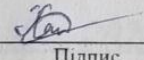
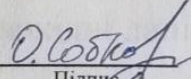
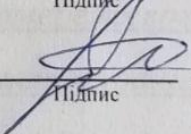


КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями

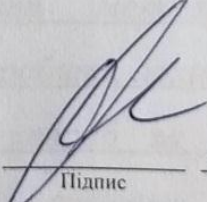
Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студент 4 курсу, група КН-19-2  І.М. Харіш
Курс, група виконавця Підпис Ініціали, прізвище
Керівник: викладач кафедри КН  О.В. Собко
Науковий ступінь, посада Підпис Ініціали, прізвище
Нормоконтроль: к.т.н., доцент кафедри КН  Р.О. Багрій
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор

01 06 2023 р.

 О.В. Бармак
Підпис Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

Освітня програма освітньо-професійна програма підготовки бакалавра

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

(підпис)

д.т.н., професор О.В. Бармак

« 06 » 03 2023 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями»

2. Завдання видано студенту Харишу Івану Миколайовичу
(прізвище, ім'я, по батькові)

3. Керівник роботи викладач кафедри КН Собко Олена Віталіївна
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від « 01 » 03 2023 р. № 5

5. Дата видачі завдання студенту: « 03 » 03 2023 р.

6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Провести аналіз предметної області нейромережевого діагностування дерматологічних хвороб, визначити особливості застосування нейронних мереж для задач класифікації зображень. Виконати аналіз існуючих рішень для подібних задач. Розробити нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями. Розробити архітектуру нейронної мережі для діагностування дерматологічних хвороб за їх зображеннями. Спроектувати структуру інформаційної системи діагностування дерматологічних хвороб й структуру відповідної бази даних. Спроектувати та створити систему що використовує розроблений метод, виконати його тестування.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником	грудень 2022	виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2023	виконано
3	Робота над розділом 1 – Характеристика предметної області нейромережевого діагностування дерматологічних хвороб та постановка задачі	січень 2023	виконано
4	Робота над розділом 2 – Нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями	березень 2023	виконано
5	Робота над розділом 3 – Програмна реалізація інформаційної системи нейромережевого діагностування дерматологічних хвороб	квітень 2023	виконано
6	Оформлення пояснювальної записки згідно вимог	травень 2022	виконано
7	Попередній захист кваліфікаційної роботи бакалавра	травень 2022	виконано
8	Захист кваліфікаційної роботи бакалавра на засіданні Екзаменаційної комісії	червень 2022	виконано

Виконавець:

студент 4 курсу, група КН-19-2

Курс, група виконавця


Підпис

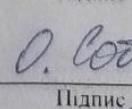
І.М. Хариш

Ініціали, прізвище

Керівник:

викладач кафедри КН

Науковий ступінь, посада


Підпис

О.В. Собко

Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: Нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-19-2 Харин Іван Миколайович

Керівник кваліфікаційної роботи бакалавра: викладач кафедри КН Собко Олена Віталіївна

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
71	36	8	25	8

Метою кваліфікаційної роботи бакалавра є розробка нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями та інформаційної системи, яка використовує розроблений метод.

Результатом виконання кваліфікаційної роботи бакалавра є створений нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями та відповідна інформаційна система діагностування дерматологічних хвороб, яка дозволяє за зображеннями ураженого фрагменту шкіри людини визначати наявні дерматологічні хвороби за допомогою нейронної мережі.

Ключові слова: діагностування дерматологічних хвороб, зображення шкіри людини, метод, нейронна мережа, нейромережа, діагностування, шкіра людини, діагностування дерматологічних хвороб.

Виконавець: студент 4 курсу, група КН-19-2 Харин І.М. Харин
Курс, група виконавця Підпис Ініціали, прізвище

Зміст

Перелік скорочень	4
Вступ.....	5
Розділ 1 Характеристика предметної області нейромережевого діагностування дерматологічних хвороб та постановка задачі	7
1.1 Аналіз предметної області	7
1.2 Особливості застосування нейронних мереж для задач класифікації зображень	11
1.3 Аналіз існуючих рішень для подібних задач	13
1.4 Мета, задачі та вимоги до реалізації програмної системи.....	17
Розділ 2 Нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями	19
2.1 Схема та кроки нейромережевого методу діагностування дерматологічних хвороб.....	19
2.2 Розробка архітектури нейронної мережі для діагностування дерматологічних хвороб.....	25
2.3 Підготовка робочих даних для системи	33
2.4 Проектування структури інформаційної системи нейромережевого діагностування дерматологічних хвороб.....	37
2.5 Вибір комбінації засобів розробки інформаційної системи нейромережевого діагностування дерматологічних хвороб	40
2.6 Висновки до розділу 2	42
Розділ 3 Програмна реалізація інформаційної системи нейромережевого діагностування дерматологічних хвороб.....	44
3.1 Структура модулів інформаційної системи й їх взаємозв'язок	44
3.2 Особливості реалізації інформаційної системи нейромережевого діагностування дерматологічних хвороб.....	49
3.3 Опис функціональних можливостей інформаційної системи	54

3.4 Дослідження ефективності застосування нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями	61
3.5 Висновки до розділу 3	65
Висновки	67
Перелік посилань.....	69
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
НМ	Нейронна мережа
ШІ	Штучний інтелект
GUI	Графічний інтерфейс користувача
ІК	Інтерфейс користувача
МН	Машинне навчання
FP	Помилково позитивний результат
FN	Помилково негативний результат
TP	Вірно позитивний результат
TN	Вірно негативний результат
ІС	Інформаційна система

Вступ

Кваліфікаційна робота бакалавра присвячена розробці нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями та ІС, яка використовує розроблений метод і дозволяє за зображеннями ураженого фрагменту шкіри людини визначати наявні дерматологічні хвороби за допомогою нейронної мережі.

