

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

**«Інтелектуальна система розпізнавання та
сегментація пухлин зі знімків томографії за
допомогою штучного інтелекту»**

**Завідувач
випускаючої кафедри**

Довбиш А.С.

Керівник роботи

Коробов А.Г.

Студента групи ІН-М.02

Полушкін В.В.

Нормоконтроль

Проценко О.Б.

СУМИ 2021

Сумський державний університет

(назва вузу)

Факультет ЕЛІП Кафедра Комп'ютерних наук

Спеціальність «122 -Комп'ютерні науки»

Затверджую:

зав.кафедрою _____

“ _____ ” _____ 20__ р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Полушкіну Віталію Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Інтелектуальна система розпізнавання та сегментація пухлин зі знімків томографії за допомогою штучного інтелекту

затверджую наказом по інституту від “ _____ ” _____ 20__ р. № _____

2. Термін здачі студентом закінченого проекту (роботи) 28.10.2021

3. Вхідні данні до проекту (роботи) Знімки томографії для дослідження та порівняння

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналітичний огляд. 2) Вибір методу рішення. 3) Розробка інформаційного та програмного забезпечення нейронної мережі

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Рисунки моделей, результати роботи

(підпис)

КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назва етапів дипломного проекту (роботи) | Термін виконання проекту (роботи) | Примітка |
|-------|-------------------------------------------------------------|-----------------------------------|----------|
| 1. | <i>Аналітичний огляд</i> | <i>10.09.2021</i> | |
| 2. | <i>Вибір методу рішення.</i> | <i>17.09.2021</i> | |
| 3. | <i>Розробка інформаційного та програмного забезпечення</i> | <i>10.10.2021</i> | |
| 4. | <i>Оформлення пояснювальної записки до дипломної роботи</i> | <i>29.09.2021</i> | |

Студент – дипломник

_____ (підпис)

Керівник проекту

_____ (підпис)

РЕФЕРАТ

Записка: 44 стор., 23 рис., 1 додаток, 25 джерел.

Об'єкт дослідження — процес сегментування пухлин на зображенні за умов неповної визначеності обумовленої довільними початковим умовами формування зображень МРТ.

Мета роботи — підвищення функціональної ефективності системи компютерної томографії пухлин шляхом створення інформаційної технології машинного навчання.

Методи дослідження — методи кластеризації даних та машинне навчання.

Результати — розроблено алгоритм та програмне забезпечення системи сегментації зображень. При цьому реалізовано методу глибокого навчання для виявлення пухлин і продемонстровано перевагу продуктивності всієї системи як щодо виявлення пухлин, так і щодо заходів локалізації. Розроблений алгоритм реалізовано у формі програмного продукту, створеного з використанням мови Python

ШТУЧНИЙ ІНТЕЛЕКТ, КЛАСТЕРИЗАЦІЯ ЗОБРАЖЕНЬ,
ВИДІЛЕННЯ КЛАСІВ, КОМП'ЮТЕРНА ТОМОГРАФІЯ, АНАЛІЗ
ДАННИХ, СЕМАНТИЧНІ РОЗРИВИ, ГРАДІЄНТНИЙ СПУСК

ЗМІСТ

| | |
|--------------------------------------------------------|----|
| ВСТУП | 6 |
| 1 АНАЛІТИЧНИЙ ОГЛЯД..... | 9 |
| 1.1 Згорткові нейронні мережі..... | 12 |
| 1.1.1 GoogLeNet..... | 15 |
| 1.1.2 Залишкові мержі..... | 16 |
| 1.1.3 LeNet-5 | 18 |
| 2 ВИБІР МЕТОДУ РІШЕННЯ..... | 20 |
| 2.1 Градієнтний спуск і зворотне поширення | 20 |
| 2.2 ResNet | 22 |
| 3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ. | 25 |
| ВИСНОВКИ..... | 36 |
| СПИСОК ЛІТЕРАТУРИ..... | 37 |
| ДОДАТОК..... | 39 |

ВСТУП

Кількість захворювань з онкологічним діагнозом зростає кожного року. За підрахунками вчених на 2038 людство становитиме 29.13 млн хворих і ця цифра продовжить зростати на 49 відсотків. Переважну частину цієї статистики посідають країни з низьким та середнім індексом розвитку людського потенціалу. У багатьох із них також значно зростуть показники факторів ризику, що впливають на захворюваність, таких як куріння, нездорове харчування, ожиріння та малорухливий спосіб життя.

На захворюваність на рак та смертність від нього впливатимуть зміни структури населення та умов життя людей, а також рівень здатності систем охорони здоров'я контролювати ситуацію», - вважає фахівець Агентства Фредді Брей.

Магнітно-резонансна томографія є передовою технологією в медицині для зображення повної інформації про анатомію м'яких тканин людини. Автоматичне виявлення пухлин здатне заздалегідь класифікувати зображення на нормальні та аномальні згідно з відсутністю або наявністю пухлини, відповідно.

Машинне навчання загалом і підходи глибокого навчання зокрема, відіграють ключову роль у комп'ютерному аналізі зображень мозку, сегментації, реєстрації та класифікації пухлинної тканини.

Останнім часом парадигм глибокого навчання приділяється значна дослідницька увага. Моделі глибокого навчання мають кілька шарів, які можна тренувати за допомогою будь-якого методу навчання, як під наглядом так і без нагляду людини

Останнім часом дослідження, моделі глибокого навчання досягли конкурентоспроможних результатів порівняно з класичними моделями навчання в діагностиці та класифікації зображень пухлини . Зокрема, моделі глибокого

навчання, такі як згорткові нейронні мережі (Рисунок 1.1) покращили сферу виявлення об'єктів у різних додатках.

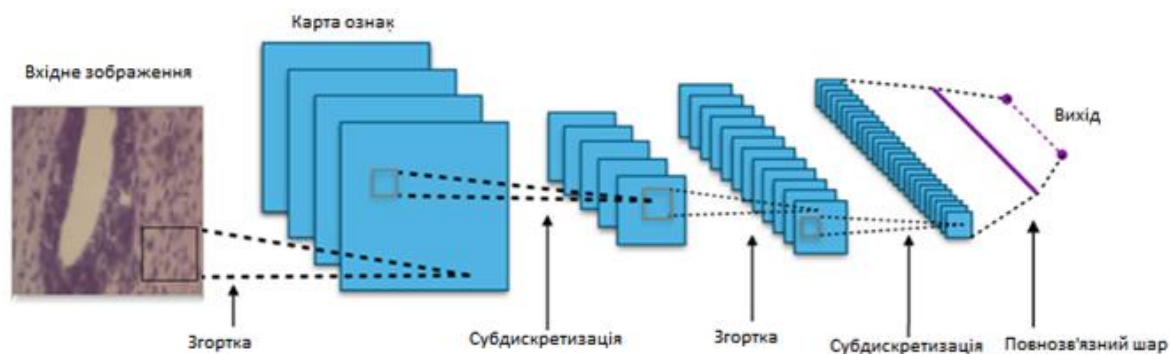


Рисунок 1.1 – Згорткова нейронна мережа

Отримання високорівневих функцій із необробленого зображення — це основне завдання глибокого навчання

Хоча кілька досліджень були спрямовані на застосування методів глибокого навчання для діагностики пухлин головного мозку, повна система автоматичного виявлення пухлин і локалізація поки що відсутня в літературі.

Крім того, інтеграція виявлення пухлини та точності локалізації в рамках єдиної системи діагностики все ще залишається відкритим завданням.

Наявність фази виявлення пухлини головного мозку, що передує локалізації пухлини в єдиній системі призводить до відкидання звичайні зображення від обробки на фазі локалізації.

Це відкриває двері для розгортання в режимі реального часу таких систем автоматичної діагностики пухлин, які заощаджують як час, так і обчислювальну потужність.

Дана робота вирішує вищезгадані проблеми шляхом розробки нової та повної системи комп'ютерної діагностики для виявлення та локалізації новоутворень з МРТ. У фазі виявлення пухлини система об'єднує згорткові нейронні мережі, які використовуються для вилучення ознак завдяки здатності вивчати функції з необроблених даних, отримуючи вихідні коди з виправленням помилок, що підтримують вектор, який використовується для класифікації ознак.

Система вважається двофазним багато модельним артефактом через її можливості виявлення та локалізації за допомогою використання різних моделей згорткових нейронних мереж

1 АНАЛІТИЧНИЙ ОГЛЯД

Виявлення пухлин пройшло значне дослідження та отримало значну кількість уваги, і за останні два десятиліття було представлено кілька методів виявлення.

Класифікація МРТ пухлин з використанням машини опорного вектора ядра, комбінація морфологічних фільтрів, дискретних вейвлет-перетворень нейронної мережі, аналізу головних компонентів та машин опорних векторів ядра була застосована для класифікації МРТ як нормальний і ненормальний. Подальші вдосконалення були реалізовані для класифікації аномального зображення як доброякісного (неракового) або злоякісного (ракового) шляхом застосування подвійного класифікатора.

Сьогодні моделі глибокого навчання мають велике значення важливість дослідження. Методи глибокого навчання забезпечують високу ефективність і можливість обробки численних МРТ з баз даних зі знімками.

Першим кроком у будь-якому дослідженні на основі даних є попередня обробка вихідних даних зображення. Попередня обробка усуває шум, забезпечуючи певний рівень парності між усіма зображеннями, які, у свою чергу, створюють далі кроки сегментації та вилучення ознак ефективний. Це передбачає виконання операцій з видалення артефакти, змінювання роздільної здатності зображення та контраст адреси відмінності, які виникають через різне обладнання для придбання і параметри.

Одним із поширених джерел шуму є упередження поля, які викликані випромінюваними низькочастотними сигналами від апарату МРТ у поєднанні з анатомією пацієнта, що в кінцевому підсумку призводить до неоднорідностей у магнітному полі. Таким чином, отримані зображення мають різну інтенсивність для тієї ж тканини, коли кожна тканина має відповідати певний рівень інтенсивності. Виникає ще одне джерело шуму з тимчасових даних. Під час курсу лікування пацієнти часто мають серію попередніх і пост-зображень. Ці серії зображень є цінними для аналітики, але для пацієнта майже неможливо

перебувати в тому самому точному положенні для попереднє та остаточне сканування. Це може ускладнити розпізнавання стан пухлини не тільки для програмного забезпечення візуалізації, але також для рентгенологів. Таким чином, зображення, зроблені за певний проміжок часу повинні бути вирівняні в процесі, відомому як реєстрація зображення.

