовується крок один (позначений для вибраного розміру кроку по всіх вертикальних або горизонтальних місцях). Можна використовувати інше значення кроку. Крім того, внаслідок збільшення значення кроку отримується функціональна карта менших розмірів.

З іншого боку, падінг дуже важливий для визначення інформації про розмір меж, що стосується вхідного зображення. Навпаки, прикордонні бічні ознаки рухаються дуже швидко. Застосовуючи падінг, розмір вхідного зображення збільшиться, і в свою чергу, збільшиться і розмір вихідної карти ознак.

Основні переваги згорткових шарів:

- розріджена підключеність: Кожен нейрон в повнозв'язному шарі з'єднується з усіма нейронами наступного шару. На відміну від цього, в ЗНМ доступні лише декілька ваг між двома сусідніми шарами. Таким чином, кількість необхідних ваг або з'єднань невелика, тоді як пам'ять, необхідна для зберігання цих ваг, також невелика; отже, такий підхід ефективний для пам'яті;

- спільне використання ваги: між будь-якими двома нейронами сусідніх шарів у ЗНМ немає розподілених ваг, оскільки цілі ваги працюють з одним і всіма пікселями вхідної матриці. Вивчення єдиної групи ваг за вхід значно зменшить необхідний час тренування та різні витрати, оскільки не потрібно вивчати додаткові ваги для кожного нейрона.

2. Шар пулінгу: основним завданням шару пулінгу є субдискретизація карт ознак. Ці карти генеруються після згорткових операцій. Такий підхід зменшує великі карти ознак, щоб створювати менші карти ознак. Одночасно він зберігає більшість домінуючої інформації (або особливостей) на кожному етапі етапі пулінгу. Подібно до згорткової операції, як крок, так і ядро спочатку призначаються за розміром перед тим, як буде виконана операція пулінгу. Кілька типів методів об'єднання доступні для використання в різних

шарах пулінгу. Ці методи включають пулінг дерев, середній пулінг, мінімальний пулінг, максимальний пулінг, загальний середній пулінг (ЗСП) та загальний максимальний пулінг. Найбільш звичними та часто використовуваними методами об'єднання є об'єднання max, min та ЗСП. Рисунок 2.3 ілюструє ці три операції пулінгу. Іноді загальна продуктивність ЗНМ в результаті знижується; це являє собою основний недолік шару пулінгу, оскільки цей шар допомагає ЗНМ визначити, чи є певна ознака на конкретному вхідному зображенні, але зосереджується виключно на визначенні правильного розташування цієї ознаки. Таким чином, модель ЗНМ упускає відповідну інформацію.

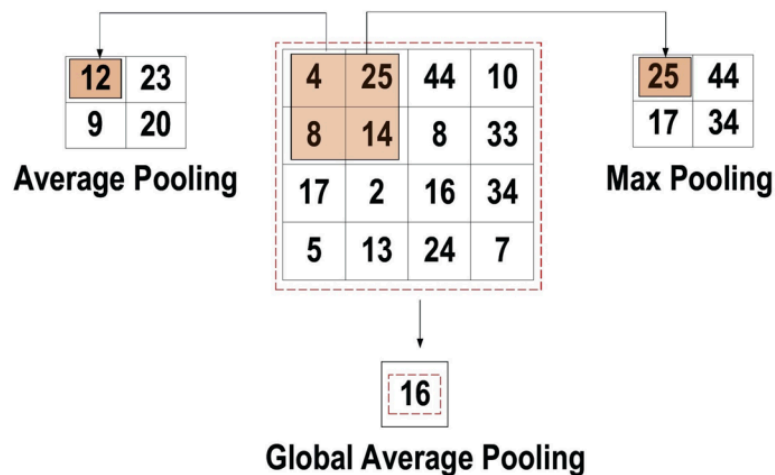


Рисунок 2.3 – Три типи операції пулінгу

3. Функція активації (нелінійність) Зіставлення вхідних даних на виході є основною функцією всіх типів функції активації у всіх типах нейронних мереж. Вхідне значення визначається шляхом обчислення зваженого підсумовування вхідного нейрона разом з його зміщенням (якщо воно є). Це означає, що функція активації приймає рішення щодо звільнення нейрона з посиленням на певний вхід шляхом створення відповідного виходу. Нелінійні активаційні шари використовуються після всіх шарів з вагами (шари, що навчаються, такі як шари повнозв'язні та згорткові шари) в архітектурі ЗНМ. Ця нелінійна ефективність активаційних шарів означає, що

відображення вхідного та вихідного шарів буде нелінійним; ці рівні дають ЗНМ можливість вивчати надскладні речі. Функція активації також повинна мати здатність диференціюватись, що є надзвичайно важливою особливістю, оскільки вона дозволяє використовувати зворотне розповсюдження помилок для навчання мережі. Наступні типи функцій активації найчастіше використовуються в ЗНМ та інших глибоких нейронних мережах:

- сигмоїд: На вхід цієї функції активації подаються дійсні числа, тоді як вихідний сигнал обмежений нулем та одиницею. Крива сигмоїдної функції має S-подібну форму і може бути представлена математично рівнянням. 2;

$$f(x)_{\text{sigm}} = \frac{1}{1+e^{-x}} \quad (2)$$

- Tanh: Це схоже на сигмоїдну функцію, оскільки її вхідні дані є дійсними числами, але вихідний сигнал обмежений від -1 до 1. Його математичне подання представлено в рівнянні. 3;

$$f(x)_{\text{tanh}} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3)$$

- ReLU: Це загальноновживана функція в контексті ЗНМ. Вона перетворює цілі значення вхідних даних у додатні числа. Менше обчислювальне навантаження є основною перевагою ReLU перед іншими. Його математичне подання представлено в рівнянні. 4.

$$f(x)_{\text{ReLU}} = \max(0, x) \quad (4)$$

Іноді під час використання ReLU може виникнути кілька значущих проблем. Наприклад, у алгоритмі зворотного розповсюдження помилок з більшим градієнтом, що проходить через нього. Передача цього градієнта в функції ReLU оновить ваги таким чином, що нейрон, ще раз не активується.

Цю проблему називають “Помираючим ReLU”. Для вирішення таких питань існують деякі альтернативи ReLU:

- Leaky ReLU: Замість зменшення масштабу негативних входів ReLU, ця функція активації гарантує, що ці входи ніколи не ігноруються. Він застосовується для вирішення проблеми вмирання ReLU. Leaky ReLU можна представити математично, як у рівнянні. 5;

$$f(x)_{LeakyReLU} = \begin{cases} x, \text{ якщо } x > 0 \\ mx, x \leq 0 \end{cases} \quad (5)$$

Коефіцієнт Leaky позначається  $m$ . Зазвичай його встановлюють дуже малого значення, наприклад 0,001.

- Noisy ReLU: функція використовує гауссовий розподіл, щоб зробити ReLU шумним. Це можна представити математично, як у рівнянні. 6;

$$f(x)_{NoisyReLU} = \max(x + Y), Y \sim N(0, \sigma(x)) \quad (6)$$

- Parametric Linear Units: Це те саме, що Leaky ReLU. Основна відмінність полягає в тому, що коефіцієнт витоку в цій функції оновлюється в процесі навчання моделі. Parametric Linear Units можна представити математично, як у рівнянні. 7.

$$f(x)_{ParametricLinear} = \begin{cases} x, \text{ якщо } x > 0 \\ ax, x \leq 0 \end{cases} \quad (7)$$

Параметр, що навчається –  $a$ .

4. Повнозв'язний шар: зазвичай цей шар знаходиться в кінці кожної ЗНМ. Усередині цього шару кожен нейрон зв'язаний з усіма нейронами попереднього шару. Він використовується як класифікатор ЗНМ. Вхід цього шару надходить від останнього пулінгу або згорткового шару. Цей вхід має

форму вектора, який створюється з карт ознак після шару flatten. Вихід повнозв'язного шару - остаточний вихід CNN, як показано на рисунку 2.4

## 2.5 Функції втрат

Остаточна класифікація досягається з вихідного рівня, який представляє останній шар архітектури ЗНМ. Деякі функції втрат використовуються у вихідному рівні для обчислення передбачуваної помилки, створеної для навчальних зразків у моделі ЗНМ. Ця помилка показує різницю між фактичним результатом та прогнозованим. Далі його буде оптимізовано за допомогою навчального процесу ЗНМ. Функцією втрат для обчислення помилки використовується два параметри. Очікуваний вихід ЗНМ є першим параметром. Фактичний результат є другим параметром. Кілька типів функцій втрат застосовуються у різних типах задач.

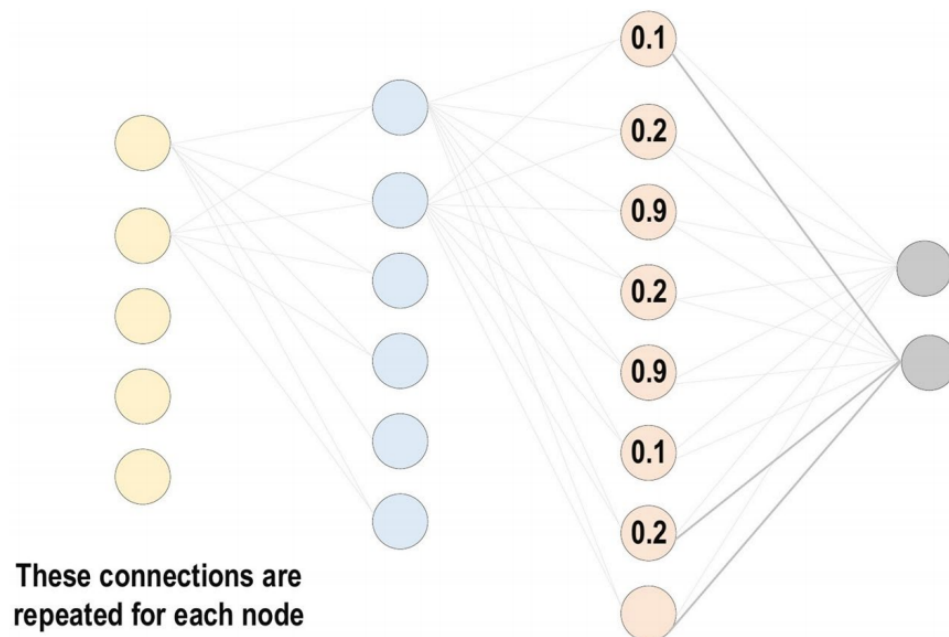


Рисунок 2.4 – Повнозв'язний шар

1. Функція перехресної ентропії або Softmax Loss: Ця функція зазвичай використовується для вимірювання продуктивності моделі ЗНМ. Результатом її є ймовірність  $p \in \{0,1\}$ . Крім того, його зазвичай використовують як заміну функції втрат квадратичної помилки у задачах класифікації. У вихідному рівні вона використовує активацію softmax для генерації вихідних даних у межах розподілу ймовірностей. Математичне представлення ймовірності класу виходу дорівнює рівнянню 8.

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^N e_k^a} \quad (8)$$

де  $e^{a_i}$  - ненормований вихід з попереднього шару, тоді як  $N$  - кількість нейронів у вихідному шарі. Математичне представлення функції перехресної ентропії є рівняння 9.

$$H(p, y) = - \sum_i y_i \log(p_i), i \in [1, N] \quad (9)$$

2. Функція втрат Евкліда: Ця функція широко використовується в задачах регресії. Крім того, це також так звана середньоквадратична помилка. Математичне представлення функції евклідових втрат є рівняння 10.

$$H(p, y) = \frac{1}{2N} \sum_{i=1}^N (p_i - y_i)^2 \quad (10)$$

3. Функція втрат Hinge: Ця функція зазвичай використовується у задачах, пов'язаних з двійковою класифікацією. Ця проблема стосується класифікації на основі максимальної маржі; це в основному важливо для SVM. Його математична формула - рівняння 11.

$$H(p, y) = \sum_{i=1}^N \max(0, m - (2y_i - 1)p_i) \quad (11)$$

Зазвичай  $m$  встановлюється рівним 1. Передбачуваний вихід позначається як  $p_i$ , тоді як бажаний вихідний результат позначається як  $y_i$ .

## 2.6 Регуляризація ЗНМ

Для моделей ЗНМ перенавчання є основною проблемою, пов'язаною з отриманням коректних результатів. Модель вважається перенавченою в тих випадках, коли модель особливо добре працює на навчальних даних та працює погано на тестових даних. Недонавчена модель - це навпаки; цей випадок трапляється, коли модель не успішна на навчальних даних. Модель називається "щойно навчена", якщо вона добре працює як на тренувальних, так і на тестувальних даних. Ці три типи зображено на рисунку 2.5.