Актуальність. Шкіра – це надзвичайно важливий орган в організмі людини, адже він є частим індикатором того, які процеси відбуваються в людському тілі і як це відображається на її функціональності. Тому різні негативні прояви на ділянках шкіри можуть викликати незворотні зміни в організмі і призвести до важких наслідків. Хворобу організму виявляють спершу по хворобі шкіри. Оскільки багато корисних функцій організму зосереджено в одному великому органі, то потрібно задуматися про способи захисту шкіри, виявлення відхилень в її роботі. Це можливо, якщо правильно підібрати техніку діагностування найбільш поширених дерматологічних хвороб.

Об’єкт дослідження – процес діагностування дерматологічних хвороб в людини за зображеннями фрагменту шкіри на базі архітектури нейронної мережі.

Предмет дослідження – моделі, методи, алгоритми та засоби для автоматизованого діагностування дерматологічних хвороб в людини за їх зображеннями.

Мета кваліфікаційної роботи бакалавра – розробка нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями та ІС, яка використовує розроблений метод і дозволяє за зображеннями ураженого фрагменту шкіри людини визначати наявні дерматологічні хвороби за допомогою нейронної мережі.

Завдання кваліфікаційної роботи бакалавра: Провести аналіз предметної області нейромережевого діагностування дерматологічних хвороб,

визначити особливості застосування нейронних мереж для задач класифікації зображень. Виконати аналіз існуючих рішень для подібних задач. Розробити нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями. Розробити архітектуру нейронної мережі для діагностування дерматологічних хвороб за їх зображеннями. Спроекувати структуру ІС діагностування дерматологічних хвороб й структуру відповідної бази даних. Спроекувати та програмно реалізувати систему що використовує розроблений метод, виконати його тестування.

Розділ 1 Характеристика предметної області нейромережевого діагностування дерматологічних хвороб та постановка задачі

1.1 Аналіз предметної області

Застосування нейромережевих алгоритмів, методів штучного інтелекту для автоматизації класифікації дерматологічних захворювань є актуальною проблемою, оскільки досвіду дерматологістів не вистачає для прогнозування точного діагнозу.

Для розробки нейромережевого методу діагностування шкірних захворювань потрібно проаналізувати інформацію про головні функції і різноманітні хвороби шкіри, видимі їх ознаки на тілі та інші характеристики, включаючи фотографії та додаткові параметри.

Наша шкіра – невід’ємна частина нашого організму, яка сформувалася задля того, щоб, в основному, при звичаюватися до навколишнього середовища і протистояти різним негативним впливам ззовні. А найбільшими загрозами ззовні є різні віруси, бактерії і ультрафіолетове випромінювання. І щоб захиститися від цих загроз, шкіра створила певні компоненти в собі, аби максимально зменшити негативні прояви. Захисна функція шкіри відіграє одну з провідних ролей в цьому. Серед інших важливих функцій шкіри потрібно виділити [1]:

- захист від механічних пошкоджень, захист внутрішніх органів, і мінімізація процесів старіння;
- підтримання балансу речовин в організмі, поглинання з зовнішнього середовища сонячного випромінювання і синтез вітаміну D, також, через шкіру може поглинатися до 2% кисню;
- регулювання процесів життєдіяльності, таких як терморегуляція, що супроводжується потовиділенням;
- синтез речовин, які забарвлюють нашу шкіру і захищають від ультрафіолетового випромінювання;

– шкіра транспортує в собі до 1 літра крові, що являє собою додатковий засіб депонування нашого організму;

– через шкіру ми сприймаємо дотик, температуру, тиск, тобто вона є одним з п'яти органів відчуття.

Зважаючи на аналізовану вище структуру нашої шкіри, можна дійти висновку, що вона є, без сумніву, важливим органом в життєдіяльності людини, а тому передчасне виявлення проблем з нею є першочерговим завданням.

Вивчення симптомів і ознак веде до коректного визначення природи захворювання шкіри і дозволяє провести точну діагностику. Супутні симптоми чи медична історія пацієнта не є настільки важливими, адже на зображеннях, відібраних для діагностики вони не відображаються.

Оскільки хвороби завжди залишають слід на нашій шкірі, які можна розрізнити по відтінку, розміру і інших характеристиках, їх підрозділяють у кілька категорій [2]:

– висипання, тобто плями різних кольорів, бульбочки, прищі (рисунок 1.1)

– сухість та лущення, призводить до свербіжу;

– почервоніння, дуже часто при дерматиті;

– висипання, наприклад, герпес (рисунок 1.2);



Рисунок 1.1 – Папули, або висипання



Рисунок 1.2 – Герпес

- зміна кольору шкіри, які проявляється при вітиліго (рисунок 1.3);



Рисунок 1.3 – Набуте вітиліго

- пухлини, пов'язані з раком шкіри;
- рубці.

Під час діагностики дерматолог спочатку проводить візуальний огляд пацієнта, опісля може застосувати різні тести як дерматоскопія, зразки крові або біопсія. В нейромережевому методі варто обмежитись тільки зображеннями вражених ділянок шкіри, які повинні містити хоча б мінімальний набір інформації для змоги проведення діагностики.

Щодо хвороб, які уражають наш організм, наведено опис і зображення деяких з них (рисунок 1.4), з їхніми особливостями:

- акне (вугрова хвороба) – вона проявляється утворенням прищів на обличчі, грудях і спині;
- екзема – запальне захворювання, що проявляється свербіжем, лущенням, почервонінням шкіри;
- розацеа – хронічне захворювання, бульбашки, вируби, почервонінням на обличчі;
- псоріаз – теж хронічне захворювання, яке супроводжується висипанням червоних плям, вкритих білою лускою;
- дерматит – запальне захворювання, спричинене алергією, контактом зі шкідливими речовинами або інфекцією;
- герпес – заразна хвороба, з'являється виразка на губах чи інших частинах тілі;
- вітиліго – втрата пігменту, через що з'являються червоні плями. Виливання, наприклад, герпес;
- лишай – інфекційне захворювання, проявами є висипи на шкірі від збудника грибка або вірусу.