Щоб усунути відмінності в контрасті в дослідженнях, де зображення взяті з кількох джерел і машин, зображення піддаються нормалізації кольорів або значень відтінків сірого. Нормалізація є майже універсальною в контрольованому зображенні вивчає і необхідний при застосуванні техніки машинного навчання. Нормалізація ефективно визначає а новий діапазон значень кольорів щодо інших зображень у наборі даних. Перед нормалізацією може знадобитися видалити шуми, наявні при скануванні будь-якої модальності.

Реєстрація зображення – це процес, за допомогою якого 2 зображення просторово вирівняні за допомогою комбінації геометричних перетворень, керованих оптимізатором. Зображення може бути геометрично представлений і перетворений у множинні способи, кожен з яких має свої плюси і мінуси. Важливо, що ключові біологічні орієнтири знаходяться в тому ж місці для точного порівняння та аналізу. Наприклад, навчання може мати багаточасовий (виникати протягом певного періоду часу)та/або мультимодальний (з різними контрастами) зображення. Завдяки широті різних видів .Проблем, які існують при реєстрації зображень, не існує універсального методу аналізу. У випадках, які включають мозок пухлини, особливо добре виражена мульти-формна гліобластома.

Реєстрація зображення має вирішальне значення для отримання точних морфологічні ознаки залежать від правильного вирівнювання області пухлини. Реєстрацію можна розділити на 4 основні складові: простір функцій, простір пошуку, стратегія пошуку та подібність. Кожен надає важливу інформацію, щоб визначити, яка техніка реєстрації для використання. Простір функцій використовується як основа для реєстрації, наприклад, краї, обриси пухлини.

Сегментація (Рисунок 1.2) відбувається після попередньої обробки і де зображення розділене на різні, області, що не перекривають риси текстури яких мають однакові ступені однорідності.



Рисунок 1.2 – Приклад сегментації зображення

Існує 3 основних типи сегментації, які варіюються за своїм ступенем комп'ютерної автоматизації: ручна сегментація, під наглядом і без нагляду.⁶ Для проведення ручної сегментації необхідний досвід нейрорадіолога, щоб накреслити периметр навколо області, що містить патологію, і повністю без комп'ютерної допомоги. Контрольована сегментація включає введення від користувача, вказуючи алгоритму, як виконувати та яких обмежень дотримуватися.

Найскладнішим є неконтрольований метод сегментації, який вимагає введення користувача. Неконтрольована сегментація – це область активного дослідження. Особливо проблематично це з гліомами через характер захворювання та навколишніх тканин. У деяких випадках регіони можна сегментувати під час процесу реєстрації, оскільки деякі функції вирівнювання також розпізнають окремі регіони.

Візуалізація деяких загальних фільтрів сегментації, застосовані до прикладу зображення на рисунку 1.1

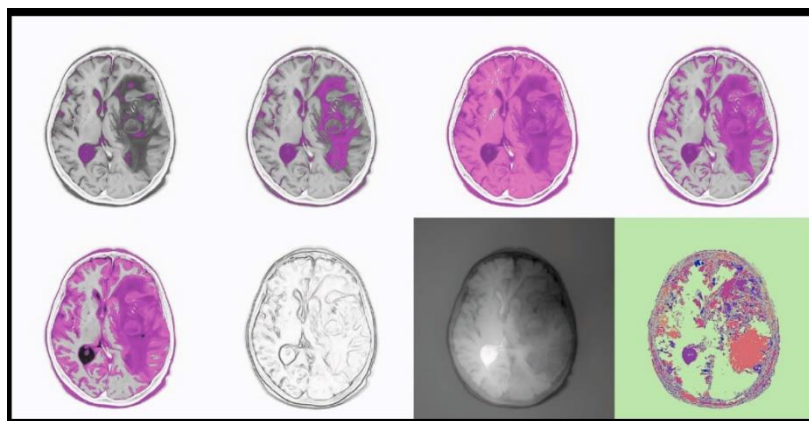


Рисунок 1.3 – Фільтри сегментації

Існує декілька етапів типового робочого процесу: отримання зображення, попередня обробка, сегментація та вилучення ознак.

1.1 Згорткові нейронні мережі

Згорткова нейронна мережа (ConvNet/CNN) – це алгоритм глибокого навчання, який може приймати вхідне зображення, призначати важливість (засвоювані ваги та упередження) різним аспектам/об'єктам зображення та мати можливість відрізнити один від іншого. Розроблений для розпізнавання візуальних шаблонів безпосередньо з піксельних зображень з мінімальною попередньою обробкою. Попередня обробка, необхідна в ConvNet, набагато рідше в порівнянні з іншими алгоритмами класифікації. Хоча в примітивних методах фільтри розробляються вручну, з достатньою підготовкою, згорткові нейронні мережі мають можливість вивчати характеристики.

Архітектура згорткових нейронних мереж аналогічна моделі зв'язку нейронів у людському мозку і була натхненна організацією зорової кори. Окремі нейрони реагують на подразники лише в обмеженій області поля зору, відомому як Рецептивне поле. Набір таких полів перекривається, щоб охопити всю візуальну область.

Метою операції згортки є вилучення з вхідного зображення високорівневих функцій (Рисунок 1.4), таких як краї.

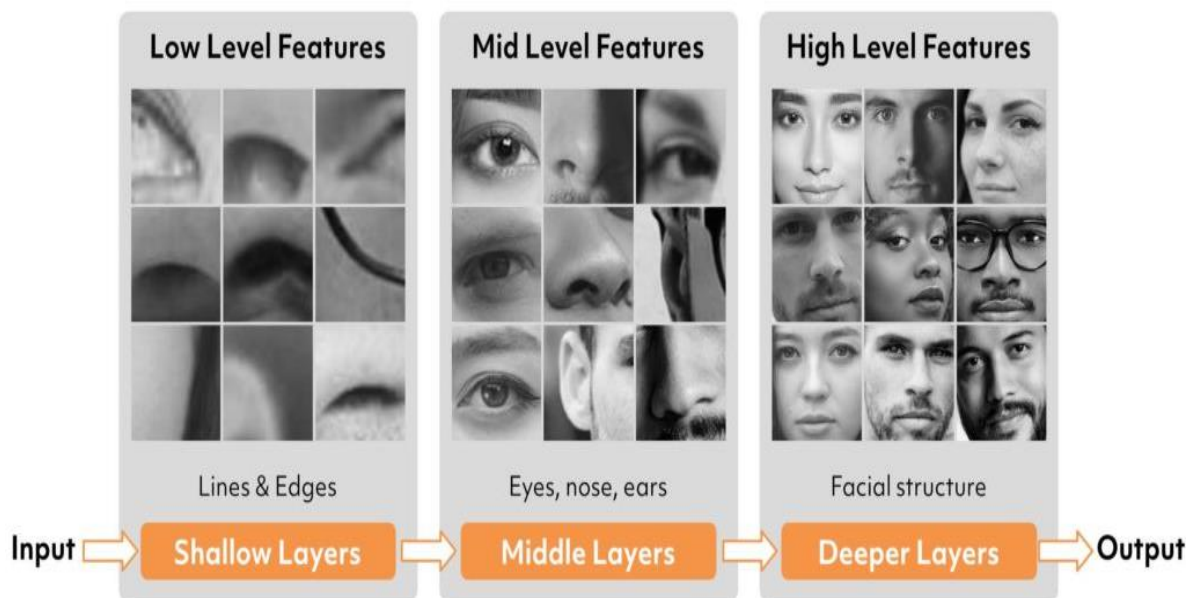


Рисунок 1.4 Приклад вилучення з вхідного зображення високорівневих функцій

Згорткові нейронні мережі не обов'язково обмежуються лише одним згортковим шаром. Зазвичай перший згортковий шар відповідає за захоплення функцій низького рівня, таких як краї, колір, орієнтація градієнта тощо. Завдяки доданим шарам архітектура також адаптується до функцій високого рівня, що дає нам повне розуміння мережі. зображень у наборі даних, як і ми.

Є два типи результатів операції — один, у якому розмірність згорнутого елемента зменшується порівняно з вхідним, а в іншому розмірність або збільшується, або залишається незмінною. Це робиться шляхом застосування дійсного заповнення у випадку першого або однакового заповнення у випадку останнього.

Подібно до шару згортки, шар об'єднання відповідає за зменшення просторового розміру згорнутого елемента. Це зменшує обчислювальну потужність, необхідну для обробки даних за рахунок зменшення розмірності. Крім того, він корисний для виділення домінуючих ознак, які є обертальними та

інваріантними положеннями, таким чином підтримуючи процес ефективного навчання моделі.

Існує два типи об'єднання: максимальне об'єднання та усереднене об'єднання (Рисунок 1.5).

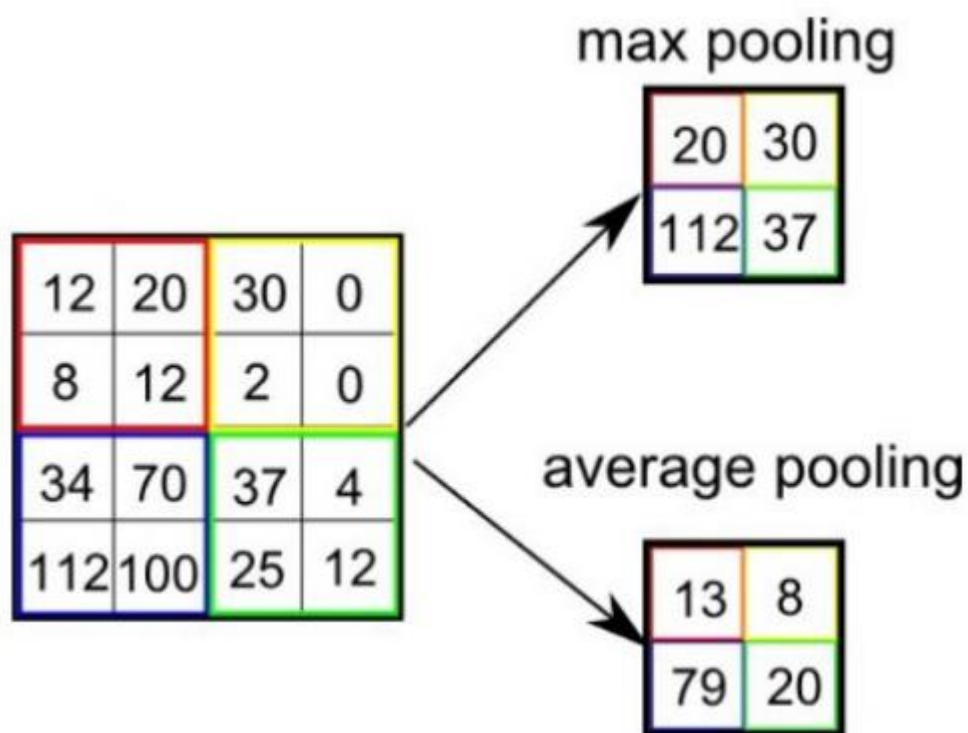


Рисунок 1.5 максимальне об'єднання та усереднене об'єднання.