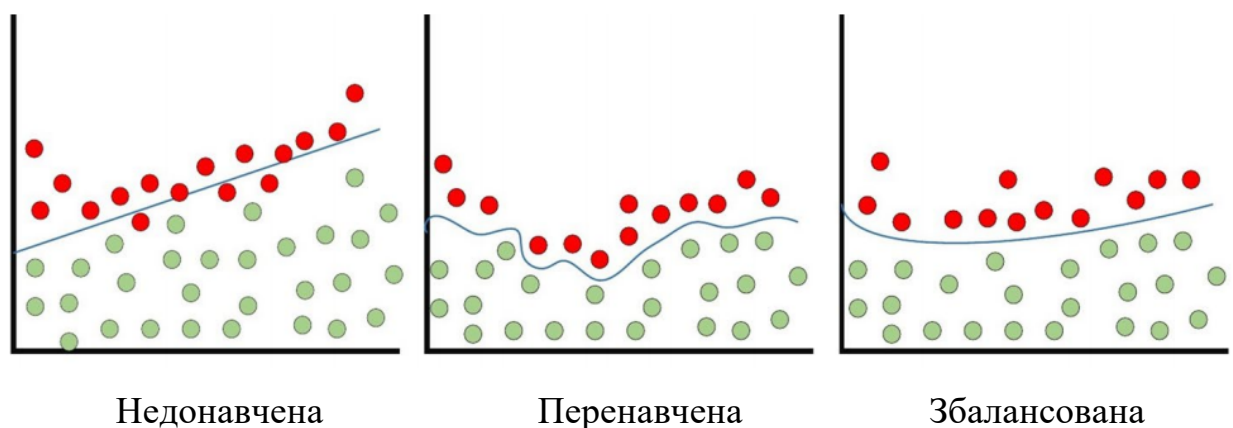


Рисунок 2.5 – Проблеми перетренованої та недотренованої моделі

1. Dropout: Це широко застосовувана техніка генералізації. Протягом кожної навчальної епохи нейрони випадково викидаються. Роблячи це, потужність вибору ознак розподіляється порівну по всій групі нейронів, а також змушує модель вивчати різні незалежні особливості. Під час тренувального процесу нейрон, що випав, не буде частиною зворотного або прямого поширення



похибки. Навпаки, повномасштабна мережа використовується для прогнозування під час процесу тестування.

2. Drop-Weights: Цей метод дуже схожий на дропаут. У кожну тренувальну епоху зв'язки між нейронами (вагами) відпадають, а не викидають нейрони; це являє собою єдину відмінність між скиданням ваги та відсівом.

3. Збільшення даних: Навчання моделі на значній кількості даних - це найпростіший спосіб уникнути перенавчання. Для цього використовується збільшення даних. Для штучного збільшення розміру набору навчальних матеріалів використовується кілька методів.

4. Batch Normalization: Цей метод забезпечує виконання вихідних активацій [18]. Цей показник відповідає одиничному розподілу Гауса. Віднімання середнього та ділення на стандартне відхилення нормалізує вихід на кожному шарі. Хоча це можна розглядати як завдання попередньої обробки на кожному рівні мережі, можна також диференціювати та інтегрувати його з іншими мережами. Крім того, він застосовується для зменшення "внутрішнього зсуву коваріації" шарів активації. У кожному шарі зміна розподілу активації визначає внутрішній зсув коваріації. Цей зсув стає дуже високим завдяки постійному оновленню ваги під час тренувань, яке може статися, якщо зразки даних тренувань збираються з численних різних джерел (наприклад, денні та нічні зображення). Таким чином, модель буде витратити додатковий час на конвергенцію, і в свою чергу, час, необхідний для навчання, також збільшиться. Для вирішення цієї проблеми в архітектурі ЗНМ застосовується шар, що відповідає операції батч нормалізації. Переваги використання батч нормалізації такі:

- це запобігає виникненню проблеми зникнення градієнта;
- він може ефективно контролювати ініціалізацію низької ваги;
- це значно скорочує час, необхідний для конвергенції мережі (для великомасштабних наборів даних це буде надзвичайно корисно);
- він намагається зменшити залежність від тренувань гіперпараметрами;

- шанси на перенавчання зменшуються, оскільки це має незначний вплив на регуляризацію.

## 2.7 Вибір оптимізатора

Дві основні проблеми постають в процесі навчання: першим питанням є вибір алгоритму навчання (оптимізатор), в той час як другим питанням є використання багатьох удосконалень (таких як AdaDelta, Adagrad та momentum) разом з алгоритмом навчання для покращення виходу.

Функції втрат, які базуються на численних параметрах, що підлягають вивченню (наприклад, відхилення, ваги, тощо) або мінімізація помилок (варіація між фактичним та передбачуваним результатом), є основною метою всіх контрольованих алгоритмів навчання. Часто для ЗНМ використовують методи градієнтного навчання. Параметри мережі повинні завжди оновлюватися протягом усіх навчальних епох, тоді як мережа також повинна шукати локально оптимізовану відповідь у всі навчальні епохи, щоб мінімізувати помилку.

Швидкість навчання визначається як розмір кроку оновлення параметра. Епоха тренувань являє собою повне повторення оновлення параметрів, що включає повний набір навчальних даних одночасно. Потрібно підбирати швидкість навчання, щоб вона не впливала на процес навчання недосконало, хоча це гіперпараметр.

Градiєнтний спуск або градієнтний алгоритм навчання: Щоб мінімізувати помилку навчання, цей алгоритм багаторазово оновлює параметри мережі протягом кожної епохи навчання. Щоб правильно оновити параметри, йому потрібно обчислити градієнт цільової функції (нахил), застосовуючи похідну першого порядку відносно параметрів мережі. Далі параметр оновлюється у зворотному напрямку градієнта, щоб зменшити

помилку. Процес оновлення параметрів виконується за допомогою алгоритму зворотного розповсюдження похибки, в якому градієнт на кожному нейроні повертається до всіх нейронів попереднього шару. Математичне подання цієї операції представлено у рівнянні 12.

$$w_{ijt} = w_{ijt-1} - \Delta w_{ijt}, \Delta w_{ijt} = \eta * \frac{\partial E}{\partial w_{ij}} \quad (12)$$

Кінцева вага в поточній епосі тренувань позначається  $w_{ijt}$ , тоді як вага в попередню  $(t - 1)$  епоху тренувань -  $w_{ijt-1}$ . Швидкість навчання дорівнює  $\eta$ , а похибка передбачення -  $E$ .

Є різні альтернативи алгоритму навчання на основі градієнта, які зазвичай використовуються. Визначають наступні алгоритми:

1. Батч градієнтний спуск (BGD): Під час виконання цієї методики [19] параметри мережі оновлюються лише на один раз, враховуючи всі набори навчальних даних мережі. Він обчислює градієнт всього навчального набору і згодом використовує цей градієнт для оновлення параметрів. Для малого набору даних модель ЗНМ сходиться швидше і створює надзвичайно стабільний градієнт за допомогою цього методу. Оскільки параметри змінюються лише один раз за кожну епоху навчання, це вимагає значної кількості ресурсів. На відміну від цього, для великого набору навчальних даних необхідний додатковий час, і він може сходитися до локального оптимуму (для неопуклих випадків).

2. Стохастичний градієнтний спуск (SGD): параметри оновлюються на кожному навчальному зразку [20]. Бажано доволіно підбирати навчальні зразки в кожну епоху до початку навчання. Для великого набору навчальних даних ця методика одночасно ефективніша для пам'яті і набагато швидша, ніж батч градієнтний спуск. Однак, оскільки він часто оновлюється, він робить надзвичайно зашумлені кроки у напрямку відповіді, що, в свою чергу, призводить до того, що поведінка конвергенції стає дуже нестабільною.

3. Міні батч градієнтний спуск: У цьому підході навчальні зразки розділені на кілька міні-партій, в яких кожна міні-партія може розглядатися як малогабаритна колекція зразків без пересічень між ними [21]. Далі, оновлення параметрів виконується після обчислення градієнта на кожній міні-партії. Перевага цього методу полягає в поєднанні переваг як методів BGD, так і SGD. Таким чином, він має стійку конвергенцію, більшу обчислювальну ефективність та більш ефективне використання пам'яті.

4. Momentum: Для нейронних мереж цей метод застосовується в цільовій функції. Це підвищує як точність, так і швидкість тренування, підсумовуючи обчислюваний градієнт на попередньому етапі тренування, який зважується через коефіцієнт (відомий як коефіцієнт імпульсу). Однак, він просто застряє у локальному, а не глобальному мінімумі. Це є головним недоліком градієнтних алгоритмів навчання. Разом з алгоритмом навчання для вирішення цього питання використовується імпульс, який можна виразити математично, як у рівнянні 13.

$$w_{ijt} = (\eta * \frac{\partial E}{\partial w_{ij}}) + (\lambda * \Delta w_{ijt-1}) \quad (13)$$

Приріст ваги в поточній t-й епосі тренувань позначається як  $\Delta w_{ijt-1}$ , тоді як  $\eta$  - швидкість навчання, а приріст ваги у попередню  $(t - 1)$  епоху тренувань. Значення коефіцієнта імпульсу підтримується в межах від 0 до 1; у свою чергу, розмір кроку оновлення ваги збільшується у напрямку до мінімуму, щоб мінімізувати помилку. Оскільки значення коефіцієнта імпульсу стає дуже низьким, модель втрачає здатність уникати локального мінімуму. Навпаки, коли значення коефіцієнта імпульсу стає високим, модель розвиває здатність набагато швидше сходиться.

5. Адаптивна оцінка моменту (Адам): Це ще одна техніка оптимізації або алгоритм навчання, яка широко використовується. Адам [22] представляє останні тенденції в оптимізації глибокого навчання. Це представлено

матрицею Гесса, яка використовує похідну другого порядку. Адам - це стратегія навчання, розроблена спеціально для навчання глибоких нейронних мереж. Більша ефективність пам'яті та менша обчислювальна потужність - дві переваги Адама. Він об'єднує плюси як Momentum, так і RMSprop. Він використовує квадратичні градієнти для масштабування швидкості навчання як RMSprop, і це схоже на Momentum за допомогою ковзного середнього градієнта. Рівняння Адама математично представлено у вигляді рівняння 14.

$$w_{ijt} = w_{ijt-1} - \frac{\eta}{\sqrt{E[\delta^2]^t + \epsilon}} * E[\delta^2]^t \quad (14)$$

## 2.8 Розробка алгоритмів (зворотне розповсюдження помилки)

Почнемо з позначення, яке однозначно позначає ваги в мережі. Позначимо  $w_{ij}^h$  як вагу для з'єднання від  $i$ -го входу або (нейрона на  $(h - 1)$ ) до  $j$ -го нейрона в  $h$ -му шарі. Отже, на рисунку 2.6 показано вагу з'єднання з нейроном у першому шарі з іншим нейроном у наступному шарі в мережі.

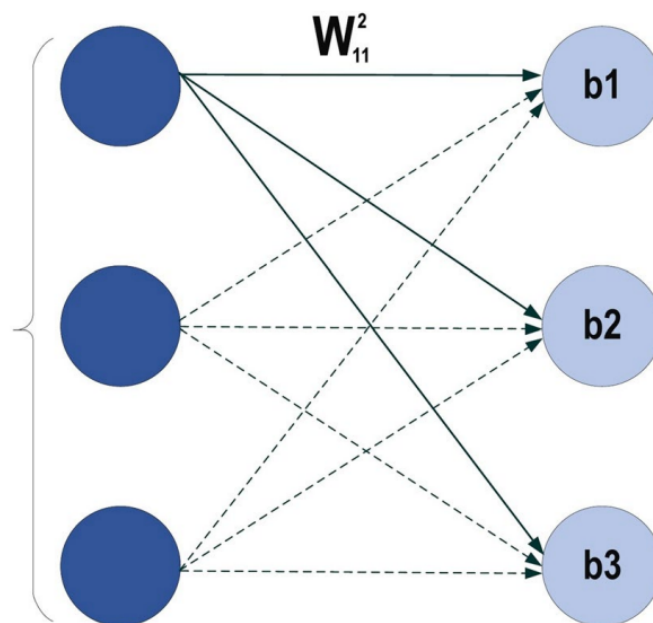


Рисунок 2.6 – Приклад нейронної мережі

Де  $w_{11}^2$  представляє вагу від першого нейрона в першому шарі до першого нейрона у другому шарі, виходячи з того, що друга вага для того самого нейрона буде  $w_{21}^2$ , що означає, що вага походить від другого нейрона в попереднього шару до першого шару наступного шару, який є другим у цій мережі. Оскільки зміщення не є зв'язком між нейронами для шарів, тому з ним легко впоратися, кожен нейрон повинен мати своє власне зміщення, у деяких мережах кожен шар має певне зміщення. З наведеної мережі видно, що кожен шар має своє власне зміщення. Кожна мережа має такі параметри, як номер шару в мережі, кількість нейронів у кожному шарі, кількість ваг (зв'язків) між шарами, кількість зв'язків може бути легко визначена на основі кількості нейронів. Наприклад, у кожному шарі, якщо є десять входів, які повністю з'єднуються з двома нейронами в наступному шарі, кількість з'єднань між ними дорівнює ( $10 * 2 = 20$ ), як визначається помилка, і вага оновлюється, ми уявимо, що у нашій нейронній мережі є два шари,

$$\text{похибка} = \frac{1}{2}(d_i - y_j)^2$$

де  $d$  - мітка індувіального входу  $i$ , а  $y$  - вихід того самого окремого входу. Зворотне розповсюдження помилки – це про змінення ваги та зміщення в мережі на основі змін функції витрат (Похибки). Зрештою, це означає обчислення часткових похідних  $\partial E / \partial w_{ij}^h$  та  $\partial E / \partial b_j^h$ . Але для їх обчислення вводиться локальна змінна  $\delta_j^1$ , яка називається локальною помилкою в  $j$ -му нейроні  $h$ -го шару. Виходячи з цієї локальної помилки, зворотне розповсюдження помилки дасть процедуру для обчислення  $\partial E / \partial w_{ij}^h$  та  $\partial E / \partial b_j^h$ , для визначення помилки та оновленні ваги, уявимо, що в нейронній мережі є два шари що показано на рисунку 2.7.