Рисунок 1.4 – Деякі хвороби шкіри

Система діагностування для охарактеризування хвороб навчатиметься, працюючи з такими елементами, а саме:

- форма та розмір ураженого місця: опис відбувається через круги, прямокутники, інші фігури;
- колір шкіри: RGB-код;
- інтенсивність кольору, тобто яскравість пікселів на зображенні від 0 до 255;
- діаметр і площа пухирів або висипань в пікселях;

Отже, проаналізувавши вищенаведене, можна сказати, що основним параметром предметної області нейромережевого діагностування дерматологічних хвороб є зображення, що відображають ділянки шкіри, які містять прояви цих самих хвороб. Проведення діагностик таким способом міститиме багато нюансів, але нейромережа матиме більше ресурсів для перевершення людських здібностей в діагностуванні дерматологічних хвороб.

1.2 Особливості застосування нейронних мереж для задач класифікації зображень

Підходів вирішення задач класифікації графічних зображень є кілька, серед них варто виділити використання згорткових нейронних мереж [3], такі типи НМ використовуються найчастіше для оброблення саме зображень. Звичайні штучні НМ, наприкладі багат шарового перцептронну можуть бути неефективні для виконання даного завдання класифікації.

Було створено вже десятки архітектур згорткових НМ і випробувано їх в багатьох задачах комп'ютерного зору. Але найбільш ефективною наразі є архітектура EfficientNet [4]. Створена в 2019 році, вона по новому підходила до покращення точності мережі з допомогою складеного коефіцієнта, що керував розмірами ширини, глибини, роздільної здатності (рисунок 1.5). Тобто масштабування в EfficientNets відбувається набагато краще, ніж в попередніх

архітектурах. Тип такого масштабування виправданий, адже на вході мережі можуть бути зображення різної роздільної здатності, а тому витягуватися ознаки з нього будуть по-різному. Сімейство моделей EfficientNets досягає найвищої точності на наборах даних ImageNet(84,3%), CIFAR-100 (91,7%), Flowers (98,8%).

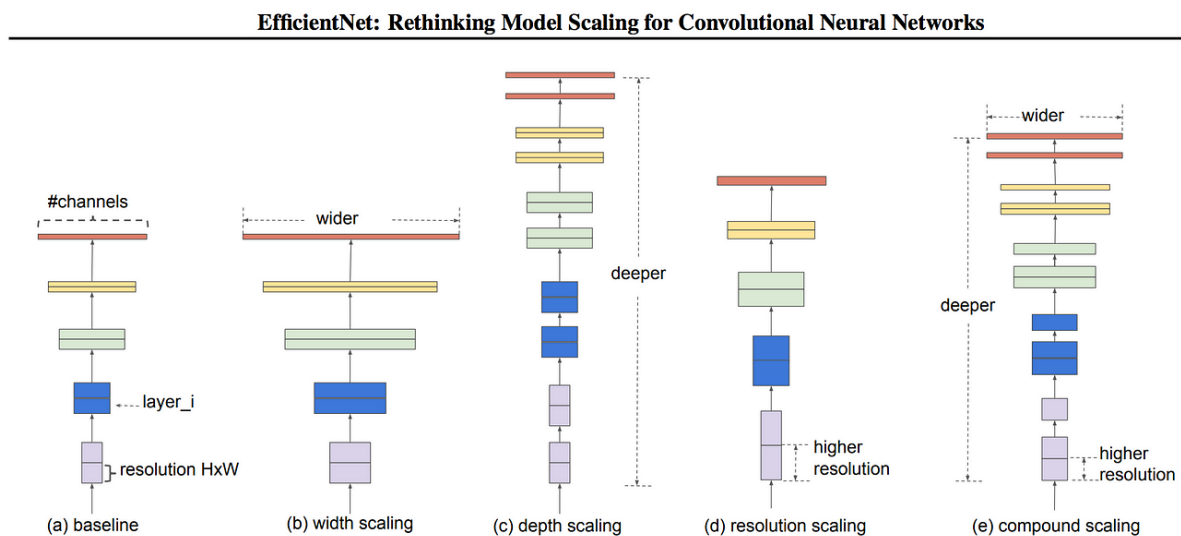


Рисунок 1.5 – Масштабування моделі в EfficientNet [4]

Переваги згорткових НМ:

- Вони вважаються одними з найкращих у задачах класифікації зображень, вирізняючись простотою і зручністю при використанні.
- Чудово можуть аналізувати картинки високої роздільної здатності, адже на відміну від інших алгоритмів не використовують об'ємну попередню обробку.
- Можливість реалізації алгоритмів роботи і навчання на графічних процесорах.
- Навчання проходить методом зворотнього розповсюдження, що є класичним способом навчання.
- Згорткові НМ здатні швидко обробляти дані, що робить їх ідеальними для великомасштабних завдань класифікації.

Недоліки:

- Навчання згорткових НМ потребує велику кількість часу, вони вимагають великої об'єм даних для обробки.

- Часто змінюються більшість параметрів мережі, від чого результат стає неоднозначним.

Згорткові НМ застосовуються повсюдно через ряд важливих особливостей як от використання графічних процесорів для пришвидшення розрахунків чи розв'язання завдань комп'ютерного зору. Тому вони набули популярності і в медицині. Діагностика симптом хвороби з допомогою зображення зарекомендувала себе чудово. Оцінюючи переваги і недоліки різних НМ, можна дійти висновку, що саме згорткові НМ найкраще справляться з поставленим завданням. Вони є точними, надійними і для їх створення та навчання представлено багато бібліотек з відкритим вихідним кодом.