Максимальне об'єднання повертає максимальне значення з частини зображення, охопленої ядром. З іншого боку, усереднене об'єднання повертає середнє значення всіх значень з частини зображення, охопленої ядром.

Максимальне об'єднання також виконує роль шумозаглушувача. Воно повністю відкидає шумні активації, а також виконує видалення шумів разом із зменшенням розмірності. У свою чергу, усереднене об'єднання просто виконує зменшення розмірності як механізм придушення шуму. Отже, ми можемо сказати, що максимальне об'єднання працює набагато краще, ніж усереднене об'єднання.

Існують різноманітні архітектури згорткових нейронних мереж, які виконали ключову роль у створенні алгоритмів. Деякі з них приведені нижче:

- LeNet
- AlexNet
- VGGNet
- ZFNet
- GoogLeNet
- ResNet

1.1.1 GoogLeNet

Мережа, навчена за допомоги ImageNet, класифікує зображення на 1000 категорій об'єктів, таких як клавіатура, миша, олівець і багато тварин. Мережа, навчена на Places365, подібна до мережі, яка навчалася в ImageNet, але класифікує зображення на 365 різних категорій місць, таких як поле, парк, злітно-посадкова смуга та вестибюль. Ці мережі вивчили різні представлення функцій для широкого діапазону зображень. Обидві попередньо навчені мережі мають розмір вхідного зображення 224 на 224.

Архітектура GoogLeNet сильно відрізняється від попередніх сучасних архітектур, таких як AlexNet і ZF-Net. Він використовує багато різних методів, таких як згортка 1×1 і глобальне середнє об'єднання, що дозволяє створювати більш глибоку архітектуру. В архітектурі ми обговоримо деякі з цих методів:

Згортка 1×1 : початкова архітектура використовує згортку 1×1 у своїй архітектурі. Ці згортки використовувалися для зменшення кількості параметрів (ваги та зміщень) архітектури. Зменшуючи параметри, ми також збільшуємо глибину архітектури.

Глобальне об'єднання (Рисунок 1.6):

У попередній архітектурі, наприклад AlexNet, повністю підключені рівні використовуються на кінці мережі. Ці повністю пов'язані шари містять більшість параметрів багатьох архітектур, які спричиняють збільшення витрат на обчислення.

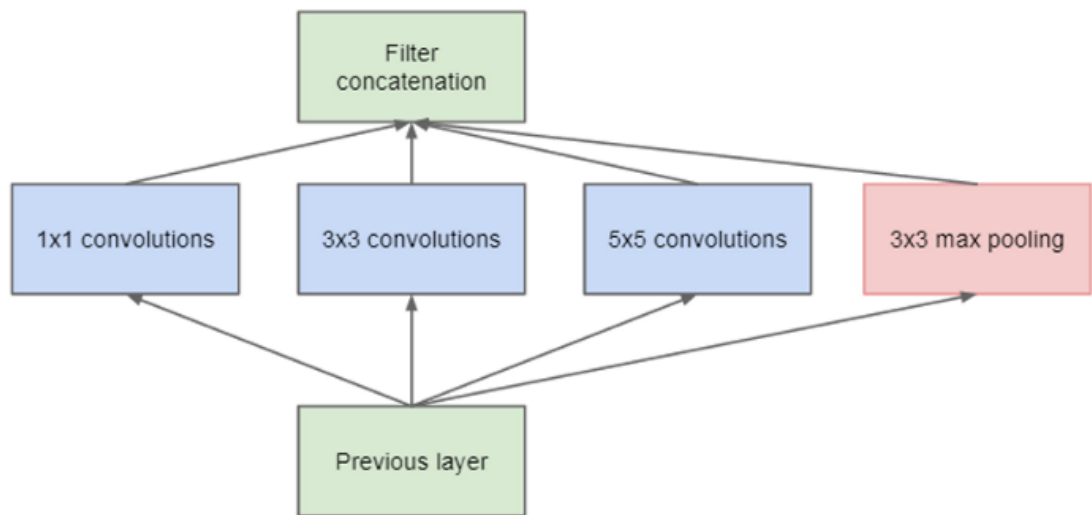


Рисунок 1.6 - глобальне об'єднання

В архітектурі GoogLeNet існує метод, який називається глобальним об'єднанням, який використовується в кінці мережі. Цей шар приймає карту об'єктів розміром 7×7 та усереднює її до 1×1 . Це також зменшує кількість параметрів, які можна навчати, до 0 і покращує точність на 0,6%.

1.1.1 Залишкові мережі

Після першої архітектури на основі CNN (AlexNet), яка виграла конкурс ImageNet 2012, кожна наступна архітектура-переможець використовує більше шарів у глибокій нейронній мережі, щоб зменшити рівень помилок. Це працює для меншої кількості шарів, але коли ми збільшуємо кількість шарів, у глибокому навчанні виникає поширена проблема, пов'язана з градієнтом зникнення/розриву. Це призводить до того, що градієнт стає 0 або занадто великим. Таким чином, коли ми збільшуємо кількість шарів, рівень помилок навчання та тестування також збільшується (Рисунок 1.7).

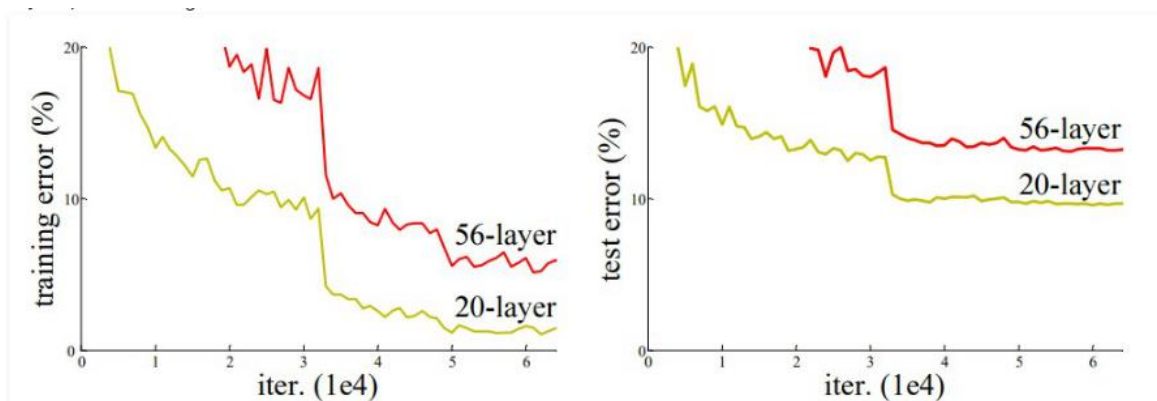


Рисунок 1.7 – рівень помилок навчання

Щоб подолати цю проблему, Microsoft ввела глибоку «остаточну» структуру навчання. Замість того, щоб сподіватися на те, що кожні кілька складених шарів безпосередньо відповідають бажаному основному представленню, вони можуть дозволити цим шарам відповідати «остаточному». Формулювання $F(x) + x$ може бути реалізовано за допомогою нейронних мереж із з'єднаннями для швидкого доступу.

Об'єднання швидкого доступу (швидкі з'єднання) пропускають один або кілька слоїв і виконують сопоставлення ідентифікаторів. Їх виходи додаються до виходу складених шарів. Використовуючи залишкову мережу, можна вирішити безліч проблем, таких як:

- ResNet відносно просто збільшує точність завдяки збільшенню глибини, чого з іншими мережами складніше досягти.
- ResNet відносно легко оптимізувати. «Прості» мережі (які просто складають шари) показують велику помилку навчання, коли глибина збільшується.

LeNet-5

LeNet-5, новаторська 5-рівнева згортка мережа Лекуна та інших у 1998 році(Рис. 1.8). Мережа має 5 рівнів з параметрами, які можна вивчати, і тому називається Ленет-5. Вона має три набори шарів згортки з комбінацією усередненого об'єднання. Після шарів згортки та усередненого об'єднання ми маємо два повністю пов'язані шари. Ця згортка класифікує цифри та букви, була застосована кількома банками для розпізнавання рукописних чисел на чеках (чеках), оцифрованих у вхідних зображеннях у відтинках сірого розміром 32x32 пікселя. Здатність обробляти зображення з більш високою роздільною здатністю вимагає більших і більш згорткових шарів, тому ця методика обмежена доступністю обчислювальних ресурсів.

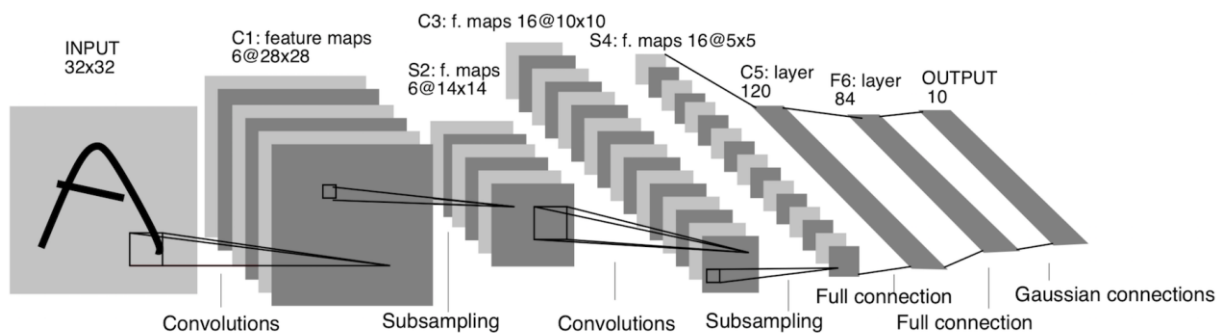


Рисунок 1.8 – модель LeNet-5

Навіть після збільшення глибини 152-шарова ResNet (11,3 мільярда операцій з плаваючою комою в секунду) має меншу складність, ніж мережа GoogLeNet

З урахуванням всіх переваг ResNet являє собою гарну компенсацію між глибиною та продуктивністю, ця мережа складається з п'яти згорткових шарів, одного усередненого об'єднання. Результати моделі мають найменшу кількість помилок 3.6% (Рис 1.9)

| Year | CNN | Developed by | Place | Top-5 error rate |
|------|--------------------|--------------------------------------------------|-------|------------------|
| 1998 | LeNet(8) | Yann LeCun et al | | |
| 2012 | AlexNet(7) | Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever | 1st | 15.3% |
| 2013 | ZFNet() | Matthew Zeiler and Rob Fergus | 1st | 14.8% |
| 2014 | GoogLeNet(19) | Google | 1st | 6.67% |
| 2014 | VGG Net(16) | Simonyan, Zisserman | 2nd | 7.3% |
| 2015 | <u>ResNet(152)</u> | Kaiming He | 1st | 3.6% |

(Рис 1.9) Порівняльна таблиця

2 ВИБІР МЕТОДУ РІШЕННЯ

Залишкова нейронна мережа (ResNet) являє собою штучну нейронну мережу, основу на пропускових з'єднаннях (skip connections), а також на зрізанні (shortcuts), для швидкого переміщення на наступні шари.