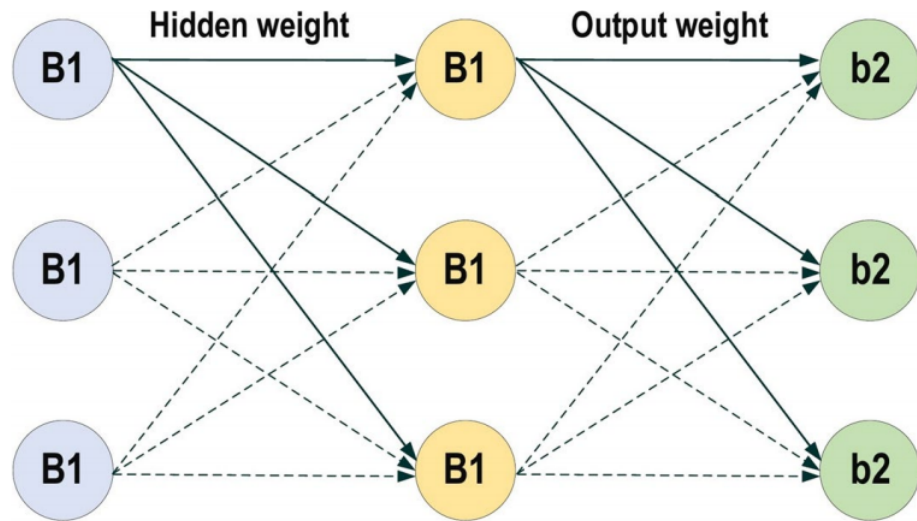


Рисунок 2.7 - Функції активації нейронів

Вихідна помилка для  $\delta_j^1$ , кожна  $1 = 1:L$ , де  $L$  не нейрон на виході

$$\delta_j^1(k) = (-1)e(k)\vartheta'(v_j(k)) \quad (15)$$

де  $e(k)$  - похибка епохи  $k$ , як показано в рівнянні (2) та  $e(k)\vartheta'$  є похідною функції активації для  $v_j$  на виході. Зворотне розповсюдження помилки на всіх інших рівнях, крім виходу

$$\delta_j^h(k) = \vartheta'(v_j(k)) \sum_{l=1}^L \delta_l^1(k) w_{jl}^{h+1}(k) \quad (16)$$

де  $\delta_j^1(k)$  - вихідна помилка, а  $w_{jl}^{h+1}(k)$  - вага після шару, де потрібно отримати помилку. Після пошуку помилки на кожному нейроні в кожному шарі, тепер ми можемо оновити вагу у кожному шарі на основі рівнянь. (15) та (16).

## 2.9 Аналіз існуючих робіт

Штучний інтелект та глибоке навчання в основному використовуються в техніках обробки зображень для сегментації, ідентифікації та класифікації МРТ-зображень, а також використовуються для класифікації та виявлення пухлин головного мозку. Так багато робіт вже зроблено щодо класифікації та сегментації зображень МРТ головного мозку. Sheikh Basheera et al., [23] запропонували метод класифікації пухлин головного мозку, де пухлина спочатку сегментується з МРТ-зображення, а потім сегментована частина витягується за допомогою попередньо навченої згорткової нейронної мережі з використанням стохастичного градієнтного спуску. Мухаммед Саджад та ін. [24] запропонували класифікувати багатоступеневі пухлини, застосовуючи методику збільшення даних МРТ зображень, а потім налаштовуючи їх за допомогою попередньо навченої моделі CNG VGG-19. Carlo, Ricciardi et al., [25] представили підхід до класифікації МРТ пухлини аденоми гіпофіза за допомогою мультиноміальної логістичної регресії та k-найближчих алгоритмів. Такий підхід досяг точності 83% для мультиноміальної логістичної регресії та 92% для k-найближчого сусіда з кривою AUC 98,4%. Khwaldeh, Saed et al., [26] представив основу для класифікації зображень МРТ головного мозку на здорові та нездорові, а також систему оцінки класифікації нездорових зображень мозку на низькі та високі оцінки, модифікуючи модель Alex-Net CNN, яка показала 91 % точності. Nyoman Abiniwanda et al., [27] навчили згорткову нейронну мережу для класифікації трьох специфічних класів пухлин головного мозку, а саме: Менінгіоми, Гліоми та Гіпофіза, які досягли 98,51% точності на тренувальній вибірці та 84,19% точності на перевіірочній. Сунанда Дас [28] також навчили модель CNN із технікою обробки зображень для виявлення різних типів пухлини



мозку та досягли точності 94,39% із середньою точністю 93,33%. Ромео, Валерія [29] представили підхід до радіомічного машинного навчання для прогнозування ступенів пухлини та вузлового статусу на КТ сканування первинних уражень пухлини та отримали найвищу точність у 92,9% від Наїва Байєса та k-найближчого сусіда. Мухаммед Тало [30] використовували попередньо навчену модель CNN ResNet34 підхід до навчання передачі разом із збільшенням даних для класифікації нормальних та аномальних зображень МРТ головного мозку та отримали 100% точність. Аршія Реман та співавт. [31] використовували три різні попередньо навчені моделі CNN (VGG16, AlexNet та GoogleNet) для класифікації пухлин головного мозку на гіпофіз, гліому та менінгіому. Під час цього підходу до навчання, VGG16 набуває точності, яка становить 98,67%. Ахмет інар [32] модифікували попередньо навчену модель ResNet50 CNN, видаливши останні 5 шарів та додавши замість неї 8 нових шарів та порівняли її точність з іншими попередньо навченими моделями, такими як GoogleNet, AlexNet, ResNet50. Оновлена модель ResNet50 показала ефективні результати, досягнувши точності 97,2%.

Недоступність до маркованих даних є однією з основних перешкод на шляху використання глибокого навчання в медичну сферу та охорону здоров'я. Оскільки нещодавня розробка програм глибокого навчання в інших галузях показала, що чим більшими будуть дані, тим кращими будуть результати. Сегментація даних та збільшення даних здійснюються з використанням глибокого навчання у згаданій літературі, а також використовувались різні попередньо навчені моделі CNN, що використовують підхід трансферного навчання для класифікації пухлин головного мозку. Попередньо підготовленими моделями, які в основному використовуються є VGG-16, ResNet-50 та Inception-v3, які попередньо навчені на масовій кількості наборів даних, таких як ImageNet. А для радіологічних досліджень та експериментів потрібно виконати точну настройку, замороживши шари, щоб зменшити параметри, якщо набір даних

малий, ми також повинні замінити повністю зв'язані шари відповідно до міток набору даних. Робота даних систем вимагає великої обчислювальної потужності та спеціалізованих процесорів (графічні процесори), що вимагає великих витрат, і ще одним недоліком трансферного навчання є те, що розмір вхідного зображення є фіксованим, тому нам доводиться коригувати наші зображення відповідно до розміру вхідного сигналу попередньо навченої моделі. Тому було обрано не дуже великий набір зображень МРТ головного мозку. Було застосовано техніку збільшення даних разом із технікою обробки зображень на цих МРТ-зображеннях, а потім навчено модель CNN з нуля на цих доповнених попередньо оброблених даних зображення, щоб визначити, чи містить зображення МРТ пухлину чи ні. Також було порівняно продуктивність діагностики та обчислювальне споживання моделі з моделями VGG-16 та ResNet.

## 2.10 Висновки до розділу 2

У даному розділі було розглянуто згорткові нейронні мережі та їх побудова. Було зроблено аналіз та наведено переваги використання згорткових нейронних мереж для поставленої задачі. Також було розглянуто шари ЗНМ, функції втрат, вибір оптимізатора та алгоритм навчання.

## РОЗДІЛ 3 ОПИС АРХІТЕКТУРИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ

### 3.1 Запропонований підхід

У цьому дослідженні було застосовано методи обробки зображень та збільшення даних на наборі даних із 2530 зображень МРТ головного мозку [33]. Було навчено їх за допомогою моделі CNN із 8 згорткових шарів та порівняно точність запропонованої моделі CNN із попередньо підготовленими моделями VGG-16, ResNet-50 та Inception-v3, використовуючи підхід трансферного навчання. Набір даних включає 1550 зображень злоякісного раку та 980 доброякісних неракових пухлин. Було розділено набір даних на 3 окремі сегменти для навчання, перевірки та тестування. Тренувальні дані призначені для навчання моделі, дані для перевірки - це вибірккові дані для оцінки моделі та налаштування параметрів моделі. Тестувальні дані призначені для остаточного оцінювання нашої моделі. Запропонований спосіб складається з різних фаз. Огляд запропонованої методології наведено на рисунку 3.1.



Рисунок 3.1 Огляд запропонованої методології

### 3.2 Обробка зображень

По-перше, було обрізано темні краї зображень і взято лише ділянку мозку із зображень МРТ, використовуючи техніку відкритого коду Computer Vision (CV) Canny Edge Detection [34]. Canny Edge Detection - це багатофазний алгоритм, який використовується для ідентифікації країв об'єкта на зображенні. На рисунку 3.2, краї реального МРТ головного мозку показані за допомогою техніки «хитрого» виявлення країв, і обрізана лише частина зображення мозку.

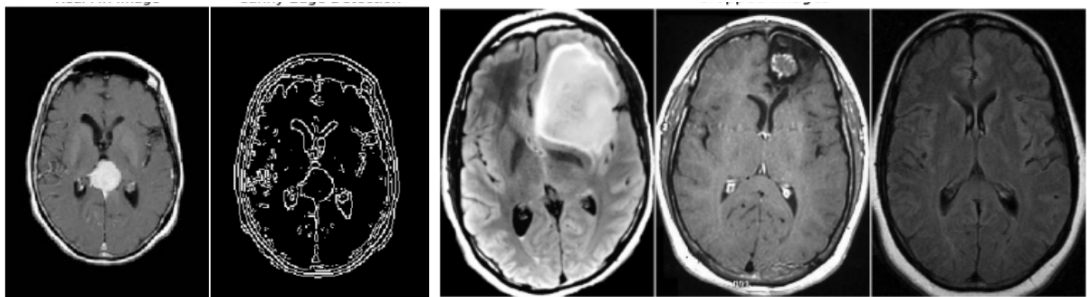


Рисунок 3.2 - Пошук країв та обрізання частини зображення мозку

### 3.3 Аугментація даних

Аугментація (збільшення) даних - це стратегія штучного збільшення кількості та складності існуючих даних [35]. Для навчання глибокої нейронної мережі потрібен великий обсяг даних для точної настройки параметрів. Було застосовано техніку збільшення даних [36] до навчального набору даних, додавши модифікації до зображень, вносячи незначні зміни,

такі як гортання, обертання та яскравість. Це збільшить обсяг навчальних даних, і модель буде розглядати кожну з цих невеликих змін як окреме зображення, і це дасть змогу запропонованій моделі краще вчитися та краще показувати себе з даними, яких вона не бачила. На малюнку 3.3 показано численні доповнені зображення з одного зображення.

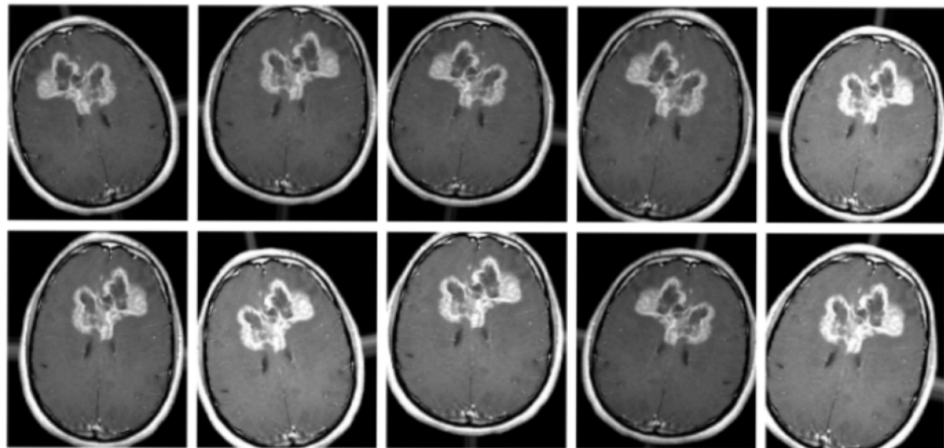


Рисунок 3.3 – Застосування аугментації даних

### 3.4 Опис запропонованої моделі згорткової нейронної мережі

Було запропоновано наступну модель CNN, було застосовано аугментовані дані зображення МРТ вхідного розміру  $224 \times 224$ , що мають кольорові канали RGB із розміром партії(батч) 32 у модлі CNN. Спочатку було додано один згортковий шар з 16 фільтрів із розміром фільтра  $3 \times 3$ . Причиною розміщення невеликої кількості фільтрів, а саме 16 є виявлення країв, кутів та ліній. А потім на нього було додано шар об'єднання(пулінг) максимуму з фільтром  $2 \times 2$ , щоб отримати максимальну підсумок цього зображення, потім ми збільшили кількість згорткових шарів та кількість фільтрів до 32, 64 та 128, маючи той самий фільтр розміром  $3 \times 3$ . Це поєднує ці невеликі патерни, оскільки кількість фільтрів збільшується, і знаходить

більші патерни, такі як коло, квадрат тощо. Після цього було застосовано шари пулінгу максимум після згорткових шарів, щоб отримати максимум від них. Наступним кроком було застосування повнозв'язного шару із 256 нейронів разом із вихідним шаром softmax, який обчислює оцінку ймовірності для кожного класу та класифікує кінцеві рішення про те, що або вхідне зображення МРТ містить рак, або не містить рак, так чи ні. На рисунку 3.4 показано схему запропонованої архітектури CNN.