1.3 Аналіз існуючих рішень для подібних задач

Оскільки штучний інтелект (ШІ) постійно розвивається, його стали застосовувати в більшості сфер життєдіяльності людини. В медицині використання ШІ почалося порівняно нещодавно, і основним напрямком дослідження було діагностування хвороб по зображенням, класифікуючи їх на спеціальні охарактеризовані категорії.

Існуючих рішень вже багато, наприклад, у 2021 році згорткову НМ на основі EfficientNet-B4 (рисунок 1.6) від Google навчили розпізнавати дермоскопічні зображення псоріазу та інших папулосквамозних захворювань, при цьому порівнюючи вихідні дані з думками 230 кваліфікованих лікарів-дерматологів [5]. В дану архітектуру завантажили 7000 зображень ураженої шкіри з понад 1000 пацієнтів міста Пекін. Загалом, істотної різниці показники не видали, думки дерматологів і штучного інтелекту досягли узгодженості зі

стандартом і не перевищували 0,05%. Таким чином, можна однозначно діагностувати дерматологічну хворобу на прикладі псоріазу і з допомогою НМ.

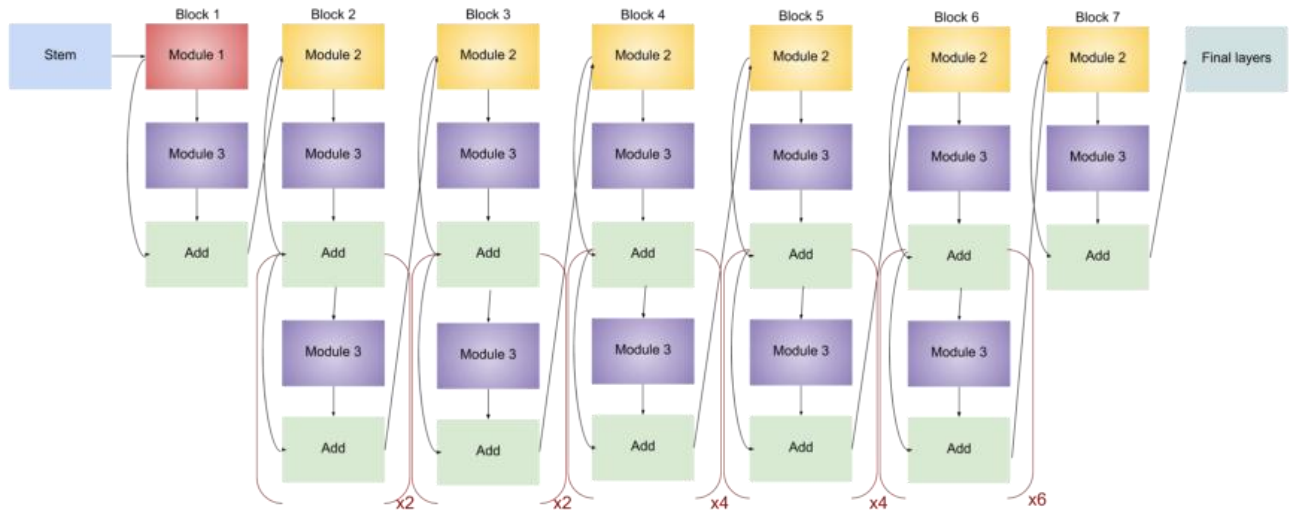


Рисунок 1.6 – Архітектура EfficientNet-B49 [4]

Найбільш проривом в дослідженні дерматологічних хвороб з допомогою НМ стала розроблена Google програма DermAssist [6]. Вона дозволяє за лічені хвилини дізнатися про проблеми з вашою шкірою. Ця програма досі вдосконалюється, проходячи тестування на ринку продуктів аби працювати з різними типами і кольорами шкіри. Щоб почати користуватися DermAssist'ом потрібно заповнити форму, надану вам Google, в якій вказуєте ваші контактні дані, тип мобільного пристрою і модель, що ви використовуєте, вік, країну, і тип шкіри по шкалі Фіцпатріка [7]. DermAssist, навчений на мільйонах зображень, може розрізнити більшість відомих дерматологам хвороб, і надасть вам інформацію саме по вашій проблемі у вигляді відповідності одній або декільком можливим хворобам. Вигляд застосунку зображено на рисунку 1.7