Пропуск ефективно спрощує мережу, використовуючи менше шарів з початку навчання. Задяки цьому вдається прискорити навчання, тим самим зменшується ймовірність появи проблеми впливу зникаючих градієнтів, оскільки існує менше шарів для поширення. Пропущені шари поступово відновлюються, вивчаючи простір функцій. Ближче до кінця навчання, коли всі шари розширені, він залишається ближче до різноманіття [потрібне уточнення] і, таким чином, навчається швидше. Мережа, що не має залишкових частин, може дослідити більший простір функцій.

2.1 Градієнтний спуск і зворотне поширення

У згорткових нейронних мережах є два напрямки : прямий і зворотний напрямки. Через те, що спочатку відбувається ініціалізація зваженої функції, зважена функція не може бути оцінена,. Функцію потрібно виправляти знову і знову. Використовується зворотне поширення для передачі похибки фіксації на нижні шари з верхніх шарів. А потім нижні шари можуть підтримувати зважені функції за допомогою фіксованих помилок. Постає питання, як знайти «виправлені помилки»? На допомогу приходить метод градієнтного спуску.

Метод градієнтного спуску (GD) є одним із найпоширеніших алгоритмів оптимізації в сфері машинного навчання.

Як відомо, більшість математичних моделей мають похибки через оцінку факторів. Існує концепція, яка називається функцією втрат, щоб описати загальну похибку між вибірками і

виходом функції. Ми можемо вважати, що математична модель є рівнянням лінійної регресії, і надати їй функцію гіпотези:

$$h_{\theta}(x) = \theta_0 + \theta_1 x,$$

θ_1, θ_2 - факторами рівняння

Тоді функція втрат повинна мати вигляд:

$$J(\theta_0, \theta_1) = \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2$$

Метод градієнтного спуску спрямований на оптимізацію математичної моделі та функції втрат, що означає зміну коефіцієнтів вихідних функцій і зменшення значення функції втрат. Оскільки ми хочемо зменшити значення функції втрат, ми повинні спочатку аналізувати зміни функцій. Часткові похідні є хорошим варіантом для цього. Тут слід згадати деякі концепції ініціалізації методу градієнтного спуску. Градієнт, який вимірює тенденцію цільової функції, можна розглядати як часткову похідну результату θ_i до втрат функції як градієнт. Якщо цільова функція є опуклою функцією, її можна оптимізувати.

Отже, загальний алгоритм градієнтного спуску такий:

- Ініціалізація $\theta_1, \theta_2, \dots, \theta_n$ в цільових функції, кінцевою відстанню рекурсії ϵ , з розміром кроку - α
- Розрахувати градієнт, часткову похідну функції втрат цільової функції
- Якщо $J(\theta_1, \theta_2, \dots, \theta_n) \leq \epsilon$, зупинити виконання, в іншому випадку продовжувати процес
- використовуємо градієнт, помножений на розмір кроку α
- поновити усі θ до $\theta_i = \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta_1, \theta_2, \dots, \theta_n)$ а потім повернутися до кроку 2.

Окрім оновлення вагових функцій, виправлені помилки повинні передаватися через шари. Це схоже на поєднання алгоритму градієнтного спуску, правила ланцюга та алгоритм зворотного поширення.(Рисунок 2.1)

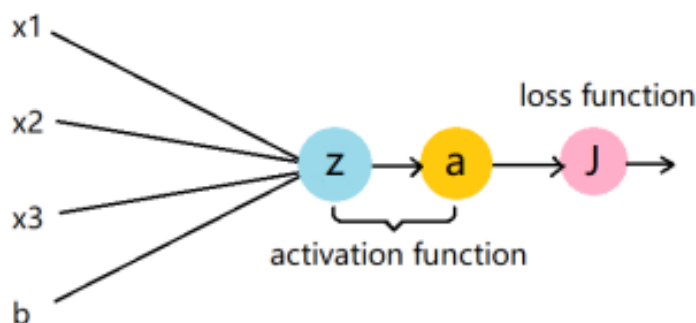


Рисунок 2.1 - Приклад зворотного поширення в одній розрахунковій одиниці

Ми припускаємо, що навчальна вибірка (x,y) , x є вхідні дані, що проходить через функцію активації $h_{w,b}(x)$. І розрахункова одиниця отримує вихідні дані "а", а потім результат "а" через функцію втрат . Результатом функції втрат є вартість J

Представлений алгоритм згорткової нейронної мережі, який дуже популярний у галузі машинного навчання та розпізнавання зображень, що поєднує в собі механізм зворотного поширення та метод градієнтного спуску.

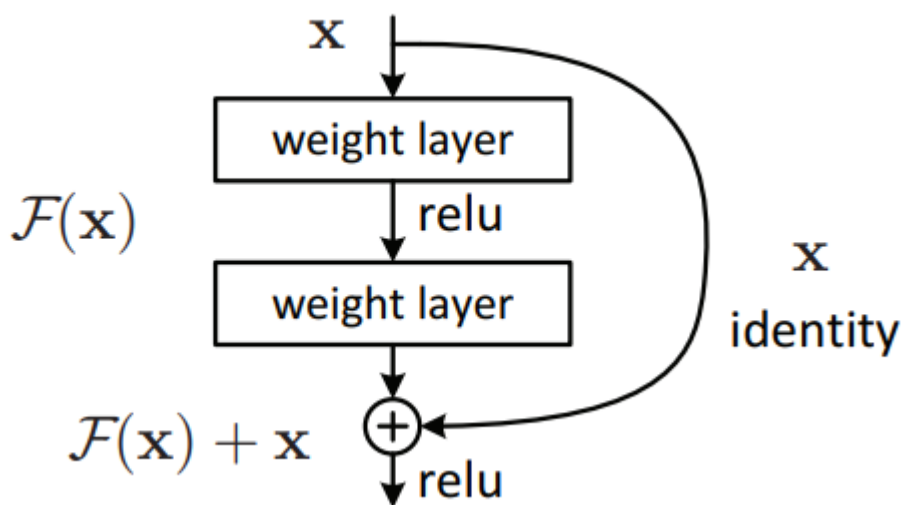
2.2 ResNet

Щоб вирішити проблему градієнта, що зникає, ця архітектура ввела концепцію під назвою залишкова мережа. У цій мережі ми використовуємо

техніку, яка називається пропуском з'єднань. З'єднання пропуску пропускає навчання з кількох шарів і підключається безпосередньо до виходу.

Підхід, який лежить в основі цієї мережі, полягає в тому, що замість того, щоб шари вивчали базове відображення, ми дозволяємо мережі відповідати залишковому відображенню. Отже, замість скажімо $H(x)$, початкове відображення

Нехай мережа $F(x) := H(x) - x$, що в результаті дає нам $H(x) := F(x) + x$.



(Рисунок 2.1) Схема моделі

Перевага додавання цього типу з'єднання з пропуском полягає в тому, що якщо який-небудь шар погіршує продуктивність архітектури, він буде пропущено шляхом регуляризації. Таким чином, це призводить до навчання дуже глибоких нейронних мереж без проблем, викликаних зникаючим градієнтом.

Існує подібний підхід, який називається “highway networks”, у цих мережах також використовується пропуск з'єднання. Подібно до LSTM, ці з'єднання пропуску також використовують параметричні вентилялі. Однак ця архітектура не забезпечує кращу точність, ніж архітектура ResNet.

ResNets впроваджуються майже у всі нові технології AI для створення найсучасніших систем. Принцип роботи ResNets полягає в тому, щоб побудувати більш глибоку мережу в порівнянні з іншими простими мережами і одночасно знайти оптимізовану кількість шарів, щоб заперечити проблему зникаючого градієнта. ResNet також було розгорнуто в наборі даних CIFAR-10, результати чудові, а його архітектура дала рейтинг помилок у топ-5 з 3,30%, таким чином мережа займає найвищі позиції в конкурсі ILSVRC.

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

Даний етап зосереджений на створенні програмного забезпечення нейронної.