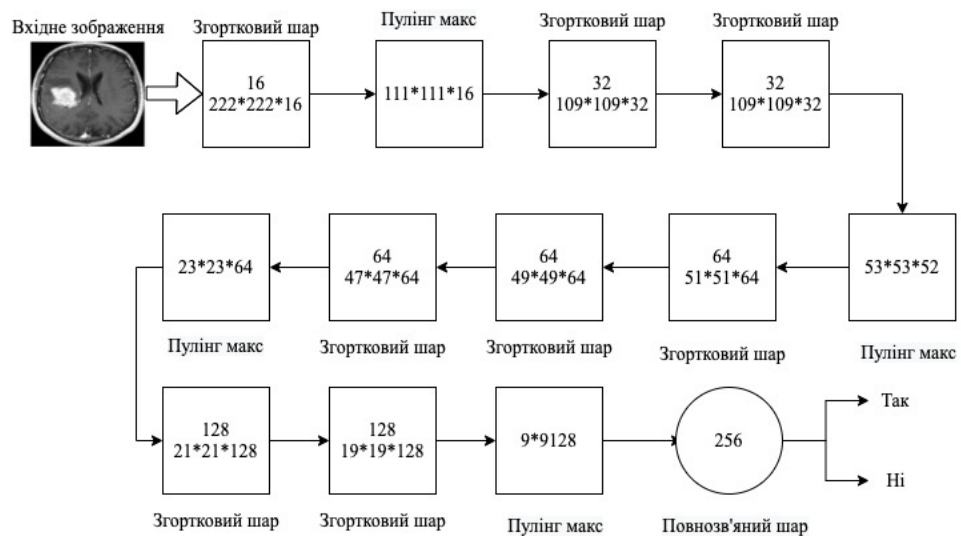


Рисунок 3.4 – Архітектура запропонованої ЗНМ

Було застосовано функцію активації Rectified Linear Unit (ReLU) у кожному згортковому шарі. Функція активації перетворює зважену вхідну суму на вихід цього вузла [37]. ReLU часто використовується в прихованих шарах згорткової нейронної мережі. Математично ReLU може бути представлена наступним чином

$$f(z) = \max(0, z)$$

де  $z$  - вхідний сигнал, коли  $z$  від'ємний або дорівнює 0, він перетворює від'ємний вхід на 0. Коли вхідне значення більше 0, вихідний результат буде 1. Таким чином, похідна від ReLU буде

$$f'(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

Отже, якщо вхідне значення 0, то цей нейрон є мертвим нейроном у функції ReLU, і він не буде активований.

### 3.5 Функція втрат

У машинному навчанні використовується функція втрат для обчислення помилки між передбаченими алгоритмічними значеннями та справжніми значеннями. Після цього ця помилка мінімізується за допомогою будь-якого методу оптимізації. У даному випадку було використано функцію перехресної ентропії (Cross Entropy) [38]. Оскільки робиться двійкова класифікація МРТ-зображень, буде використовуватися двійкова перехресна ентропія. У двійковій перехресній ентропії частота помилок обчислюється від 0 до 1. Математично вона представлена як

$$J(z) = [y \log P(y) + (1 - y) \log(1 - P(y))]$$

де  $y$  - фактичні результати, а  $P(y)$  - прогнозовані результати. Отже, коли фактичні результати  $y$  становитимуть 0, тоді перший доданок буде нулем, оскільки  $y$  множиться з  $\log$ . І коли  $y$  буде 1, тоді другий доданок буде нулем ( $1-y$ ) буде нулем, і він множиться з  $\log$ . І якщо  $y = P(y)$ , тоді  $J(y)$  (Loss) буде нульовим.

### 3.6 Оптимізація

У глибоких нейронних мережах застосовуються різні методи оптимізації, регулюючи такі параметри, як ваги та швидкість навчання, щоб зменшити втрати. Було вирішено використовувати оптимізатор адаптивного оцінювання моменту (Адама) [39]. Оптимізатор Адама - це поєднання RMSprop та стохастичного градієнтного спуску з імпульсом (momentum).

Метод стохастичного градієнтного спуску запропонований Дідеріком Кінгмою [40]. У простому стохастичному градієнтному спуску беруть похідну ваг,  $dW$  та похідну Bias,  $db$  для кожної епохи. І помножається на швидкість навчання.

$$\begin{aligned} W &= W - \eta \times dW, \\ b &= b - \eta \times db, \end{aligned}$$

Хоча стохастичний градієнтний спуск з імпульсом  $V$  є рухомих середнім значенням градієнтів, тут є рухоме середнє значення від 0 до 1, коли обчислюється  $dW$  і  $db$  для поточного батчу.

$$\begin{aligned} V_{dw} &= \beta \times V_{dw} + (1 - \beta) \times dW, \\ V_{db} &= \beta \times V_{db} + (1 - \beta) \times db, \end{aligned}$$

Подібним чином, Root Mean Squared Prop - це адаптивна методологія швидкості навчання, представлена Geoff, Hinton [41]. У RMSProp беремо експоненціально середній квадрат градієнтів, що рухається. Отже, у RMSProp,

$$\begin{aligned} S_{dw} &= \beta \times S_{dw} + (1 - \beta) \times dW^2, \\ S_{db} &= \beta \times S_{db} + (1 - \beta) \times db^2, \end{aligned}$$



де  $\beta$  - це гіперпараметр, який контролює експоненційно зважене середнє. Отже, поєднуючи особливості середньозваженого середнього за минулі градієнти та середньозваженого середнього квадрату минулих градієнтів, впроваджується техніка оптимізації Адама, отже

$$\begin{aligned} W &= W - \eta \times (V_{dw} / \sqrt{S_{dw} + \epsilon}), \\ b &= b - \eta \times (V_{db} / \sqrt{S_{db} + \epsilon}), \end{aligned}$$

де  $\epsilon$  (Епсілон) - це маленьке число, яке запобігає діленню на нуль (Епсілон =  $10^{-8}$ ), а  $\eta$  - швидкість навчання з різним діапазоном значень.

### 3.7 Трансферне навчання

У глибокому навчанні іноді використовується підхід трансферного навчання, коли замість створення нової моделі CNN для класифікації зображень використовується попередньо навчена модель CNN, яка вже навчена на величезному наборі даних, наприклад ImageNet, та використовується повторно. Замість того, щоб розпочинати навчальний процес з нуля, трансферне навчання використовує попереднє навчання.

#### 3.7.1 VGG-16

Було використовували попередньо навчену модель згорткової нейронної мережі VGG-16, яка налаштована заморожуванням деяких шарів,

щоб уникнути перенавчання, оскільки набір даних не дуже великий. VGG16 - це модель CNN з 16 згорткових шарів, запропонована у 2014 р. Карен Сімонян та Ендрю Ціссерман [42]. Форма вхідного зображення мережі становить  $224 \times 224 \times 3$ . Вона включає 16 шарів згортки з фіксованим розміром фільтра  $3 \times 3$  і 5 шарів пулінгу максимум з розміром  $2 \times 2$ . І у кінці 2 повнозв'язних шари з вихідним шаром softmax. Модель VGG-16 - це велика мережа, приблизно з 138 мільйонами параметрів. Вона складається з великої кількості згорткових шарів для створення глибокої нейронної мережі, які покращують здатність вивчати приховані ознаки. На рисунку 3.5 показано архітектуру мережі VGG-16.

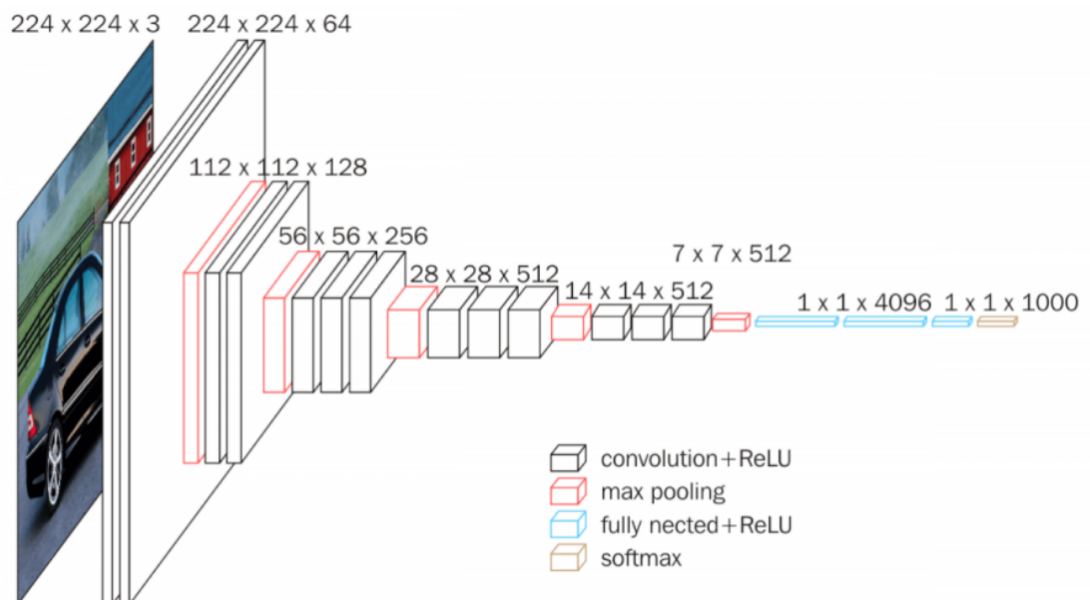


Рисунок 3.5 – Архітектура моделі VGG-16

### 3.7.2 Inception-v3

Нейронна мережа Inception - це також попередньо навчена модель, відома як GoogleNet, яка була представлена Google у 2014 році [43]. Inception була з 22 шарів, що мають 5М параметрів із розміром фільтра  $1 \times 1$ ,  $3 \times 3$  та

$5 \times 5$  для вилучення ознак у різних масштабах разом із максимальним пулінгом. Причиною використання фільтрів  $1 \times 1$  є економія часу на обчислення. Пізніше в 2015 році Google розширив модель Inception до InceptionV3 [44], в якій згорткові шари розкладаються на фактори для зменшення параметрів. Заміна  $5 \times 5$  згорткових фільтрів на два фільтри  $3 \times 3$  для зменшення обчислень, не впливаючи на продуктивність мережі. Модель InceptionV3 складається з 48 шарів. На рисунку 3.6 показано архітектуру InceptionV3.

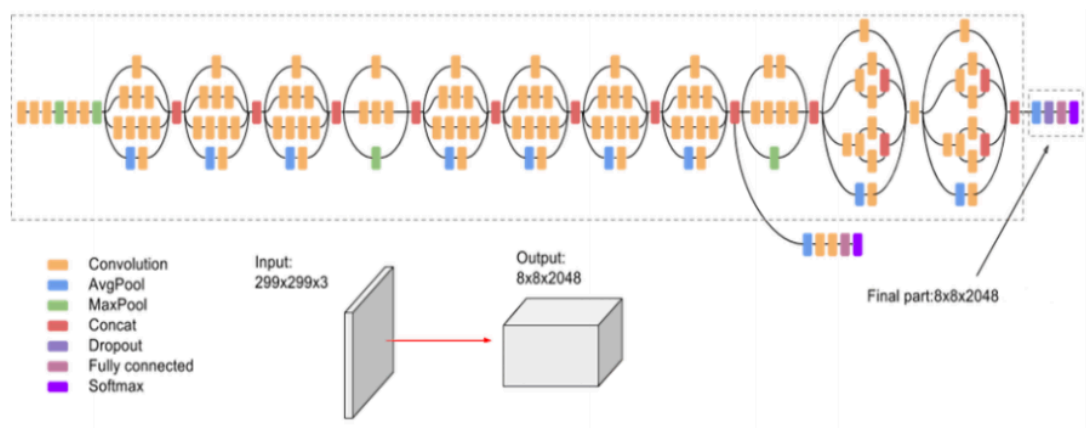


Рисунок 3.6 – Архітектура моделі Inception-v3

### 3.7.3 ResNet50

ResNet50 - це 50-шарова «залишкова мережа» з 26М параметрів. Залишкова мережа - це модель глибокої згорткової нейронної мережі, яка була впроваджена корпорацією Майкрософт у 2015 році [45]. У Залишковій мережі, на відміну від навчання ознак, мережа навчається з залишків, які є відніманням ознак, які навчені з входів шарів. ResNet використовував пропуск з'єднання для розповсюдження інформації по шарах. ResNet підключає вхід  $n$ -го рівня безпосередньо до деякого  $(n + x)$  рівня, що

дозволяє складати додаткові шари та робити глибоку нейронну мережу. На рисунку 3.7 показано архітектуру ResNet50.