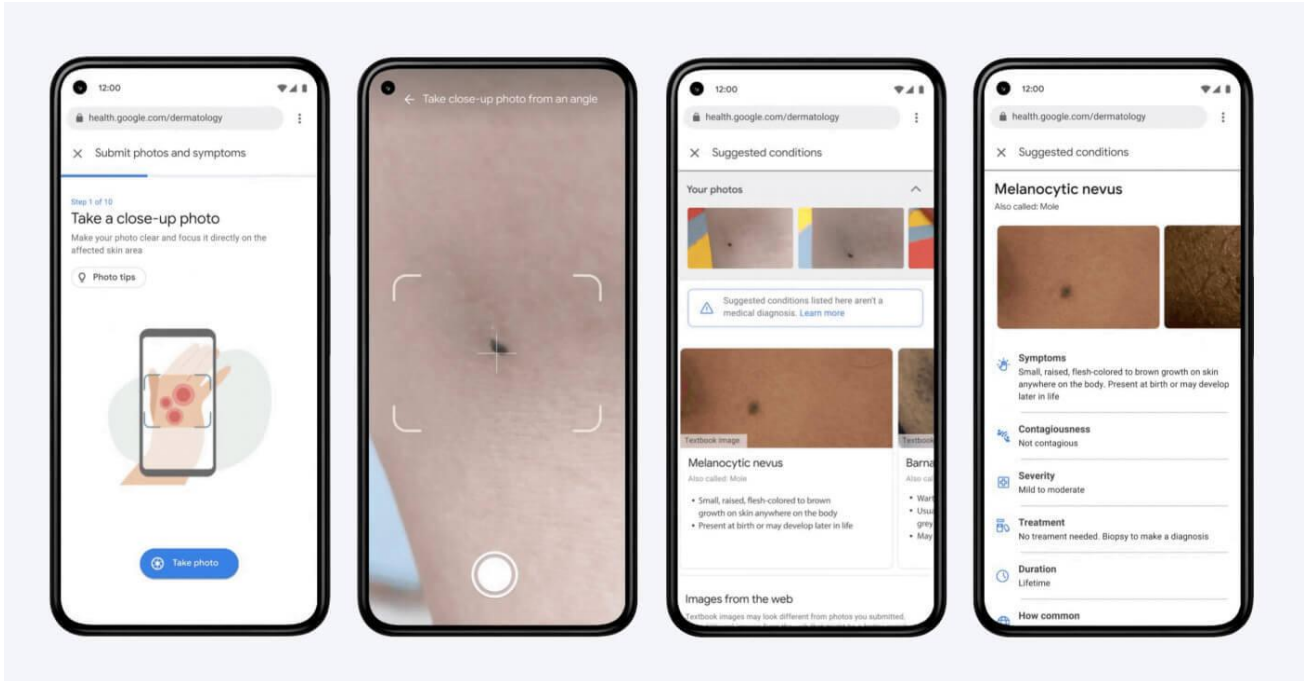


Рисунок 1.7 – DermAssist [6]

Схожим на DermAssist є застосунок Miiskin [8], який спочатку розроблявся, як платформа для досягнення певних цілей в дерматологічних дослідженнях в Великобританії та Європі. Він став популярним і вийшов на міжнародний ринок. З допомогою нього науковці можуть проводити дослідження раку шкіри, інші споживачі аналізувати тілесні рани та захворювання, включаючи такі як псоріаз та екзема. Пацієнти використовують Miiskin, щоб доглядати за своєю шкірою, і консультуватися з лікарями-дерматологами, надаючи їм потрібну інформацію онлайн. Тож, основною функцією Miiskin є моніторинг стану шкіри та родимок, яка допомагає пацієнту перевірити, ідентифікувати будь-які зміни зовнішнього вигляду шкіри. Світлини екрану програми зображено на рисунку 1.8

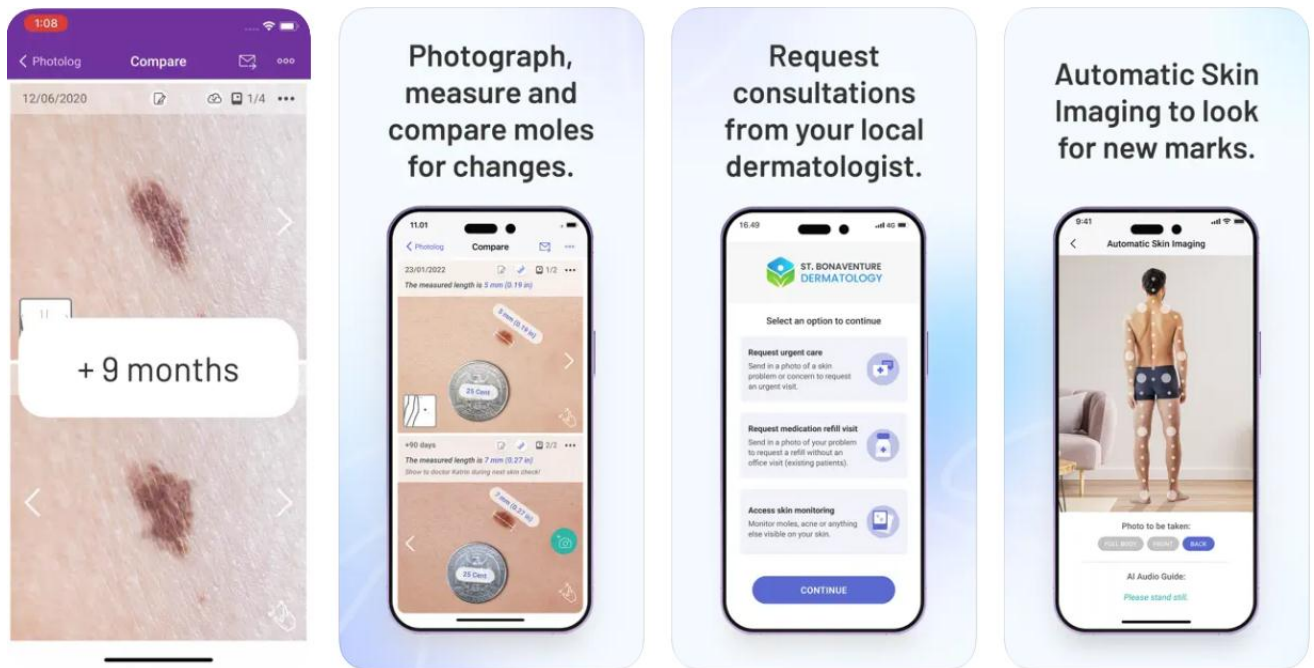


Рисунок 1.8 – Miiskin [8]

Miiskin має декілька важливих особливостей, щоб результати її роботи були ефективними:

- Користувач робить зображення, а інтегрована автоматична обробка робить їх високої якості для найбільш точної ідентифікації.
- Мапування шкіри – функція, які відслідковує різні утворення, родинки і плями.
- Щоб аналізувати зміни розмірів даних утворень, програма оснащується інструментами доповненої реальності.

Серед вищеперечислених DermAssist та Miiskin, з'являється все більше дерматологічних застосунків, які теж покращуються і можуть конкурувати за 1 місце:

- SkinVision.
- Derma AI.
- First Derm.
- MDacne.
- DermEngine.

– AI Skin Diagnosis.