Створювану програму можна розділити на декілька етапів(Рис 5):

- візуалізація даних;
- навчання моделі класифікатора;
- сегментація;
- оцінка мережі;

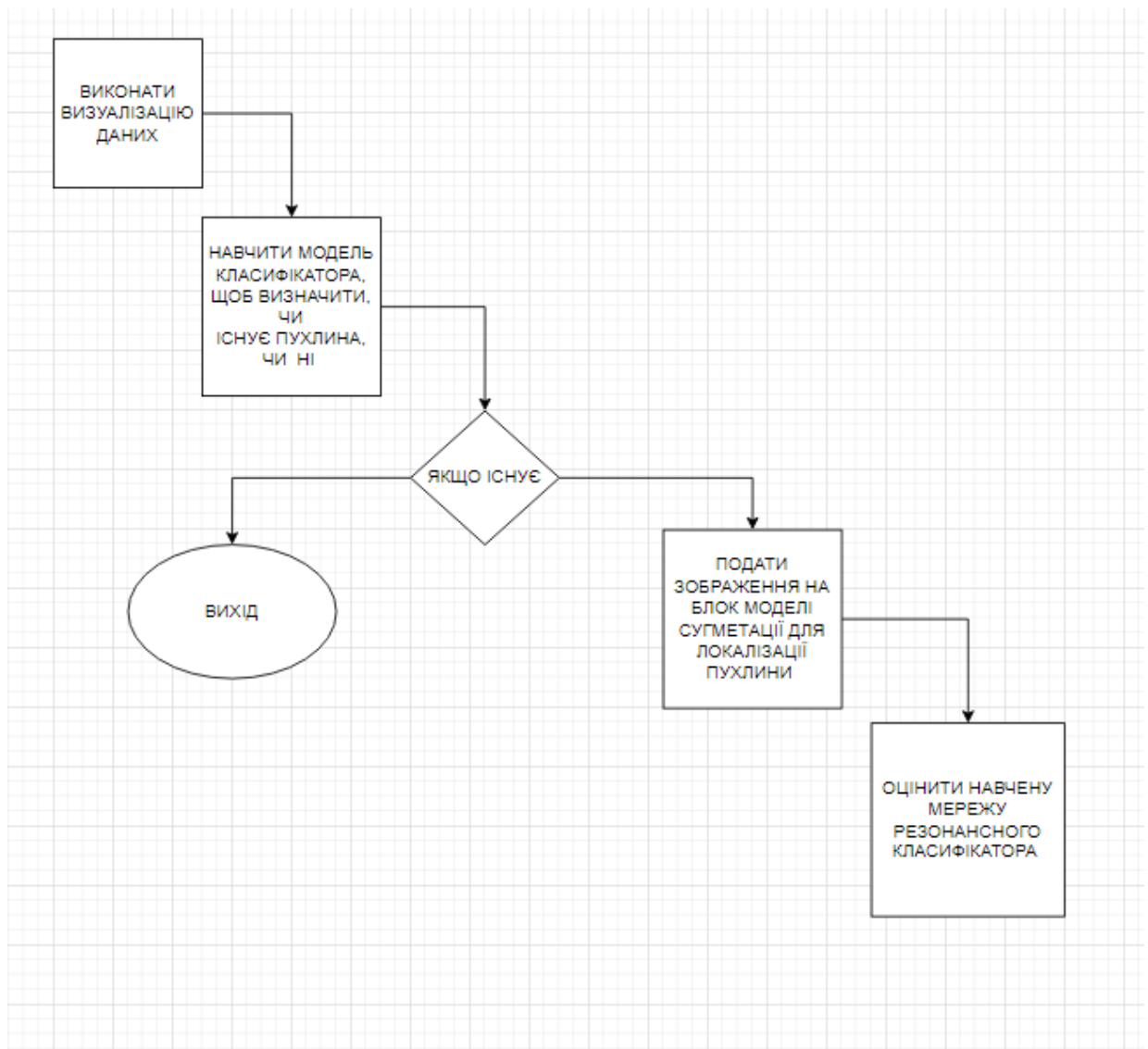


Рисунок 3.5 – Блок схема програми

Для початку візуалізації даних було створено таблицю даних, що маю ідентифікатор зображення, шлях до зображення, шлях до створюваної маски щоб ми могли відобразити зображення за допомогою функції matplotlib та відповідно сама маска.

| | | | |
|-----------------------|---------------------------------------------------|---------------------------------------------------|---|
| TCGA_DU_6401_19831001 | TCGA_HT_A61B_19991127/TCGA_HT_A61B_19991127_86... | TCGA_HT_A61B_19991127/TCGA_HT_A61B_19991127_86... | 0 |
| TCGA_DU_6401_19831001 | TCGA_HT_A61A_20000127/TCGA_HT_A61A_20000127_87... | TCGA_HT_A61A_20000127/TCGA_HT_A61A_20000127_87... | 0 |
| TCGA_DU_6401_19831001 | TCGA_HT_A61B_19991127/TCGA_HT_A61B_19991127_87... | TCGA_HT_A61B_19991127/TCGA_HT_A61B_19991127_87... | 0 |
| TCGA_DU_6401_19831001 | TCGA_HT_A61A_20000127/TCGA_HT_A61A_20000127_88... | TCGA_HT_A61A_20000127/TCGA_HT_A61A_20000127_88... | 0 |
| TCGA_DU_6401_19831001 | TCGA_HT_A61B_19991127/TCGA_HT_A61B_19991127_88... | TCGA_HT_A61B_19991127/TCGA_HT_A61B_19991127_88... | 0 |

Рисунок 3.6 – Таблиця даних

Побудуємо стовпчасту діаграму, яка показує, скільки існує зразків від хворих пацієнтів, а також від здорових пацієнтів

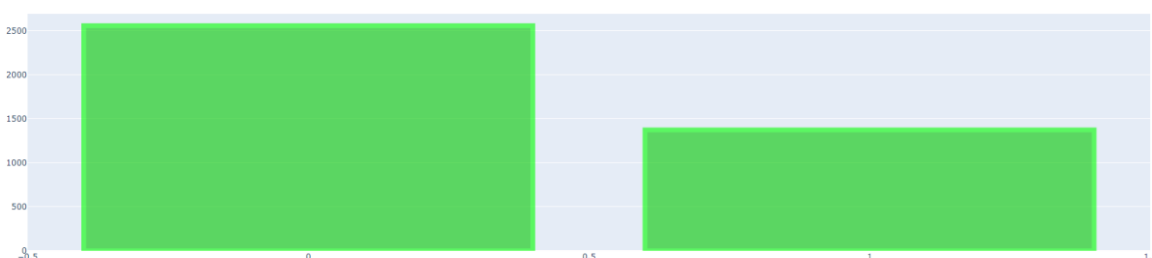


Рисунок 3.7 – Діаграма зразків

Згодом, наступне що треба побачити реальні зображення МРТ, фактичну маску та нанести обидва зображення один на одного. Сканованні МРТ і побудовані фактичні маски, пов'язаної зі знімками.

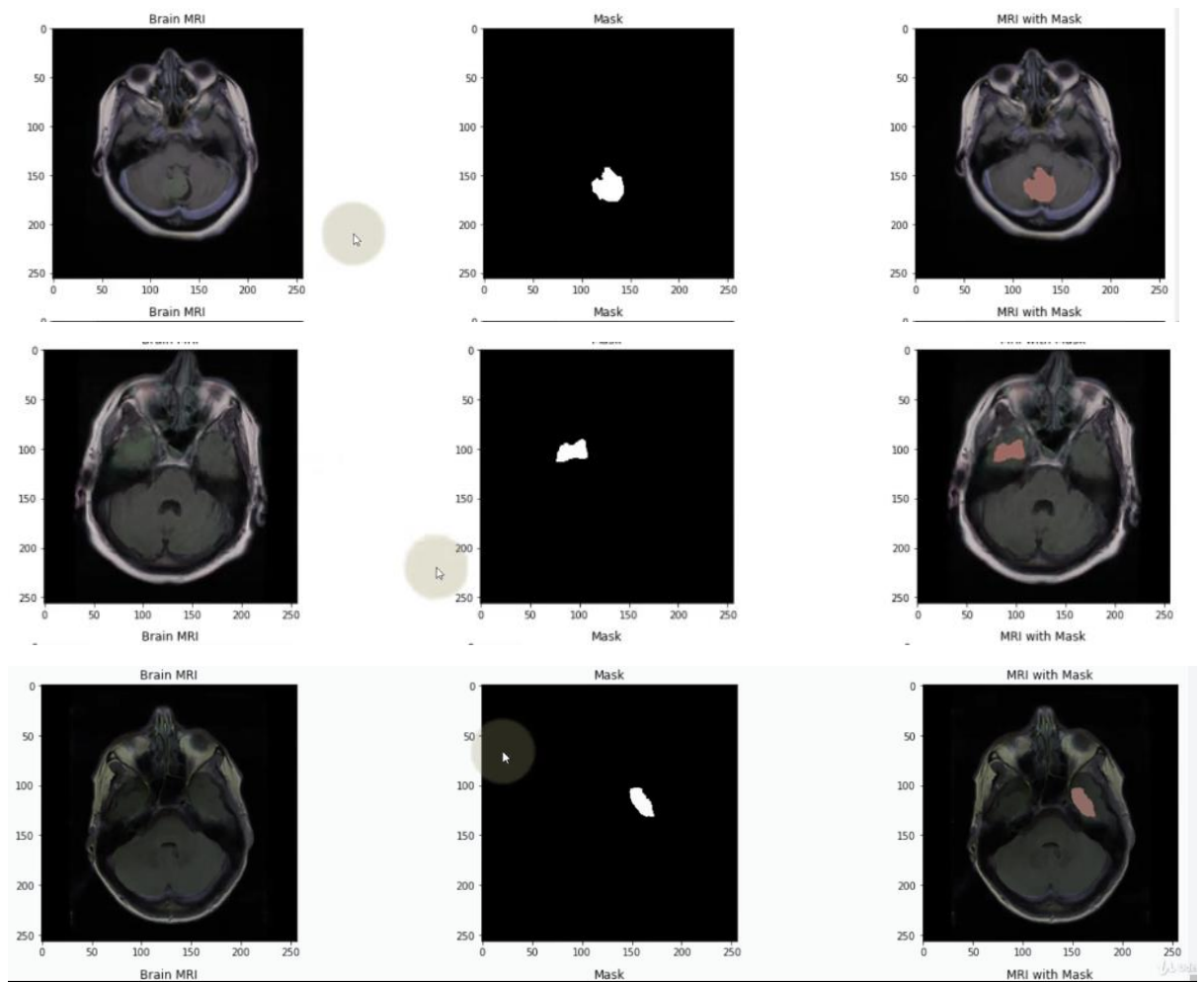


Рисунок 3.8 – Зображення нанесені один на одного

Отже, ми маємо шар того, що ми називаємо ядрами або детекторами функцій, і відповідальність цих шарів полягає в тому, щоб витягти особливості із зображення, ми передаємо дані до наступного шару, цей шар відповідає за зменшення розміру карт об'єктів, як наприклад із 64 на 64, до 32 на 32. І тепер, коли у нас є ці в основному зменшені карти об'єктів, ми можемо взяти всі ці карти об'єктів і вирівняти їх. Далі можемо подавати вперед щільну штучну нейронну мережу, щоб виконати класифікацію.

Далі потрібно розділити дані для навчання та тестування, а також потрібно змінити масштаб зображення, нормалізувати їх і відвести 15% розподілу перевірки, тому дані тестування будуть використані для тестування моделі після завершення процесу навчання, тому ідея, яка лежить в основі

цього, щоб переконатися, що модель здатна узагальнювати, а не запам'ятовувати.