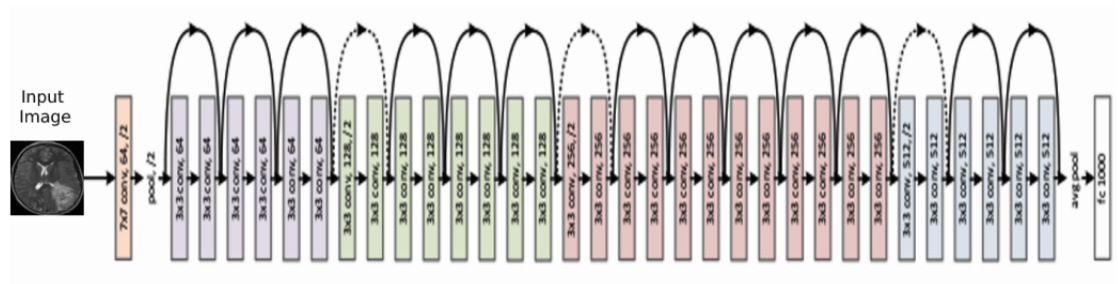


Рисунок 3.7 – Архітектура моделі ResNet50

### 3.8 Аналіз результатів

Було використано набір MPT зображень пухлини головного мозку, який є загальнодоступним і складається з 2530 реальних зображень головного мозку, розроблених рентгенологами на основі даних реальних постраждалих пацієнтів. Було розділено дані на навчальну, перевіірочну та тестувальну. Після розподілу було отримано 1770 зображень для навчання, 480 зображень для перевірки, 280 для тестування оцінки точності запропонованої моделі. На першому етапі було зроблена аугментація даних для збільшення набору даних шляхом незначних змін у MPT-зображеннях та вилучення цих доповнених зображень із запропонованої моделі CNN. Було використано python та бібліотеки на центральному процесорі i7 із тактовою частотою 2,5 ГГц та 16 Гб оперативної пам'яті. Запропонована модель показала 96% точності на навчальних даних та 89% точності на перевіірочних даних. Використовуючи підхід трансферного навчання, було навчено попередньо треновані моделі VGG16, ResNet-50 та Inception-v3 CNN на одному і тому ж наборі даних для порівняння точності запропонованої моделі CNN. VGG-16 продемонстрував 90% на тренувальних даних та 87%

точності на перевіірочній, ResNet-50 показав 92% на тренувальних даних та 87% на перевіірочних, а Inception-v3 продемонстрував 93% на тренувальних та 83% на перевіірочній. На рисунку 3.8 показано графік точності етапу тестування та перевірки під час ітерацій запропонованої моделі CNN, VGG-16, ResNet-50 та Inception-v3.

Було оцінено модель на тестуваних даних. У таблиці 3.1, істинно-позитивні (True Positive(TP)) та істинно-негативні (True Negative(TN)) показують правильну класифікацію, де TP показує зображення хворого мозку позитивними, а TN – показує зображення здорового мозку позитивними, в той час як хибно-позитивні(False Positive(FP)) та хибно-негативні(False Negative(FN)) показують неправильну класифікацію. FP показує зображення здорового мозку як позитивну (є пухлина) і на FN, яка показує зображення мозку з пухлиною як негативну (немає пухлини).

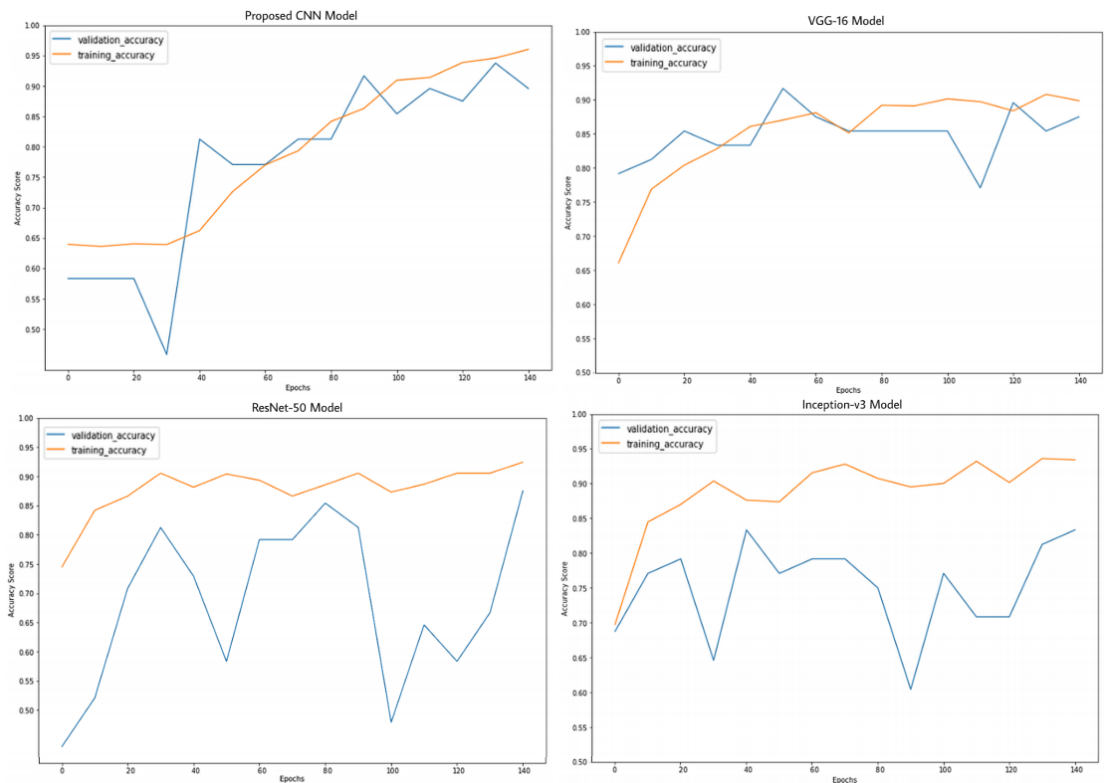


Рисунок 3.8 – Графік точності запропонованої моделі, VGG-16, ResNet-50 та Inception-v3

Були розраховані наступні метрики Accuracy, Precision, Recall, та F1-Score для запропонованої CNN та інших попередньо навчених моделей, як показано у таблиці 3.1.

Таблиця 3.1 – Класифікація моделі

Модель	True Positive	True Negative	False Positive	False Negative
Запропонована ЗНМ	140	120	20	0
VGG-16	140	130	10	0
ResNet-50	130	120	20	10
Inception-v3	110	100	30	40

$$Accuracy = TruePositive(TP) + TrueNegative(TN)/N$$

де N – загальна кількість прикладів.

Precision оцінює, скільки отриманих позитивних відповідей є правильними.

$$Precision = TruePositive (TP) / (TruePositive (TP) + FalsePositive (FP))$$

Recall оцінює, скільки отриманих позитивних відповідей було правильно передбачено з позитивних даних.

$$Recall = TruePositive (TP) / (TruePositive (TP) + FalseNegative (FN))$$

Тоді як оцінка F1 – це зважене середнє значення Precision та Recall.

$$F1 - Score = 2 \times (Precision \times Recall) / (Precision + Recall)$$

Точність, Precision, Recall, F1-Score та час навчання попередньо навчених моделей та запропонованої моделі CNN наведені в таблиці 2.

Також було побудовано площу під кривою ROC для всіх моделей і обчислено оцінку ROC-AUC, яка наведена у таблиці 3.2. Діаграма кривої ROC має False Positive Rate на осі X та True Positive Rate на осі Y. На рисунку 3.9 показано ROC криву моделей. True Positive Rate відомий як Recall, який показує позитивні дані, які правильно спрогнозовані як позитивні.

Таблиця 3.2 – Метрики

Модель	Точність	Precision	Recall	F1-Score	ROC-AUC Score	Час (сек)
Запропонована ЗНМ	0,93	0,88	1.0	0.94	1.0	3085
VGG-16	0.96	0.93	1.0	0.97	0.96	6846
ResNet-50	0.89	0.87	0.93	0.90	0.89	9091
Inception-v3	0.75	0.77	0.71	0.74	0.75	5630

$$\text{TruePositiveRate(TPR)} = \text{TruePositive(TP)} / (\text{TruePositive(TP)} + \text{FalseNegative(FN)})$$

Тоді як False Positive Rate показує негативні дані, які неправильно прогнозовані як позитивні.

$$\text{FalsePositiveRate(FPR)} = \text{FalsePositive(FP)} / (\text{FalsePositive(FP)} + \text{TrueNegative(TN)})$$

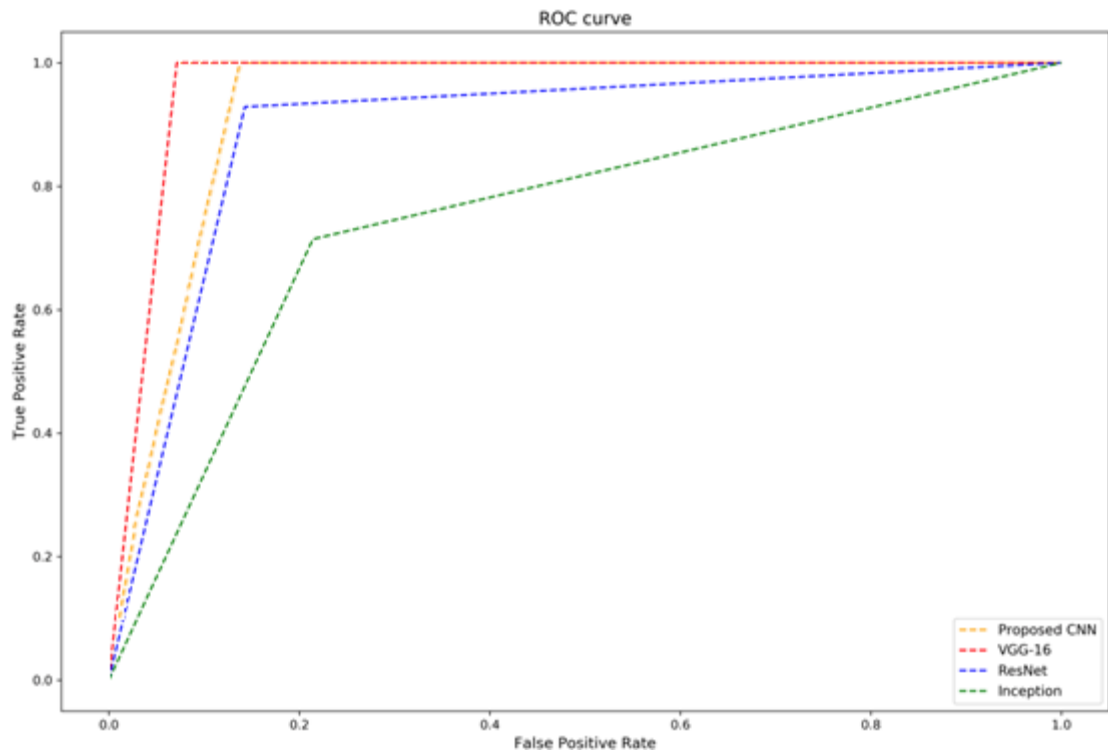


Рисунок 3.9 – Графік кривої ROC-AUC

### 3.9 Висновки до розділу 3

У даному розділі було описано процес підготовки зображень для використання у нейронній мережі та запропоновано нову архітектуру згорткової нейронної мережі. Також був використаний підхід трансферного навчання та порівняно отримані результати попередньо тренованих нейромереж та нової, які показали, що точність запропонованої моделі досить висока.



## РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

Стартап як форма малого ризикового (венчурного) підприємництва впродовж останнього десятиліття набула широкого розповсюдження у світі через зниження бар'єрів входу в ринок (із появою Інтернету як інструменту комунікацій та збуту стало простіше знаходити споживачів та інвесторів, займатись пошуком ресурсів, перетинати кордони між ринками різних країн), і вважається однією із наріжних складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

Проте створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка, що за різними оцінками складає від 10% до 20%. Ідея стартап-проекту, взята окремо, не вартує майже нічого. Головним завданням керівника проекту на початковому етапі його існування є перетворення ідеї проекту у працюючу бізнес-модель, що починається із формування концепції товару (послуги) для визначеної клієнтської групи за наявних ринкових умов. Розроблення та виведення стартап-проекту на ринок передбачає здійснення низки кроків, в межах яких визначають ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів. Далі наведено маркетинговий аналіз стартап проекту. В межах цього етапу:

- а) розробляється опис самої ідеї проекту та визначаються загальні напрями використання потенційного товару чи послуги, а також їх відмінність від конкурентів;
- б) аналізуються ринкові можливості щодо його реалізації;
- в) на базі аналізу ринкового середовища розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

#### 4.1 Опис ідеї проекту

В межах підпункту було проаналізовано і подано у вигляді таблиць:

- а) зміст ідеї (що пропонується);
- б) можливі напрямки застосування;
- в) основні вигоди, що може отримати користувач товару (за кожним напрямком застосування);
- г) чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подані у вигляді таблиці (таблиця 4.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 4.1 - Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Дана комплексна система дозволяє розв'язати проблему відслідковування раку головного мозку. На основі результатів класифікації відбувається розв'язання цієї проблеми – вчасне визначення хвороби на початковій стадії	1. Визначення стану здоров'я людини та наявності раку Головного мозку	Визначення захворювання на початковій стадії, вчасне лікування та нелінійності, що приводить до необхідності
	2. Аналіз стану здоров'я людини в динаміці	Можливість відстежування стану пацієнта

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

- а) визначення переліку техніко-економічних властивостей та характеристик ідеї;
- б) визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;
- в) проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (табл. 4.2).