Отже, всі наведені застосунки, зазвичай, створювалися під мобільні платформи задля зручності користування, але є й такі, де потрібно заповнити форму і просто завантажити зображення з будь-якого пристрою. Програми діагностування дерматологічних захворювань керуються єдиним принципом, в якому їм надаються зображення уражених ділянок шкіри пацієнтів і по них нейромережа проводить діагностику. Ефективність лише залежить від вибраного методу класифікації з використанням нейромереж.

Оскільки, існує вже дуже багато програмних рішень діагностики дерматологічних хвороб, а тому цій проблемі приділяють багато уваги і створення ідеальної ІС діагностики має велике значення в сфері медицини.

1.4 Мета, задачі та вимоги до реалізації програмної системи

Метою роботи є розробка нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями та ІС, яка використовує розроблений метод і дозволяє за зображеннями ураженого фрагменту шкіри людини визначати наявні дерматологічні хвороби за допомогою нейронної мережі. Для досягнення мети потрібно вирішити наступні задачі:

– розробити нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями;

– розробити архітектуру нейронної мережі для діагностування дерматологічних хвороб в людини за їх зображеннями;

– спроектувати структуру ІС, яка використовує розроблений метод і дозволяє за зображеннями ураженого фрагменту шкіри людини визначати наявні дерматологічні хвороби за допомогою нейронної мережі;

– виконати вибір засобів розробки ІС нейромережевого діагностування дерматологічних хвороб за їх зображеннями;

- виконати програмну реалізацію ІС нейромережевого діагностування дерматологічних хвороб;
- провести тестування створеної ІС.

Розділ 2 Нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями

2.1 Схема та кроки нейромережевого методу діагностування дерматологічних хвороб

Нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями дозволяє одержувати результат у вигляді діагностованої хвороби за фотозображеннями уражених ділянок шкіри. Суть нейромережевого методу діагностування дерматологічних хвороб полягає в тому, аби класифікувати зображення, надіслані пацієнтом, в попередньо організовані категорії захворювань, які найкращим чином характеризують його. А для цього потрібно створити і налаштувати нейромережу розпізнавати вхідні дані. З цим завданням справляються згорткові нейронні мережі. Хоча вони з'явилися ще наприкінці 20-го століття, тільки починаючи з 2000-х років вони перетворилися на безцінний інструмент в руках дослідників, ставши незамінним компонентом в клінічній діагностиці в таких сферах, як радіологія, гістологія, офтальмологія і дерматологія. Особливо в дерматології, архітектури глибокого навчання для класифікації зображень показали видатні результати. Створення ІС, яка зможе розпізнавати по зображеннях різні дерматологічні хвороби складається з кількох основоположних етапів, які присутні в проектуванні майже кожної подібної системи.

Алгоритми НМ зможуть правильно обробляти інформацію, якщо їм надати потрібні дані для навчання. Ці дані можуть бути попередньо аналізовані, підготовлені, і міститися в датасетах, або ж можна підготувати власні, але це займе більше часу. Датасети для медичної сфери містять зображення видимих хвороб, аномалій чи певні документи, які характеризують ті чи інші патологічні зміни в організмі людини, в нашому випадку, це зображення захворювань на шкірі. Підготовка такого датасету включає в себе очищення його від

непотрібних зображень, або наповнення новими, приведення їх до однакового розміру, формату і типу даних для збереження пам'яті.

Після цього датасет розділяють на набори для навчання і тестування. Нейромережева модель використовує тренувальний набір для навчання за завчасно наданими характеристиками зображень з вказаними на них хворобами. Ця інформація потрібна для налаштування параметрів моделі, щоб здійснювати точні передбачення на нових зображеннях. Тестувальний набір відповідає за оцінку якості роботи мережі з діагностики захворювань. Дані зображення містять лише вигляд захворювання без їхнього визначення. Модель використовує їх щоб порівняти правильні відповіді і відповіді мережі на немарковані дані, це дає змогу оцінити якість процесу навчання на попередньому наборі. Також є варіант при якому датасет розподіляється на 3 набори (рисунок 2.1)