Тому ми робимо так, що навчаючи мережу, ми будемо використовувати набір даних перехресної перевірки аби оцінити продуктивність тренованої моделі як програми навчання під час ходу навчання.

Однак якщо досягнуто стану з помилкою під час навчання, це означає, що мержа починає запам'ятовувати дані а не узагальнювати.

Під час компілювання моделі треба використати раннє зупинення, щоб вийти з навчання, якщо втрата підтвердження не зменшується навіть після декількох епох.

Епоха означає навчання нейронної мережі всіма навчальними даними за один цикл. В епоху ми використовуємо всі дані рівно один раз. Прохід вперед і назад разом зараховуються як один прохід.

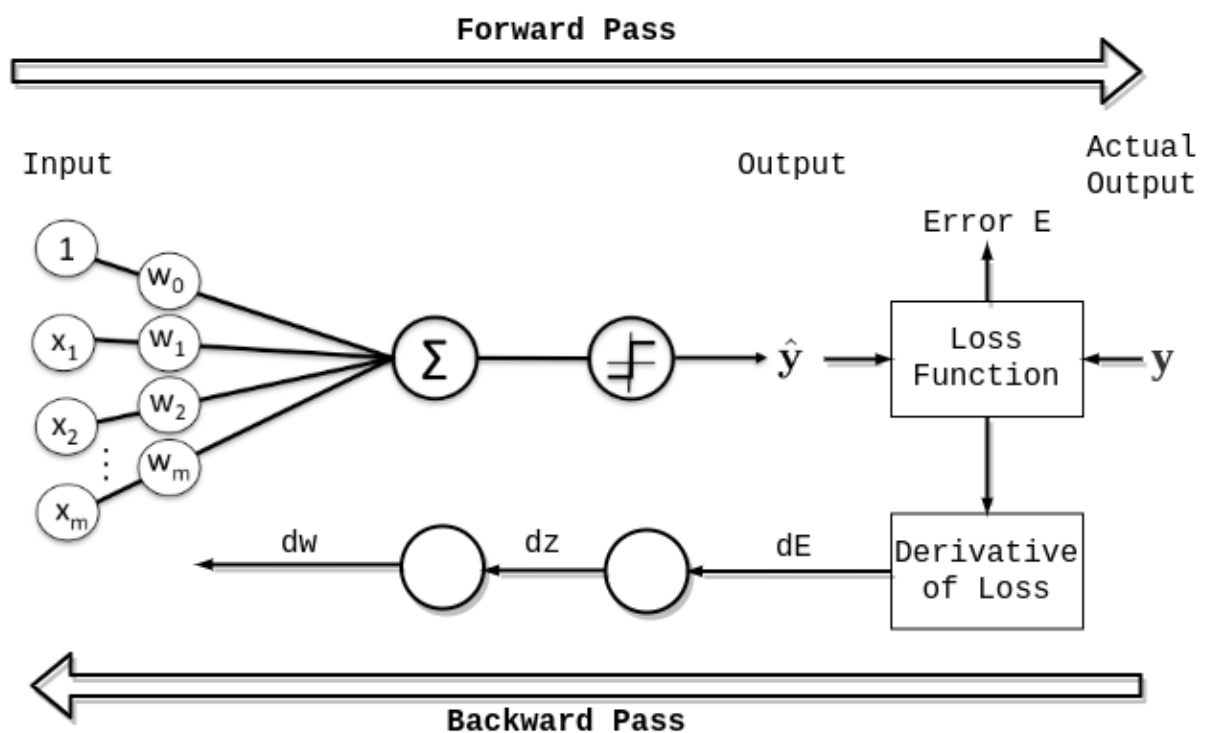


Рисунок 3.8 – Епоха мережі

Іноді епоху змішують з ітерацією. Щоб прояснити поняття, давайте розглянемо простий приклад, де ми маємо 1000 точок даних, як показано на малюнку нижче:

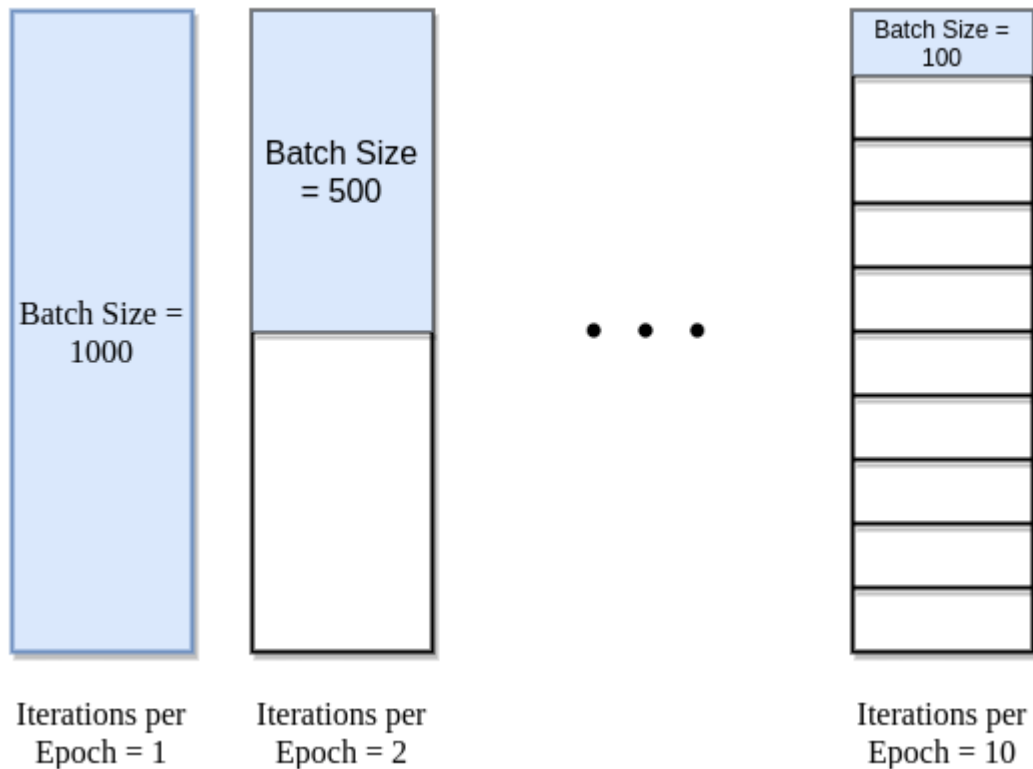


Рисунок 3.9 – Пакети мережі

Якщо розмір пакету становить 2000, ми можемо завершити епоху за одну ітерацію. Аналогічно, якщо розмір пакету становить 1000, епоха займає дві ітерації. Отже, якщо розмір пакету дорівнює 1000, для завершення епоси потрібно 100 ітерацій. Просто для кожної епохи необхідна кількість ітерацій, помножена на розмір пакету, дає кількість точок даних.

Ми можемо використовувати кілька епох у навчанні. У цьому випадку нейронна мережа подає одні й ті ж дані більше одного разу.

А далі потрібно зберегти найкращу модель з найменшими втратами перевірки. Так як, тренування моделі досить довгий процес для того щоб скоротити час краще зберегти архітектуру моделі у файлі json для подальшого використання.

Дані моделі та ваги завантажуються із збережених файлів і створюється нова модель. Важливо скопіювати завантажену модель перед її використанням. Це робиться для того, щоб прогнози, зроблені за допомогою

Для сегментації нам потрібно виконати декілька кроків:

- Нормалізація пакету згортки
- блок res з наступним об'єднанням
- кінцевий результат

Отже, спочатку у нас є етап номер один, який полягає в нормалізації пакету згортки. Він виконується між шарами нейронної мережі, а не в необроблених даних. Це виконується за допомогою міні-пакетів замість повного набору даних. Він служить для прискорення навчання та використання вищих темпів навчання, полегшуючи навчання.

Пакетна норма – на зображенні (Рис 12) , представленою червоною лінією – застосовується до виходу нейронів безпосередньо перед застосуванням функції активації.

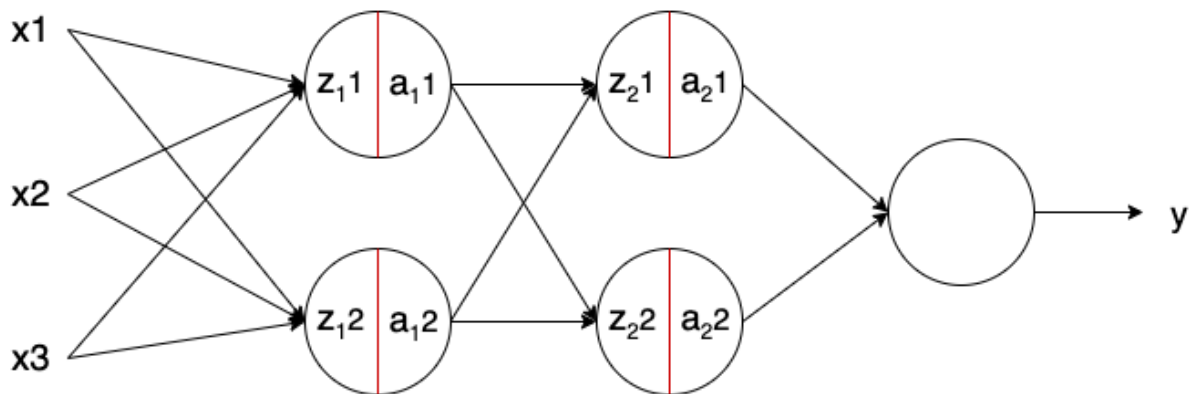


Рисунок 3.13 – Пакетна норма

Зазвичай нейрон без пакетної норми обчислюється таким чином:

$$z=g(w,x)+b;$$

$$a=f(z)$$

вважаючи $g()$ лінійним перетворенням нейрона, w вагами нейрона, b зміщенням нейронів і $f()$ функцією активації.

В кінці ми отримуємо модель згорткової сегментації у форматі hdf5, яка гарантує, що вихідна маска матиме той самий розмір, що й вхідне зображення.

Оцінемо ефективність навченої моделі. Треба створити фрейм даних для результату. Об'єднаймо фрейм даних, що містить прогнозовані результати, з вихідними даними тесту. Зчитуємо зображення та конвертуємо їх у формат RGB.