Таблиця 4.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/ п	Техніко- економічні характерис тики ідеї	(потенційні) товари/концеп ціїконкурентів		W (слаб ка сторо на)	N (нейтр альна сторон а)	S (сил ьна сто р она)
		Мій проект	FORecast4u			
1.	Точніст ь прогнозу вання	Застосув аннякращої моделі	Відсутнє прогнозування			+
2.	Ризик невірн ого прогн озу	Існують, через велику кількість факторів	Відсутність прогнозу		+	

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

## 4.2 Технологічний аудит ідеї проекту

В межах даного підрозділу було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 4.3):

- 1) за якою технологією буде виготовлено товар згідно ідеї проекту?
- 2) чи існують такі технології, чи їх потрібно розробити/доробити?
- 3) чи доступні такі технології авторам проекту?

Таблиця 4.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявніс ть технолог ій	Доступність технологій
1	Створення системи класифікації кардіограм	Використання мови програмування Python	Наявна	Доступна
		Tensorflow	Наявна	Доступна
		Використання мови програмування R	Наявна	Доступна
		Використання мови програмування Java	Наявна	Наявна
Обрана технологія реалізації ідеї проекту: мова програмування Python. Бібліотека Tensorflow				

За результатами аналізу таблиці зроблено висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було обрано такі технології, як Python 3.4 та Tensorflow через їх доступність та безкоштовність.

### 4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час 77 ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 4.4).

Таблиця 4.4 - Попередня характеристика потенційного ринку стартап- проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	15000000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Ліцензії, регуляторні
5	Специфічні вимоги до стандартизації та сертифікації	+
6	Середня норма рентабельності в галузі (або по ринку), %	350

Середню норму рентабельності в галузі було порівняно із банківським відсотком на вкладення. Останній є меншим, тому є сенс вкладати гроші саме у цей проект.

За результатами аналізу таблиці 4.4 було зроблено висновок, що ринок є привабливим для входження.

Надалі були визначені потенційні групи клієнтів, їх характеристики та сформовано орієнтовний перелік вимог до товару для кожної групи (табл. 4.5).

Таблиця 4.5 - Характеристика потенційних клієнтів стартап-проекту

<b>№ п/п</b>	<b>Потреба, що формує ринок</b>	<b>Цільова аудиторія (цільові сегменти ринку)</b>	<b>Відмінності у поведінці різних потенційних цільових груп клієнтів</b>	<b>Вимоги споживачів до товару</b>
1.	Потреба в відокремленні сигналу від шуму побудови прогнозів.	Аналітики, аналітичні відділи, лікарі	Велика кількість даних.	Простота використання, висока точність
2.	Створення якісного прогнозу та визначення наявності захворювання	Користувачі додатку	Цікавить простота у використанні, низька ціна підтримки системи	Швидкість обробки, низька ціна
3.	Створення точного довгострокового прогнозу, відстеження стану здоров'я користувача в часі	Великі медичні компанії, користувачі додатку	Цікавить передусім точність довгострокових прогнозів	Висока якість прогнозу

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 4.6, 4.7).

Таблиця 4.6 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Вихід на ринок продуктів з кращими характеристиками	<p>Передбачити додаткові переваги власного програмного продукту (ПП) для того, щоб повідомити про них самепісля виходу на ринок конкурентів.</p> <p>Вдосконалення технічних моментіввласного продукту.</p> <p>Обрати нову цільову аудиторію і зосередитися на ній: зниження цін.</p>
2	Зміна потреб користувачів	Користувачам необхідне програмне забезпечення з іншим функціоналом	Передбачити можливість додавання нового функціоналу до створюваного ПП

Таблиця 4.7 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Конкуренція	Відсутність аналогічного продукту для вітчизняного користувача.	Адаптація програмного продукту до вітчизняних особливостей.
2	Поява нових методів прогнозування	З'являться нові методи, що будуть швидше та ефективніше прогнозувати показники	Покращити ПП додаванням нового функціоналу, розширення можливостей

## Продовження таблиці 4.7

3	Поява нових методів моделювання	З'являються нові методи, що будуть швидше, та більш точно моделювати процеси	Покращити ПП додаванням нового функціоналу, розширення можливостей
---	---------------------------------	--	--

Надалі було проведено аналіз пропозиції: визначили загальні риси конкуренції на ринку (таблиця 4.8).

Таблиця 4.8 - Ступеневий аналіз конкуренції на ринку

<b>Особливості конкурентного середовища</b>	<b>В чому проявляється дана характеристика</b>	<b>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</b>
1. Вказати тип конкуренції - монополія	На ринку присутні декілька компаній-конкурентів, але їх товар дещо відрізняється між собою.	Підтримка якості продукту та постійні нововведення, вдосконалення.
2. За рівнем конкурентної боротьби - міжнародний	Компанії-конкуренти з інших країн	Створити основу ПП так, щоб можна було легко переробити даний ПП для використання у галузях інших країн.
3. За галузевою ознакою - міжгалузева	Продукт може використовуватись для різних галузей	Постійне вдосконалення продукту, що не має прив'язки до сфери



Продовження таблиці 4.8

4. Конкуренція за видами товарів: - товарно-видова	Конкуренція міжвидами ПП, їх особливостями	Створити ПП, враховуючи недоліки конкурентів
5. За характером конкурентних переваг - нецінова	Вдосконалення технології створення ПП, щоб собівартість була нижчою	Удосконалення моделі. Використання більш дешевих технологій для розробки, ніж використовують конкуренти, але тільки якщо ці технології відповідають необхідним вимогам якості.
6. За інтенсивністю - не марочна	Бренд присутній, але його роль незначна	Реклама, участь у конференціях, семінарах.

Було проведено аналіз конкуренції у галузі за моделлю М. Портера (табл. 4.9).

Таблиця 4.9 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	<i>Навести перелік прямих конкурентів</i>	<i>Визначити бар'єри входження в ринок</i>	<i>Визначити фактори сили постачальників</i>	<i>Визначити фактори сили споживачів</i>	<i>Фактори загроз з боку замінників</i>

Продовження таблиці 4.9

	SAS Matlab	Наявність вже існуючих рішень	-	Контроль якості продукту	Наявність більш широкого функціоналу, зручнішого інтерфейсу та авторитет (перевірена якість)
<b>Висновки:</b>	Досить інтенсивна конкурентна боротьба з вже закріпив шимися на ринку гравцями	Є можливість і виходу на ринок, але є і конкуренти. Строки – 18 місяців.	-	Клієнти диктують умови роботи на ринку: зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для побудови моделей і прогнозів.	Необхідно випускати ПЗ не гірше, ніж у конкурентів та розширяти функціонал.

За результатами аналізу табл. 4.9 було зроблено висновок про можливість роботи на ринку з огляду на конкурентну ситуацію. Також було зроблено висновок щодо характеристик, які повинен мати проект, щоб бути конкурентоспроможним на ринку.

Цей висновок був врахований при формулюванні переліку факторів конкурентоспроможності у наступному пункті. На основі аналізу конкуренції, проведеного в табл. 4.9, а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (таблиці 4.6, 4.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлено у табл. 4.10

Таблиця 4.10 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Багатофункціональність	Жоден конкурент не є настільки багатункціональним, не здатен на прогноз, чистку шумів та класифікацію сигналів
2	Якість	Можливість використання системи без складних імплементація в існуючу екосистему автомобіля
3	Висока якість прогнозу, велика кількість допоміжних статистичних даних	Робота з клієнтами – великими компаніями та окремими спеціалістами

За визначеними факторами конкурентоспроможності (табл. 4.10) проведено аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.11 - Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з ... (назва підприємства)						
			-3	-2	-1	0	+1	+2	+3
1	Якість аналізу	15					*		
2	Простота використання	20			*				
3	Орієнтованість на кінцевого споживача	7					*		

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 4.11). Перелік ринкових загроз та

ринкових можливостей було складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

Таблиця 4.12 - SWOT-аналіз стартап-проекту

<b>Сильні сторони:</b> Точність аналізу Простота використання Автономність	<b>Слабкі сторони:</b> Потрібен час для навчання системи Інтерфейс користувача
<b>Можливості:</b> Аналіз емоцій людини на основі біосигналів Застосування системи для попередження серцевих нападів	<b>Загрози:</b> Конкуренція

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. таблицю 4.9, аналіз потенційних конкурентів). Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (таблиця 4.13).

Таблиця 4.13 - Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Ліцензування алгоритму і патентів	85%	6 місяців
2	Створення ПП з подальшим з подальшою інтеграцією в систему автомобіля	75%	18 місяців

Після аналізу було обрано альтернативу №1.

#### 4.4 Аналіз ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: було проведено опис цільових груп потенційних споживачів (таблиця 4.14).

Таблиця 4.14 - Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Компанії (українські та міжнародні) діяльність яких пов'язана з фінансово-економічною сферами.	Висока	Високий	Сильна	Складно

Продовження таблиці 4.14

2	Приватні підприємства міського та міжнародного рівня, діяльність яких пов'язана з фінансово-економічною сферами.	Висока	Високий	Сильна	Складно
3	Приватні підприємства, обласного рівня.	Помірна	Помірний	Помірна	Середня складність
4	Підприємства регіонального характеру	Помірна	Слабкий	Слабка	Просто
5	ФОП, які діють у фінансово-економічній сферах.	Слабка	Слабкий	Слабка	Просто

Які цільові групи обрано: 1,2,3.

За результатами аналізу потенційних груп споживачів було обрано цільові групи, для яких буде запропоновано даний товар, та визначено стратегію охоплення ринку - стратегію диференційованого маркетингу (компанія працює з декількома сегментами).

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (таблиця 4.15).

Таблиця 4.15 - Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1		Визначити потреби кожної з груп, розробити відповідно до них стратегії привабливості клієнтів та маркетингової комунікації	Цінова політика, універсальність продукту, орієнтованість на кінцевого споживача	Стратегія диференціації

Наступним кроком обрано стратегію конкурентної поведінки (таблиця 4.16).

Таблиця 4.16 - Визначення базової стратегії конкурентної поведінки

<b>№ п/п</b>	<b>Чи є проект «першопрохідцем» на ринку?</b>	<b>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?</b>	<b>Чи буде компанія копіювати основні характеристик товару конкурента, і які?</b>	<b>Стратегія конкурентної поведінки*</b>
1	«Прешопроходець»	Шукати нових	Ні	Стратегія заняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. таблицю 4.5), а також в залежності від обраної базової стратегії розвитку (таблиця 4.15) та стратегії конкурентної поведінки (таблиця 4.16) розроблено стратегію позиціонування (таблиця 4.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап-компанії на ринку.

Таблиця 4.17 - Визначення стратегії позиціонування

<b>№ п / п</b>	<b>Вимоги до товару цільової аудиторії</b>	<b>Стратегія розвитку</b>	<b>Ключові конкурентоспроможні позиції власного стартап-проекту</b>	<b>Вибір асоціацій, які мають сформувати комплексну позицію власного проекту</b>
1	Легкість розуміння, зручний інтерфейс, надійний, швидкий, точний та достовірний ПП для побудови моделей і прогнозів.	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Економія часу; Зручність застосування; Практичність та точність результату

#### 4.5 Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього у таблиці 4.18 підсумовано результати попереднього аналізу конкурентоспроможності товару. Концепція товару - письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 4.18 - Визначення ключових переваг концепції потенційного товару

<b>№ п/п</b>	<b>Потреба</b>	<b>Вигода, яку пропонує товар</b>	<b>Ключові переваги перед конкурентами(існуючі або такі, що потрібно створити)</b>
1	Швидкість отримання результату	Швидка побудова моделі та створення прогнозу	Відсутність необхідності звертатися до сторонньої особи/компанії для побудови моделі та прогнозу. Дані компанії- користувача, якими оперує ПП, не передаються третім особам, чого вимагає політика безпеки багатьох компаній.



Продовження таблиці 4.18

2	Зручність застосування	Не потрібно мати глибоких знань, для того щоб побудувати модель та спрогнозувати показники	ПП сам обирає необхідний та оптимальний метод для побудови моделі та прогнозу. Непотрібно мати глибоких знань у прогнозуванні для того, щоб користуватися ПП
3	Практичність та точність результату	Користувач отримує точні (з малою похибкою розбіжності) результати.	Користувач на виході роботи ПП отримує модель та прогноз, котрі відповідають необхідним показникам достовірності та точності.

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 4.19).

1-й рівень При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби і / або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб.

2-й рівень Цей рівень являє рішення того, як буде реалізований товар в реальному/ включає в себе якість, властивості, дизайн, упаковку, ціну.

3-й рівень Товар з підкріпленням (супроводом) - додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості, доставка, умови оплати та ін).

Таблиця 4.19 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
<i>I. Товар за задумом</i>	Зручність та швидкість отримання практичного результату щодо побудови моделі та прогнозування процесів.

Продовження таблиці 4.19

II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. функція побудови моделі процесу 2. функція побудови прогнозу		
	Якість: достовірність побудови математичної моделі, достовірність побудови прогнозу		
	Пакування: відсутнє		
	Марка: StatLabs «Forec»		
III. Товар із підкріпленням	До продажу: відсутнє		
	Після продажу: персональна підтримка в обслуговуванні за додаткову плату.		
Вихідний код та математична модель будуть закриті. На ідею зареєстровано патент.			

Після формування маркетингової моделі товару слід відмітити, що проект буде захищено від копіювання за допомогою ноу-хау. Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субституту, а також аналіз рівня доходів цільової групи споживачів (таблиця 4.20). Аналіз проведено експертним методом.