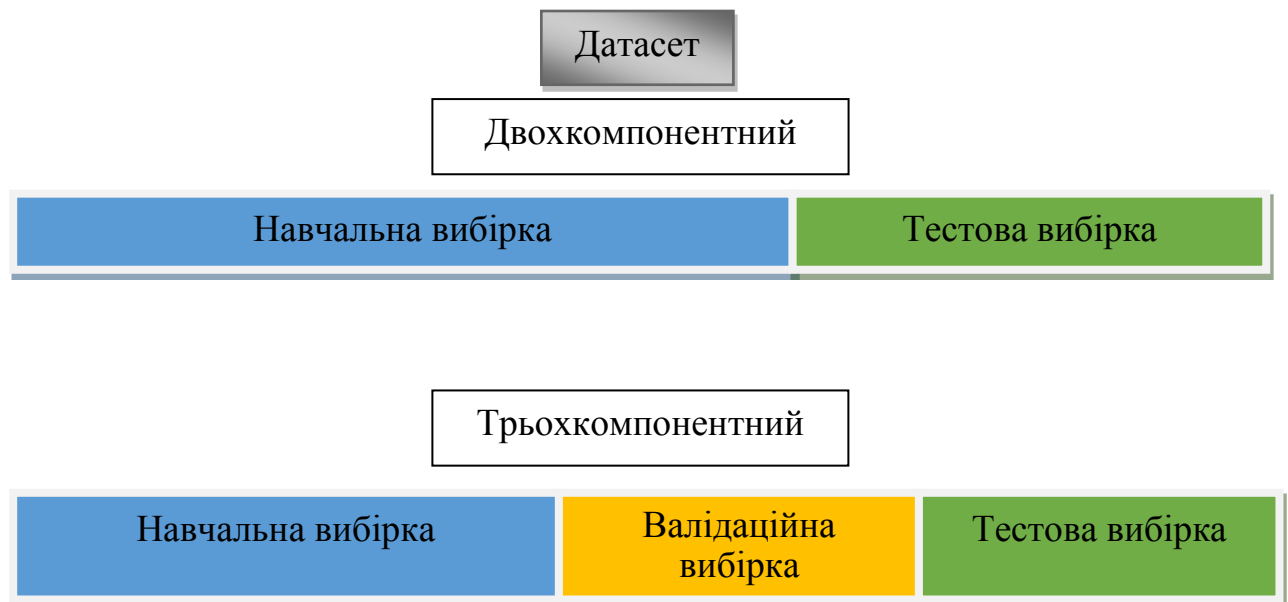


Рисунок 2.1 – Приклад розділення датасету на вибірки

Який об'єм даних з датасету буде виділений для наборів залежить від таких факторів, як поставленого завдання класифікації та доступного обсягу даних.

Зазвичай, в завданнях комп'ютерного зору та взагалі в МН, виділяють приблизно 70-80% даних для навчального набору, 10-15% для валідаційного набору і 10-15% для тестового набору, що залишилися. Така практика показала себе виправданою, але можна використовувати і інші значення. Потрібно лише пам'ятати, що розподілення буде коректним за умови, що вибрані випадкові частини з датасету і вони містять дані, які характеризують датасет в цілому. При такому розпорядку, НМ буде навчатиметься на різноманітних даних і поглиблювати знання про нові. Часто проблемами виступають особливості даних в датасеті – незбалансовані класи, тому потрібно враховувати і таку можливість, щоб підібрати відповідний розподіл.

Маючи готовий датасет, вибирається архітектура НМ відповідно до складності створюваної системи і об'єму датасету. Цей процес описується на схемі (рисунок 2.2)

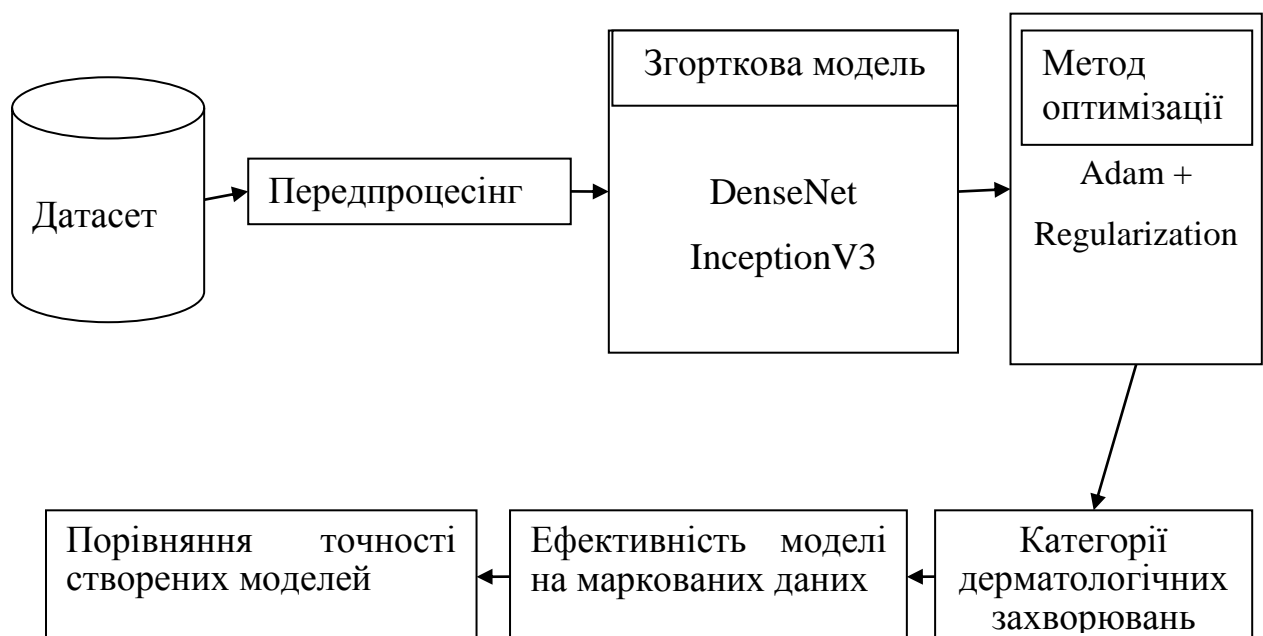


Рисунок 2.2 – Порядок вибору архітектури згорткової нейронної мережі

На схемі (рисунок 2.2) показано, що як для вибору підходящої архітектури НМ, потрібно проаналізувати датасет і обробити зображення в ньому. Для цього зображення піддаються різним методам обробки, як змінення розміру, контрастності.

Наступним кроком обрано архітектуру DenseNet, InceptionV3, кожна з яких має свої особливості побудови і кількість налаштовуваних параметрів. Навчившись на тренувальному наборі, модель зможе відбирати з зображень потрібні ознаки, як форми і текстури, і побудує оптимальні ваги нейронів для класифікації зображень.

Також, в якості підбору оптимальних значень параметрів використано оптимізатори процесу навчання, як adam [9] і regularization. Adam є одним з найпопулярніших, і є нащадком AdaGrad и RMSprop, і може адаптуватися до швидкості навчання різних параметрів.

Охарактеризовано категорії дерматологічних захворювань, які модель буде класифікувати. Це виконується за рахунок створення виходів нейронної моделі, кожен з яких відповідає за окремий клас.

Оцінка результативності навченої моделі включає перевірку її ефективності в розпізнаванні дерматологічних хвороб, керуючись такими метриками: чутливість, точність та F-міра [10].

Якщо потрібно удосконалити модель задля покращення точності, то можна змінити її параметри або сам процес підготовки даних і навчання.