Отримаємо маску для зображення а також маску передбачення. Нанесемо маску на зображення(Рисунок 3.14)

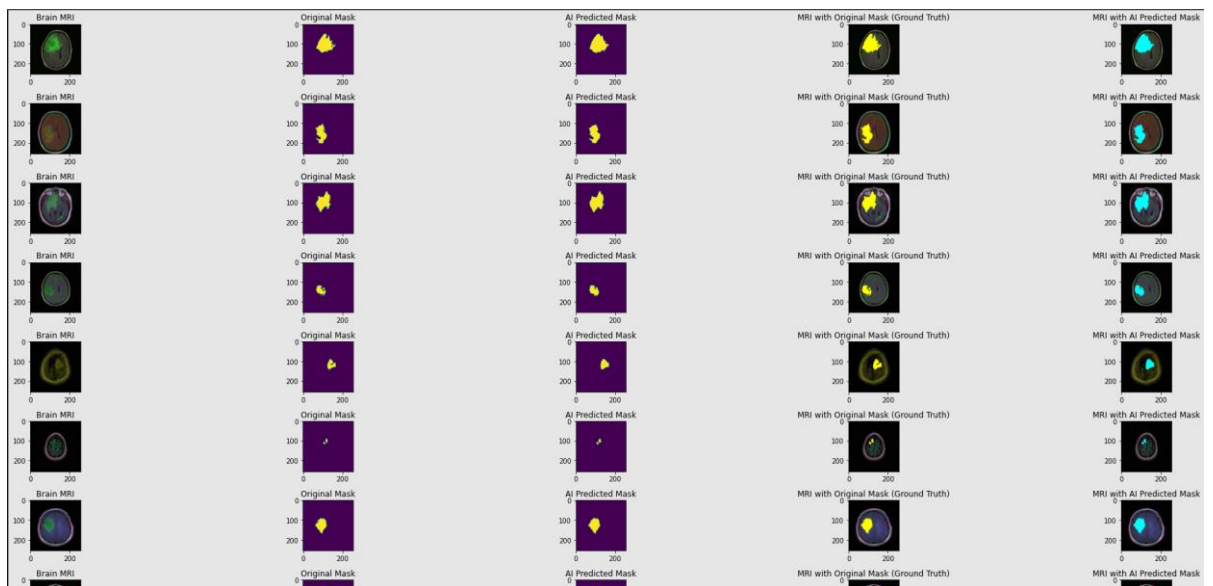


Рисунок 3.14 – порівняльна таблиця

В результаті отримуємо таблицю із зображеннями. У першому стовпці оригінальне зображення. У другому область, яка була виділена доктором. У третьому, маска створена мережею. Наступні два стовпці містять маски, поверх оригінального зображення.

Коли наші класи надзвичайно незбалансовані, це означає, що клас або деякі класи домінують над зображенням, тоді як деякі інші класи становлять лише невелику частину зображення. На жаль, дисбаланс класів поширений у багатьох

реальних наборах даних, тому його не можна ігнорувати. Перевіримо результати роботи мережі програмно.

Для програмної оцінки моделі семантичної сегментації було використано метрику Intersection-Over-Union. Intersection-Over-Union (IoU), також відома як Міра Жаккара (коефіцієнт флористичної спільності). Міра Жаккара, є однією з найбільш часто використовуваних метрик у семантичній сегментації... і не дарма. IoU — це дуже простий показник, який надзвичайно ефективний.

IoU — це площа перекриття між прогнозованою сегментацією та основною істиною, поділена на площу об'єднання між передбачуваною сегментацією та основною істиною, як показано на рисунку 3.15.

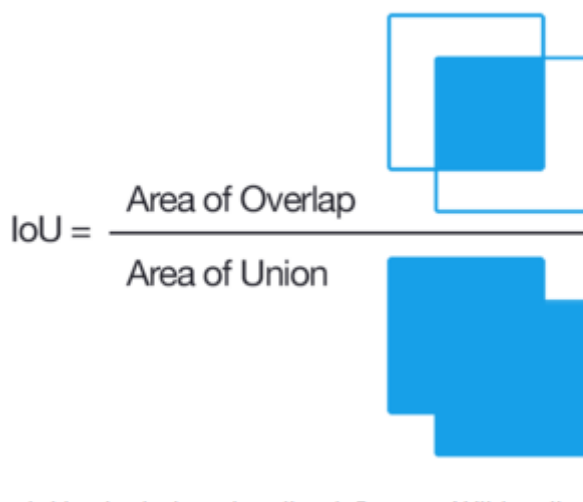
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Рисунок 3.15. - Міра Жаккара

Цей показник коливається в діапазоні від 0–1 (0–100%), при цьому 0 означає відсутність перекриття, а 1 означає сегментацію, яка повністю перекривається. Чим нижчий показник, тим гірший результат прогнозу.

Результати застосування даної метрики для 25 зображень продемонстровані на рисунку 3.16

```
# An example of first bounding box
first_bb_points = [[125, 105], [220, 105], [220, 195], [125, 195]]
stencil = numpy.zeros(img.shape).astype(img.dtype)
contours = [numpy.array(first_bb_points)]
color = [255, 255, 255]
cv2.fillPoly(stencil, contours, color)
result1 = cv2.bitwise_and(img, stencil)
result1 = cv2.cvtColor(result1, cv2.COLOR_BGR2RGB)
plt.imshow(result1)

# An example of second bounding box
second_bb_points = [[125, 105], [220, 105], [220, 195], [125, 195]]
stencil = numpy.zeros(img.shape).astype(img.dtype)
contours = [numpy.array(second_bb_points)]
color = [255, 255, 255]
cv2.fillPoly(stencil, contours, color)
result2 = cv2.bitwise_and(img, stencil)
result2 = cv2.cvtColor(result2, cv2.COLOR_BGR2RGB)

intersection = numpy.logical_and(result1, result2)
union = numpy.logical_or(result1, result2)
iou_score = numpy.sum(intersection) / numpy.sum(union)
print('Intersection Over Union is %s' % iou_score)
```

Intersection Over Union is 1.0
Intersection Over Union is 0.9952308944980808
Intersection Over Union is 0.999116438093043
Intersection Over Union is 0.9954759081126191
Intersection Over Union is 0.984346633315851
Intersection Over Union is 0.9974529780564263
Intersection Over Union is 0.9934812487728254
Intersection Over Union is 0.9980538065254723
Intersection Over Union is 0.9989310937201756
Intersection Over Union is 0.9989559972159926
Intersection Over Union is 0.9999618145715595
Intersection Over Union is 0.994566748508825
Intersection Over Union is 0.9960489924930858
Intersection Over Union is 0.9997709136726356
Intersection Over Union is 0.9971149939049166
Intersection Over Union is 0.9980425270591847
Intersection Over Union is 0.9938927664974619
Intersection Over Union is 0.9963890321812626
Intersection Over Union is 0.9999618291472632
Intersection Over Union is 0.9943845190481961
Intersection Over Union is 0.9928931386484042
Intersection Over Union is 0.9986775667227699
Intersection Over Union is 0.9966489484631385
Intersection Over Union is 0.9895555015788196
Intersection Over Union is 1.0

Рисунок 3.16. – Результати метрики

Можемо переконатись, що передбачення моделі працюють. У деяких випадках розроблена мережа отримує 100% співпадіння. Як бачимо результат згенерований мережею досить послідовний, а також результат передбачуваної маски збігається з оригінальною. Для побудови матриці невідповідностей використаємо функцію plt (Рис 3.17)

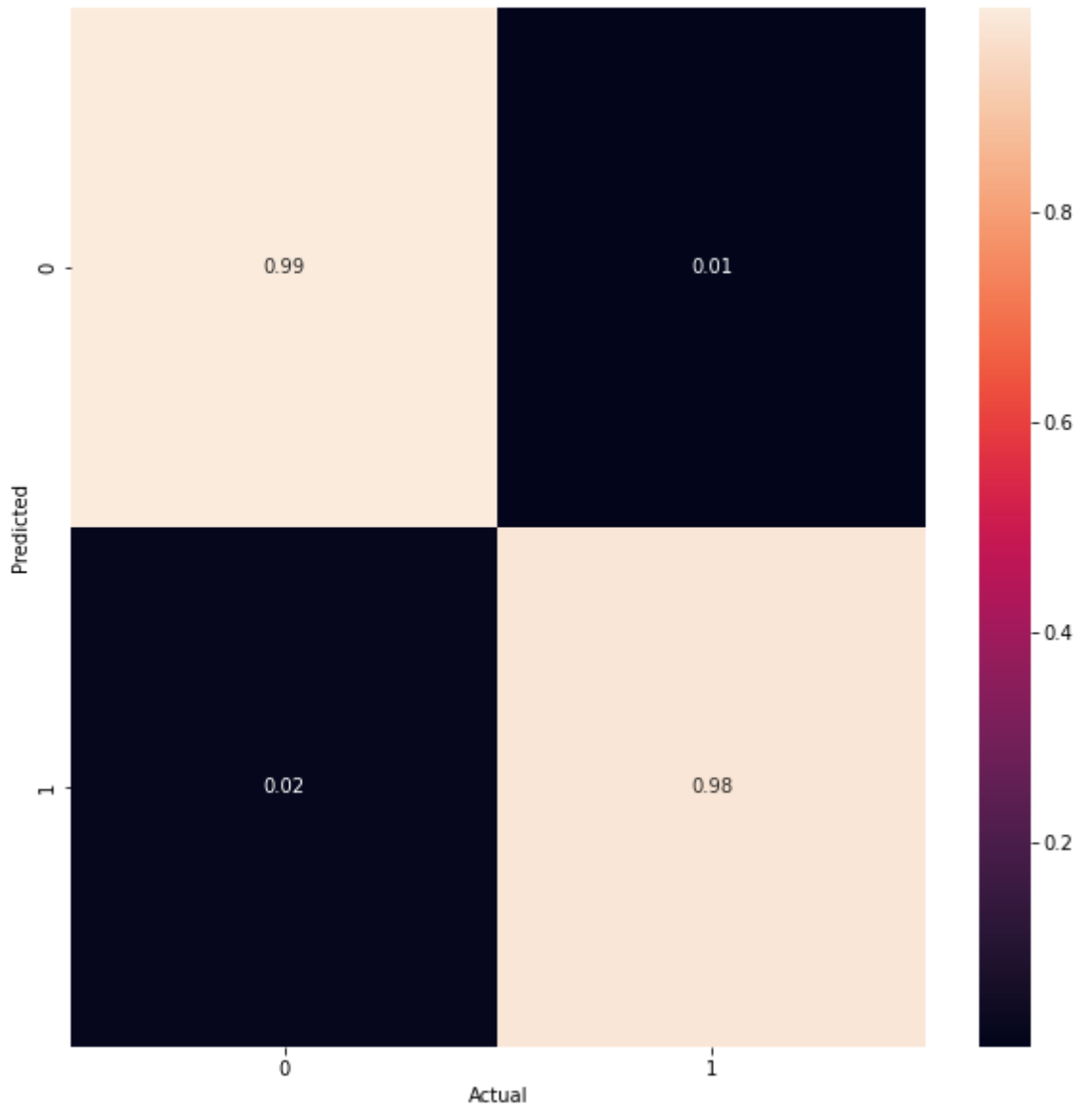


Рисунок 3.17 – Матриця невідповідностей

Похибка становить лише 2 відсотки для класу з пухлиною, і 1 відсоток для класу без пухлин.