Таблиця 4.20 - Визначення меж встановлення ціни

<b>№ п/п</b>	<b>Рівень цін на товари-замінники</b>	<b>Рівень цін на товари-аналоги</b>	<b>Рівень доходів цільової групи споживачів</b>	<b>Верхня та нижня межівстановлення ціни на товар/послугу</b>
1	1800\$	3500\$	У всіх трьох груп високий рівень доходів	Базова покупка 1000\$ Подальша персональна підтримка в обслуговуванні 150\$/місяць

Наступним кроком є визначення оптимальної системи збуту, в межах якого було прийняте рішення (таблиця 4.21)

Таблиця 4.21 - Формування системи збуту

<b>№ п/п</b>	<b>Специфіка закупівельної поведінки цільових клієнтів</b>	<b>Функції збуту, які має виконувати постачальник товару</b>	<b>Глибина каналу збуту</b>	<b>Оптимальна система збуту</b>
1	Цільові клієнти – компанії, які бажають впровадити у своїй роботі сучасні засоби, які допоможуть отримати вигоду та покращити дохідність. Вони цікавляться сучасними розробками та інноваційними рішеннями, тому відвідують конференції, інтернет-конференції, семінари.	Встановлення контактів із споживачами і підтримання їх. Формування попиту і стимулювання збуту. Дослідницька робота зі збору маркетингової інформації. Доробка товару, виходячи з потреб конкретного покупця.	Один (від виробника одразу споживачу)	Прямий канал збуту до споживача, мінімізувати збутові витрати розвитку маркетингового спілкування із споживачем

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 4.22).

Таблиця 4.22 - Концепція маркетингових комунікацій

№ п / п	Специфіка поведінки цілевих клієнтів	Канали комунікації, якими користують ся цілові клієнти	Ключові позиції, обрані для позиціо нування	Завдання рекламного повідомлення	Концепція реклам ного зверне ння
	Цілі клієнти – компанії, які бажають впровадити у своїй роботі сучасні засоби, які допоможуть	Конференції, інтернет-конференції, семінари, огляд професійної	Позиція на основі порівняння фірми з товарами	Створення репутації фірми — виробнику чи посереднику; збільшення чистого	Шукає вірний шлях для розвитку вашої компанії? Досить
	отримати вигоду та покращити дохідність. Вони цікавляться сучасними розробками та інноваційними рішеннями, тому відвідують конференції, інтернет-конференції, семінари.	літератури, інтернет, періодичні видання у різноманітних (профільних) галузях.	конкурентів; Відмінні особливості споживача	прибутку та рентабельності фірми; – збільшення потоків покупців та обсягів продажу; – стабілізація обсягів продажу в період зменшення попиту та загального спаду ділової активності.	даремно гаяти час на вгадування вірної стратегії! Користуйтеся «Forec» і світле майбутнє вашої компанії забезпечено!

Результатом підрозділу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

#### 4.6 Висновки до розділу 4

В даному розділі було проведено аналіз програмного продукту у якості стартап-проекту. Можна зазначити, що у проекті є можливість комерціалізації, оскільки ринок потребує якісний продукт, що надає оцінки стану серцево- судинної системи.

На ринку наявна монополістична конкуренція, існує декілька фірм-конкурентів, але їх товар дещо відрізняється, тому вихід на ринок не буде легким і потребує грамотної стратегії виходу. Для впровадження ринкової реалізації проекту слід обрати альтернативу, яка передбачає розробку програмного продукту з подальшим розповсюдженням ліцензій та права на використання за певну роялті.

## ВИСНОВКИ

У цій магістерській дисертації був представлений новий підхід до класифікації пухлин головного мозку. Спочатку, використовуючи техніку виявлення краю зображення, знаходиться область, що більше цікавить на зображенні МРТ та обрізається. Наступним кроком є використання аугментації даних для збільшення розміру навчальних даних. Було запропоновано нову архітектуру CNN для класифікації пухлин головного мозку. Нарешті, було написано програму, яка знаходить рак головного мозку на зображенні та класифікує її. Для дуже точних результатів нейронна мережа вимагає дуже великого обсягу даних для тренування, але отриманий результат показує, що навіть на обраному наборі даних можна досягти непоганої точності, і отриманий рівень точності достатньо високий у порівнянні з VGG-16, ResNet-50 та Inception-v3. Середній час навчання запропонованої моделі за епоху становить 205 секунд, тоді як у VGG-16 займає 456 секунд, ResNet-50 - 606 секунд, а Inception-v3 - 375 секунд у середньому за епоху. Отже, запропонована модель потребує менше обчислювальних витрат, оскільки вимагає менше часу на навчання.

Запропонована система може відігравати ключове значення при виявленні пухлин у пацієнтів з пухлиною головного мозку. Для подальшого підвищення ефективності моделі можна було виділено наступні напрямки подальших досліджень:

1. Зробити налаштування гіперпараметрів та кращу техніку попередньої обробки даних.
2. Розширення методу для категоріальних проблем класифікації, таких як ідентифікація типів пухлин головного мозку, таких як гліома, менінгіома та гіпофіз, або може бути використаний для виявлення інших аномалій мозку.

3. Використання запропонованої системи у ранній діагностиці небезпечних захворювань в інших клінічних областях, пов'язаних з медичною візуалізацією, зокрема раку легенів та раку молочної залози, рівень смертності яких у світі дуже високий.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Thompson P.M., Moussai J., Zohoori S. et. al. Cortical variability and asymmetry in normal aging and Alzheimer. *Cereb Cortex*. 1998. Vol. 8, No. 6. P. 492-509.
2. P. Paschka, R.F. Schlenk, V.I. Gaidzik. et al. IDH1 and IDH2 mutations are frequent genetic alterations in acute myeloid leukemia and confer adverse prognosis in cytogenetically normal acute myeloid leukemia with NPM1 mutation without FLT3 internal tandem duplication. *J. Clin. Oncology*. 2010. P. 3636–3643
3. Min JK, Dunning A, Lin FY, et al. Age- and sexrelated differences in all-cause mortality risk based on coronary computed tomography angiography findings results from the International Multicenter CONFIRM (Coronary CT Angiography Evaluation for Clinical Outcomes: An International Multicenter Registry) of 23,854 patients without known coronary artery disease. *Cardiol*, 2011. P. 849–60.
4. Vaidyanathan M., Velthuisen R., Clarke L.P. et. al. Quantitation of brain tumor in MRI for treatment planning. *Proc. the 16th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society*. 1994. P. 555 -556.
5. Vezina G. MR Imaging of Brain Tumors - Recent Developments. M.D *Children's National Medical Journal*. Washington D.C. 2015. P. 10-21.
6. Werring, David J., et al. Cognitive dysfunction in patients with cerebral microbleeds on T2\*-weighted gradient-echo MRI. *Stroke*. 2004. P. 2265-2275.
7. Hajnal, Joseph V. et al. Use of fluid attenuated inversion recovery (FLAIR) pulse sequences in MRI of the brain. *Journal of computer assisted tomography*. Vol.16, No. 6. 1992. P. 841-844.



8. Kato, Hiroyuki, et al. Silent cerebral microbleeds on T2\*-weighted MRI correlation with stroke subtype, stroke recurrence, and leukoaraiosis. *Stroke*. 2002. Vol. 33, No. 6. P. 1536-1540.
9. Brain cancer. URL: <http://www.cancer.org/acs/groups/content/@editorial/documents/document/acspc-044552.pdf>
10. Fisher JL, Schwartzbaum JA, Wrensch et al. Evaluation of epidemiologic evidence for primary adult brain tumor risk factors using evidence-based medicine. *Prog Neurol Surg*. 2006. P. 54-79.
11. Wrensch M, Minn Y, Chew T. et al. Epidemiology of primary brain tumors: current concepts and review of the literature. *Neuro Oncol*. 2004. P. 278-299.
12. Bondy M, Wiencke J, Wrensch M. et al. *Genetics of primary brain tumors: a review*. *J Neurooncol*. 2004. Vol. 18. P. 69-81.
13. Kijima C, Miyashita T, Suzuki M. Two cases of nevoid basal cell carcinoma syndrome associated with meningioma caused by a PTCH1 or SUFU germline mutation. *Fam Cancer*. 2012. P. 565-570.
14. Chang SM, Parney IF, Huang W. et al. Patterns of care for adults with newly diagnosed malignant glioma. *JAMA*. 2015. Vol. 293. P. 557-564.
15. Forsyth PA, Posner JB. Headaches in patients with brain tumors: a study of 111 patients. *Neurology*. 2003. Vol. 43, No. 9. P. 1678-1683.
16. DeAngelis LM. Brain tumors. *N Engl J Medical*. 2001. Vol. 344. P. 114-123.
17. Liigant A, Haldre S, Oun A. et al. Seizure disorders in patients with brain tumors. *Eur Neuro*. 2001. Vol. 45. P. 46-51.
18. Iofe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift; arXiv preprint arXiv:1502.03167. 2015.
19. Ruder S. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747. 2015.
20. Bottou L. Large-scale machine learning with stochastic gradient descent. In: *Proceedings of COMPSTAT'2010*. Springer. 2010. P. 177–86.

21. Hinton G, Srivastava N, Swersky K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Neural Nets*. 2012. P. 8-14.
22. Zhang Z. Improved Adam optimizer for deep neural networks. *IEEE/ACM 26th international symposium on quality of service (IWQoS)*. IEEE. 2018. P. 718–732.
23. S. Basheera, M. S. S. Ram, Classification of brain tumors using deep features extracted using CNN, *J. Phys.*. 2019. P. 23-34.
24. M. Sajjad, S. Khan, M. Khan, W. Wu. et al. Multi-grade brain tumor classification using deep CNN with extensive data augmentation, *J. Comput. Sci.* 2019, P. 174–182.
25. R. Carlo, C. Renato, C. Giuseppe. et al. Distinguishing Functional from Non-functional Pituitary Macroadenomas with a Machine Learning Analysis. *Mediterranean Conference on Medical and Biological Engineering and Computing, Springer*. 2019. Vol. 9. P. 1822–1829.
26. S. Khawaldeh, U. Pervaiz, A. Rafiq et al. Noninvasive grading of glioma tumor using magnetic resonance imaging with convolutional neural networks, *J. Appl. Sci.* 2018. Vol. 8. P. 27-29.
27. N. Abiwinanda, M. Hanif, S. Hesaputra et al. Brain tumor classification using convolutional neural network. *World Congress on Medical Physics and Biomedical Engineering, Springer, Singapore*, 2019. P. 674-671.
28. S. Das, R. Aranya, N. Labiba, Brain tumor classification using convolutional neural network. *1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, 2019. P. 76-85.
29. V. Romeo, R. Cuocolo, C. Ricciardi et al., Prediction of tumor grade and nodal status in oropharyngeal and oral cavity squamous-cell carcinoma using a radiomic approach, *Anticancer Res.* 2020, Vol. 40. No 1. P. 271–280.
30. M. Talo, U. B. Baloglu, O. Yldrm et al. Application of deep transfer learning for automated brain abnormality classification using MRI images, *Cognitive Systems Research*. 2019. Vol. 54, No. 5. P. 176–188.

- 31.A. Rehman, S. Naz, M. I. Razzak et al. A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning, *Circuits Syst. Signal Process.* 2020, P. 757–775.
- 32.A. Cinar, M. Yldrm, Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture, *Med. Hypotheses.* 2020. Vol.12. P. 31-45.
- 33.N. Chakrabarty, Brain MRI images dataset for brain tumor detection, Kaggle. 2019. URL: <https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection>.
- 34.J. F. Canny, Canny Edge Detection, Open Source Computer Vision, OpenCV. URL: [https://docs.opencv.org/trunk/da/d22/tutorial\\_py\\_canny.html](https://docs.opencv.org/trunk/da/d22/tutorial_py_canny.html).
- 35.C. Shorten, T. M. Khoshgoftaa, A survey on image data augmentation for deep learning, *J. Big Data.* 2019.Vol. 35. P. 58-60.
- 36.Keras, Image data preprocessing keras API, Keras documentation. URL: <https://keras.io/api/preprocessing/image/>.
- 37.V. Nair, G. Hinton, Rectified linear units improve restricted boltzmann machines, *ICML.* 2010. P. 176-191.
- 38.S. Mannor, D. Peleg, R. Rubinstein, The cross entropy method for classification. *Proceedings of the 22nd international conference on Machine learning.* 2005. P. 167-182.
- 39.D. P. Kingma, J. Ba, A method for stochastic optimization, arXiv: 1412.6980.
- 40.H. Robbins, S. Munro, A Stochastic Approximation Method, *Ann. Math. Stat.*, 22. 1951.P. 400– 407.
- 41.G. Hinton, Neural networks for machine learning online course lecture 6a, Coursera. URL: [http://www.cs.toronto.edu/tijmen/csc321/slides/lecture\\_slides lec6.pdf](http://www.cs.toronto.edu/tijmen/csc321/slides/lecture_slides lec6.pdf).
- 42.K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv: 1409.1556.

- 43.C. Szegedy, W. Liu, Y. Jia, P. Sermanet, et al. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015. P. 1872-1901.
- 44.C. Szegedy, V. Vanhoucke, S. Ioffe, et al. Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. P. 631-639.
- 45.K. He, X. Zhang, S. Ren, J. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. P. 1283-1292.

## ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ

```

from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation, Flatten
from keras.layers.convolutional import Convolution2D, MaxPooling2D
from keras.callbacks import ModelCheckpoint
from keras.optimizers import SGD,RMSprop,Adam
from keras.utils import np_utils
from keras.initializers import constant
from skimage import io
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
import os
#import theano
import tensorflow
from PIL import Image
from numpy import *
from sklearn.utils import shuffle
from sklearn.cross_validation import train_test_split
from keras.layers.advanced_activations import LeakyReLU
from glob import glob

from keras import backend as K
path1 = 'T1Patches'
listing = os.listdir(path1)
num_samples=size(listing)
print (num_samples)
imlist = os.listdir(path1)

#im1 = array(Image.open(path1 + '/' + imlist[0]))
#im2 = array(Image.open(path1 + '/' + imlist[1]))
#from sklearn import preprocessing
#immatrix = array([preprocessing.scale(array(io.imread(path1 + '/' + im1)))
#    for im1 in sorted(imlist)],'f')
immatrix = array([array(io.imread(path1 + '/' + im1).reshape(1,31,31))
    for im1 in sorted(imlist)],'f')
print ("Shape is "+ str(immatrix.shape))
label=np.ones((num_samples,), dtype = np.int)

```

```

label[0:500] = 0

label[500:1000] = 1

label[1000:1500] = 2

label[1500:2000] = 3

label[2000:] = 4
#data,Label = shuffle(immatrix,label, random_state=2)

train_data = [immatrix,label]
batch_size = 25

num_classes = 5

nb_epoch = 100
#(X_train, y_train) = (train_data[0],train_data[1])
#(X_test, y_test) = (test_data[0],test_data[1])
#print ('X.shape = ',X.shape)
#print ('y.shape = ',y.shape)
X_train, X_test, y_train, y_test = train_test_split(train_data[0], train_data[1], test_size=0.2)
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')

#X_train /= 65536
#X_test /= 65536

X_train /= 255
X_test /= 255
print(X_train.shape[0], 'train samples')
print(X_test.shape[0], 'test samples')

# convert class vectors to binary class matrices
Y_train = np_utils.to_categorical(y_train, num_classes)
Y_test = np_utils.to_categorical(y_test, num_classes)

#%%%
K.set_image_dim_ordering('th')
#first set of CONV => CONV => CONV => LReLU => MAXPOOL
model = Sequential()

```

```

#First convolution Layer
model.add(Convolution2D(256, (3, 3), padding='same', input_shape=(1,31,31))) #Try for 25x25 patch
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2),strides=1))
model.add(Dropout(0.50))

model.add(Convolution2D(256, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2),strides=1))
model.add(Dropout(0.50))

model.add(Convolution2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2),strides=1))
model.add(Dropout(0.40))

model.add(Convolution2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2),strides=1))
model.add(Dropout(0.40))

model.add(Flatten()) # No dropout after flattening.
model.add(Dense(500))
model.add(Activation('relu'))

model.add(Dense(num_classes))
model.add(Activation('softmax'))
"""
model.add(Convolution2D(64, kernel_size=(3, 3), padding="same",strides=(1, 1), data_format='channels_first',
input_shape = (1, 31, 31), kernel_initializer = 'glorot_normal', bias_initializer=constant(0.1) ))
model.add(LeakyReLU(alpha=0.333))
model.add(Convolution2D(64, kernel_size=(3, 3), padding="same",strides=(1, 1), data_format='channels_first',
input_shape = (64, 33, 33), kernel_initializer = 'glorot_normal', bias_initializer=constant(0.1) ))
model.add(LeakyReLU(alpha=0.333))
model.add(Convolution2D(64, kernel_size=(3, 3), padding="same",strides=(1, 1), data_format='channels_first',
input_shape = (64, 33, 33), kernel_initializer = 'glorot_normal', bias_initializer=constant(0.1) ))
model.add(LeakyReLU(alpha=0.333))
model.add(MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

#second set of CONV => CONV => CONV => LReLU => MAXPOOL
model.add(Convolution2D(1

```

```

28, kernel_size=(3, 3), padding="same",strides=(1, 1), data_format='channels_first', input_shape = (64, 16, 16),
kernel_initializer = 'glorot_normal', bias_initializer=constant(0.1) ))
model.add(LeakyReLU(alpha=0.333))
model.add(Convolution2D(128, kernel_size=(3, 3), padding="same",strides=(1, 1), data_format='channels_first',
input_shape = (128, 16, 16), kernel_initializer = 'glorot_normal', bias_initializer=constant(0.1) ))
model.add(LeakyReLU(alpha=0.333))
model.add(Convolution2D(128, kernel_size=(3, 3), padding="same",strides=(1, 1), data_format='channels_first',
input_shape = (128, 16, 16), kernel_initializer = 'glorot_normal', bias_initializer=constant(0.1) ))
model.add(LeakyReLU(alpha = 0.333))
model.add(MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

#Fully connected layers

# FC => LReLU => FC => LReLU
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(256, kernel_initializer = 'glorot_normal', bias_initializer=constant(0.1)))
model.add(LeakyReLU(0.333))
model.add(Dropout(0.5))
model.add(Dense(256, kernel_initializer = 'glorot_normal', bias_initializer=constant(0.1)))
model.add(LeakyReLU(0.333))
model.add(Dropout(0.5))

# FC => SOFTMAX
model.add(Dense(5, kernel_initializer = 'glorot_normal', bias_initializer = constant(0.1)))
model.add(Activation("softmax"))
"""
opt = SGD(lr=0.1)
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
adam = Adam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-08, decay=0.0)
model.compile(loss='categorical_crossentropy', optimizer= sgd, metrics = ['accuracy'])

print ('Done.')
checkpoint = ModelCheckpoint('weights-Test-CNN.hdf5', monitor='val_loss', verbose=0, save_best_only=False,
save_weights_only=True, mode='auto', period=1)

import pandas

hist = model.fit(X_train, Y_train, batch_size=batch_size, nb_epoch=nb_epoch,
                verbose=1,shuffle=True, validation_data=(X_test, Y_test),callbacks=[checkpoint])
pandas.DataFrame(hist.history).to_csv("T1Patches.csv")

```



```

#hist = model.fit(X_train, Y_train, batch_size=batch_size, nb_epoch=nb_epoch,
#               verbose=1,shuffle=True, validation_data=(X_test, Y_test))
#pandas.DataFrame(hist.history).to_csv("14thDechistory1.csv")

train_loss=hist.history['loss']
val_loss=hist.history['val_loss']
train_acc=hist.history['acc']
val_acc=hist.history['val_acc']
xc=range(nb_epoch)
from sklearn.metrics import classification_report,confusion_matrix

Y_pred = model.predict(X_test)
print(Y_pred)
y_pred = np.argmax(Y_pred, axis=1)
print(y_pred)

y_pred = model.predict_classes(X_test)
print(y_pred)

p=model.predict_proba(X_test) # to predict probability

target_names = ['class 0(normal)', 'class 1(necros)', 'class 2(adema)', 'class 3(enhancing)', 'class 4(non-enhancing)']
print(classification_report(np.argmax(Y_test,axis=1), y_pred,target_names=target_names))
print(confusion_matrix(np.argmax(Y_test,axis=1), y_pred))

# saving weights

fname = "weights-Test-CNN.hdf5"
model.save_weights(fname,overwrite=True)
tf.summary.scalar("accuracy", accuracy)

merged = tf.summary.merge_all()

self.x_input = x_input
self.y_input = y_input
self.keep_prob = keep_prob
self.output = output
self.loss = loss
self.optimizer = optimizer
self.accuracy = accuracy
self.merged = merged

```

```

sys.stdout.write("\n<log> Building graph...")
__graph__()
sys.stdout.write("</log>\n")

def train(self, checkpoint_path, epochs, log_path, train_data, test_data):
    """Trains the initialized model.
    :param checkpoint_path: The path where to save the trained model.
    :param epochs: The number of passes through the entire dataset.
    :param log_path: The path where to save the TensorBoard logs.
    :param train_data: The training dataset.
    :param test_data: The testing dataset.
    :return: None
    """
    if not os.path.exists(path=log_path):
        os.mkdir(log_path)

    if not os.path.exists(path=checkpoint_path):
        os.mkdir(checkpoint_path)

    saver = tf.train.Saver(max_to_keep=4)

    init = tf.global_variables_initializer()

    timestamp = str(time.asctime())

    train_writer = tf.summary.FileWriter(
        logdir=log_path + timestamp + "-training", graph=tf.get_default_graph()
    )

    with tf.Session() as sess:
        sess.run(init)

        checkpoint = tf.train.get_checkpoint_state(checkpoint_path)

        if checkpoint and checkpoint.model_checkpoint_path:
            saver = tf.train.import_meta_graph(
                checkpoint.model_checkpoint_path + ".meta"
            )
            saver.restore(sess, tf.train.latest_checkpoint(checkpoint_path))

        for index in range(epochs):
            # train by batch

```

```

batch_features, batch_labels = train_data.next_batch(self.batch_size)
batch_labels[batch_labels == 0] = -1

# input dictionary with dropout of 50%
feed_dict = {
    self.x_input: batch_features,
    self.y_input: batch_labels,
    self.keep_prob: 0.5,
}

# run the train op
summary, _, loss = sess.run(
    [self.merged, self.optimizer, self.loss], feed_dict=feed_dict
)

# every 100th step and at 0,
if index % 100 == 0:
    feed_dict = {
        self.x_input: batch_features,
        self.y_input: batch_labels,
        self.keep_prob: 1.0,
    }

    # get the accuracy of training
    train_accuracy = sess.run(self.accuracy, feed_dict=feed_dict)

    # display the training accuracy
    print(
        "step: {}, training accuracy : {}, training loss : {}".format(
            index, train_accuracy, loss
        )
    )

    train_writer.add_summary(summary=summary, global_step=index)

    saver.save(
        sess,
        save_path=os.path.join(checkpoint_path, self.name),
        global_step=index,
    )

test_features = test_data.images

```

```
test_labels = test_data.labels
test_labels[test_labels == 0] = -1
```

```
feed_dict = {
    self.x_input: test_features,
    self.y_input: test_labels,
    self.keep_prob: 1.0,
}
```

```
test_accuracy = sess.run(self.accuracy, feed_dict=feed_dict)
```

```
print("Test Accuracy: {}".format(test_accuracy))
```

```
@staticmethod
```

```
def weight_variable(shape):
```

```
    """Returns a weight matrix consisting of arbitrary values.
    :param shape: The shape of the weight matrix to create.
    :return: The weight matrix con
```

```
sisting of arbitrary values.
```

```
    """
    initial = tf.truncated_normal(shape, stddev=0.1)
    return tf.Variable(initial)
```

```
@staticmethod
```

```
def bias_variable(shape):
```

```
    """Returns a bias matrix consisting of 0.1 values.
    :param shape: The shape of the bias matrix to create.
    :return: The bias matrix consisting of 0.1 values.
    """
```

```
    initial = tf.constant(0.1, shape=shape)
    return tf.Variable(initial)
```

```
@staticmethod
```

```
def conv2d(features, weight):
```

```
    """Produces a convolutional layer that filters an image subregion
    :param features: The layer input.
    :param weight: The size of the layer filter.
    :return: Returns a convolutional layer.
    """
```

```
    return tf.nn.conv2d(features, weight, strides=[1, 1, 1, 1], padding="SAME")
```

```

@staticmethod
def max_pool_2x2(features):
    """Downnsamples the image based on convolutional layer
    :param features: The input to downsample.
    :return: Downsampled input.
    """
    return tf.nn.max_pool(
        features, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding="SAME"
    )

from future import absolute_import
from future import division
from future import print_function

version = "0.1.0"
author = "Abien Fred Agarap"

import argparse
from model.cnn_softmax import CNN
from model.cnn_svm import CNNSVM
from tensorflow.examples.tutorials.mnist import input_data

def parse_args():
    parser = argparse.ArgumentParser(
        description="CNN & CNN-SVM for Image Classification"
    )
    group = parser.add_argument_group("Arguments")
    group.add_argument(
        "-m", "--model", required=True, type=str, help="[1] CNN-Softmax, [2] CNN-SVM"
    )
    group.add_argument(
        "-d", "--dataset", required=True, type=str, help="path of the MNIST dataset"
    )
    group.add_argument(
        "-p",
        "--penalty_parameter",
        required=False,
        type=int,
        help="the SVM C penalty parameter",
    )
    group.add_argument(
        "-c",

```

```

        "--checkpoint_path",
        required=True,
        type=str,
        help="path where to save the trained model",
    )
    group.add_argument(
        "-l",
        "--log_path",
        required=True,
        type=str,
        help="path where to save the TensorBoard logs",
    )
    arguments = parser.parse_args()
    return arguments

if name == "__main__":
    args = parse_args()

    mnist = input_data.read_data_sets(args.dataset, one_hot=True)
    num_classes = mnist.train.labels.shape[1]
    sequence_length = mnist.train.images.shape[1]
    model_choice = args.model

    assert (
        model_choice == "1" or model_choice == "2"
    ), "Invalid choice: Choose between 1 and 2 only."

    if model_choice == "1":
        model = CNN(
            alpha=1e-3,
            batch_size=128,
            num_classes=num_classes,
            num_features=sequence_length,
        )
        model.train(
            checkpoint_path=args.checkpoint_path,
            epochs=10000,
            log_path=args.log_path,
            train_data=mnist.train,
            test_data=mnist.test,
        )

```

```
elif model_choice == "2":  
    model = CNNSVM(  
        alpha=1e-3,  
        batch_size=128,  
        num_classes=num_classes,  
        num_features=sequence_length,  
        penalty_parameter=args.penalty_parameter,  
    )  
    model.train(  
        checkpoint_path=args.checkpoint_path,  
        epochs=10000,  
        log_path=args.log_path,  
        train_data=mnist.train,  
        test_data=mnist.test,  
    )
```