ВИСНОВКИ

Аналіз зображень МРТ значно просунувся після появи комп'ютерного зору та комп'ютерної графіки. Паралельно було зроблено багато проривів, які призвели до створення різноманітних інструментів для дослідження. Штучний інтелект застосовується в багатьох областях, включаючи МРТ-аналіз зображень, який зараз рухається у прискореному темпі, як нові дослідження, що засновані на глибокому навчанні. Таке застосування штучного інтелекту, безсумнівно, відкриє нові області досліджень і дослідів, особливо для важких захворювань, таких як пухлини.

Це дослідження зосереджено на згорткові нейронні мережі які набули популярності серед дослідників за об'єкт розпізнавання та біологічної сегментації зображення.

Розроблена система діагностики має два етапи для виявлення і локалізація пухлини. Перша фаза перетворює виявлення пухлини в проблему се МРТ а потім класифікує МРТ на нормальні та аномальні зображення. Друга фаза фокусується на локалізації пухлини всередині аномальних зображень.

Запропоновано двофазну багатомодельну глибину система на основі навчання для виявлення пухлин та локалізація з МРТ. Емпірична робота показала, що метод досягнув точності 98%. Оцінювали фазу локалізації пухлини використовуючи базу даних BraTS.

Отримані результати довели перевагу запропонованого методу глибокого навчання для виявлення пухлин і демонструють перевагу продуктивності всієї системи як щодо виявлення пухлин, так і щодо заходів локалізації.

У деяких випадках розроблена мережа отримує 100% співпадіння. Результат згенерований мережею досить послідовний, а також зображення передбачуваної маски збігається з оригінальною з безперервним розвитком технологій нейронні мережі стають важливими частинами у житті людей. Сподіваюсь що дана робота спонукатиме інших на нові досягнення в галузі охорони здоров'я.

СПИСОК ЛІТЕРАТУРИ

1. Bob Swart. Delphi 6, ADO, and XML <http://www.pinnaclepublishing.com>
2. Ю. Пан, В. Хуанг, З. Лінь, В. Чжу, Дж. Чжоу, Дж. Вонг, З. Дін, Пухлина мозку
3. Оцінювання на основі нейронних мереж і згорткових нейронних мереж.
4. 37-а щорічна міжнародна конференція IEEE Engineering in Medicine
5. та біологічного товариства, Мілан, Італія, 25–29 серпня, EMBC, 699–702 (2015)
6. Й. Б. Ян Гудфеллоу, А. Курвіль, Глибоке навчання. Забронюйте під час підготовки до
7. MIT Press (2016). <http://www.deeplearningbook.org>
8. Т. Г. Дітеріх, Г. Бакірі, Розв'язування багатокласних навчальних завдань за допомогою
9. Вихідні коди для виправлення помилок. Й. Артіф. *Intell. Res.* 2, 263–286 (1995)
10. Т. Віндетт, Г. Ардешир, у 2003 р. Міжнародна конференція з візуального
11. Інформаційна інженерія, VIE 2003. Розширені ансамблі ЕСОС для обличчя
12. *Визнання* (IET, Guildford, 2003), с. 165–168
13. J. Kittler, R. Ghaderi, T. Windeatt, J. Matas, in *Proceedings of the 2001 IEEE*
14. Конференція комп'ютерного суспільства з комп'ютерного зору та розпізнавання образів,
15. *CVPR 2001*, том 1. Перевірка обличчя за допомогою вихідних кодів для виправлення помилок,
16. R. Ghani, in *Proceedings of the 2001 IEEE International Conference on Data*
17. Майнінг. Поєднання мічених і немаркованих даних для класифікації тексту за допомогою штучного інтелекту
18. J. Zhou, C. Y. Suen, на Восьмій міжнародній конференції з аналізу документів і розпізнавання з використанням вдосконаленого вихідного кодування з виправленням помилок: цілісний підхід
19. F. Deng, S. Guo, R. Zhou, J. Chen, *Sensor multifault diagnosis with*
20. Покращені машини опорних векторів. *IEEE Trans. Автом. наук. інж. ПП(99)*
21. Р. Гіршик, Швидкий R-CNN. Міжнародна конференція з комп'ютерного зору (ICCV), Сантьяго, Чилі, 11–18 грудня 2015 р. (IEEE, Сантьяго, 2015 р.)
22. P. Reverdy, N. E. Leonard, Оцінка параметрів у softmax моделі прийняття рішень з лінійними цільовими функціями.
23. С. М. Bishop, *Розпізнавання шаблонів і машинне навчання*. (Спрингер, 2006)

24. Б. Юань, у 2016 році 29-та Міжнародна конференція систем на чіпі IEEE (SOCC).
25. Ефективна апаратна архітектура рівня softmax у глибокій нейронній мережі

ДОДАТОК

```
data_file= pandas.read_csv('data_file.csv')

fig = go.Figure([go.Bar(x = b_d['mask'].value_counts().index, y = b_d['mask'].value_counts())])

fig.update_traces(marker_color = 'rgb(0,200,0)', marker_line_color = 'rgb(0,255,0)',
marker_line_width = 7, opacity = 0.6)

fig.show()

fig, axs = plt.subplots(6,2, figsize=(16,32))

count = 0

for x in range(6):

    i = random.randint(0, len(b_d)) # select a random index

    axs[count][0].title.set_text("Brain MRI") # set title

    axs[count][0].imshow(cv2.imread(b_d.image_path[i])) # show MRI

    axs[count][1].title.set_text("Mask - " + str(b_d['mask'][i])) # plot title on the mask (0 or 1)

    axs[count][1].imshow(cv2.imread(b_d.mask_path[i]))

    count += 1

fig.tight_layout()

b_d_f_t = b_d.drop(columns = ['patient_id'])

b_d_f_t.shape

b_d_f_t['mask'] = b_d_f_t['mask'].apply(lambda x: str(x))

train, test = train_test_split(b_d_f_t, test_size = 0.15)

getdata = ImageGetdataerator(rescale=1./255., validation_split = 0.15)

train_generator=getdata.flow_from_dataframe(
dataframe=train,
directory= './',
x_col='image_path',
y_col='mask',
```

```

subset="training",
batch_size=16,
shuffle=True,
class_mode="categorical",
target_size=(256,256))

valid_generator=getdata(
dataframe=train,
directory= './',
x_col='image_path',
y_col='mask',
subset="validation",
batch_size=16,
shuffle=True,
class_mode="categorical",
target_size=(256,256))

test_getdata=ImageGetdataerator(rescale=1./255.)

test_generator=test_getdata(
dataframe=test,
directory= './',
x_col='image_path',
y_col='mask',
batch_size=16,
shuffle=False,
class_mode='categorical',
target_size=(256,256))

basemodel = ResNet50(weights = 'imagenet', include_top = False, input_tensor = Input(shape=(256, 256, 3)))

earlystopping = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=20)

checkpointer = ModelCheckpoint(filepath="ResUNet-weights.hdf5", verbose=1, save_best_only=True)

history = model_seg.fit(training_generator, epochs = 1, validation_data = validation_generator, callbacks =
[checkpointer, earlystopping])

model_json = model_seg.to_json()

with open("ResUNet-model.json","w") as json_file:
json_file.write(model_json)

with open('URes.json', 'r') as json_file:
json_savedModel= json_file.read()

model_seg = tf.keras.models.model_from_json(json_savedModel)

model_seg.load_weights('weights_seg.hdf5')

```



```

adam = tf.keras.optimizers.Adam(lr = 0.05, epsilon = 0.1)

model_seg.compile(optimizer = adam, loss = focal_tversky, metrics = [tversky])

image_id, mask, has_mask = prediction(test, model, model_seg)

df_pred = pandas.DataFrame({'image_path': image_id, 'predicted_mask': mask, 'has_mask': has_mask})

df_pred

df_pred = test.merge(df_pred, on = 'image_path')
df_pred.head()

count = 0

fig, axs = plt.subplots(10, 5, figsize=(30, 50))

for i in range(len(df_pred)):

if df_pred['has_mask'][i] == 1 and count < 10:

# convert images to RGB format

# Get mask of mage

# Get prediction mask of mage

# Assign mask to the image

count += 1

Intersection_Over_Union = numpy.logical_and(result1, result2)

union = numpy.logical_or(first, second)

iou_score = numpy.sum(Intersection_Over_Union) / numpy.sum(union)

print('IoU is %s' % iou_score)

f_bb_p = [[125, 105], [220, 105], [220, 195], [125, 195]]
stencil = numpy.zeros(img.shape).astype(img.dtype)
contours = [numpy.array(f_bb_p)]
color = [255, 255, 255]
cv2.fillPoly(stencil, contours, color)
r1 = cv2.bitwise_and(img, stencil)
r1 = cv2.cvtColor(result1, cv2.COLOR_BGR2RGB)
plt.imshow(r1)

s_bb_p = [[125, 105], [220, 105], [220, 195], [125, 195]]
stencil = numpy.zeros(img_.shape).astype(img_.dtype)

```

```
contours = [numpy.array(s_bb_p)]
color = [255, 255, 255]
cv2.fillPoly(stencil, contours, color)
r2 = cv2.bitwise_and(img_, stencil)
r2 = cv2.cvtColor(result2, cv2.COLOR_BGR2RGB)
plt.imshow(r2)
```

```
in = numpy.logical_and(r, r2)
union = numpy.logical_or(r1, r2)
iou_score = numpy.sum(in) / numpy.sum(union)
print('Intersection over Union is %s' % iou_score)
if iou_score > 0.5:
    my_list.append(1)
else:
    my_list.append(0)
```

```
fig.tight_layout()
my_list
```