Після реалізованих попередніх етапів, розроблено користувацький інтерфейс для завантаження фотографій і використання навченої моделі для отримання результатів діагностики в реальному часі.

Схема і кроки нейромережевого методу діагностування дерматологічних хвороб виглядає наступним чином (рисунок 2.3)

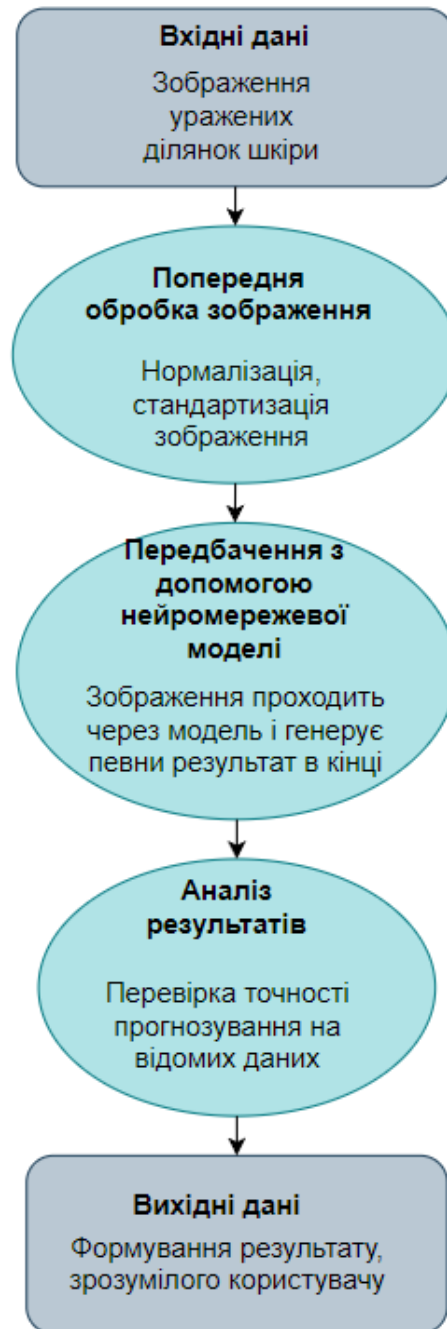


Рисунок 2.3 – Кроки нейромережевого методу діагностування дерматологічних хвороб

Вхідними даними системи нейромережевого діагностування дерматологічних захворювань є зображення уражених ділянок шкіри пацієнта. Ці зображення можуть завантажуватися з різних джерел.

При попередній обробці зображення піддаються розмірній і контрастній нормалізації, видаляють шум, збільшують чіткість, вказують глибину кольору.

Також використовуються такі техніки як вирівнювання, видалення фону або підвищення контрастності, при цьому виділяються найбільш чіткі ознаки ураження на шкірі. Методи комп'ютерного зору будуть аналізувати саме ці ознаки, які характеризуються лініями, формами, текстурою, кольоровою гамою, структурою тощо.

Далі починається ключовий процес в роботі системи – діагностування з допомогою навченої НМ. Модель оброблює зображення та генерує на виході результат, який означає одну з категорій захворювань. Отримані прогнози піддаються аналізу для отримання остаточного діагнозу. В цьому етапі можуть бути використані різні методи, такі як порівняння з класифікаційними системами або порівняння з аналогічними зображеннями в датасеті. Наприклад, система може порівнювати ознаки або характеристики на зображенні зі збереженими шаблонами або паттернами хвороб, що були вивчені в процесі навчання моделі.

В зрозумілій формі користувач отримує результат через інтерфейс. Важливо зазначити, що ефективність системи нейромережевого діагностування дерматологічних захворювань залежить від багатьох факторів, а саме, якості вхідних даних, правильного навчання моделі та здатності моделі адаптуватися до нових варіацій і різноманітності хвороб. Крім того, система повинна проходити періодичну перевірку, оновлення та підтримку, щоб забезпечити найвищу якість діагностики.

Отже, проаналізовані схеми та кроки нейромережевого методу діагностування дерматологічних хвороб. Основний функціонал системи полягає у використанні зображень хвороб для діагностування та розробці нейромережевої моделі, що може відрізнити різні типи хвороб на основі їх ознак, виділених на зображенні. Для цього потрібно підготувати датасет з зображеннями шкірних хвороб, розділити їх на навчальну і тестову вибірки за певним відсотковим співвідношенням, використати обрані архітектури згорткових НМ і провести тестування створеної системи.

Створена відповідна методу ІС підійде у використанні в сфері медицини задля діагностування дерматологічних хвороб на основі зображень. Її можна було б використовувати як автоматичний сервіс онлайн-діагностики або в лікарнях на допомогу лікарям.

Таким чином, було розроблено нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями, який дозволяє одержувати результат у вигляді діагностованої хвороби за фотозображеннями уражених ділянок шкіри.

2.2 Розробка архітектури нейронної мережі для діагностування дерматологічних хвороб

Для створення архітектури нейронної мережі для діагностування дерматологічних хвороб потрібно реалізувати наступні етапи (рисунок 2.4).

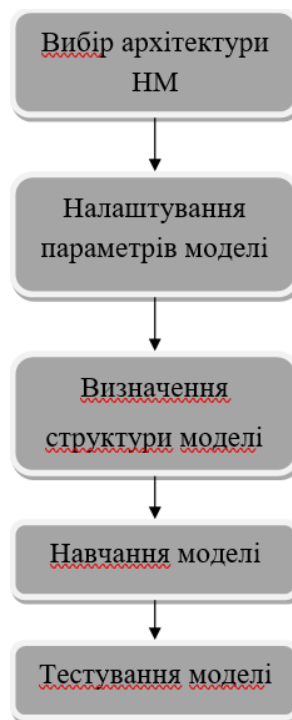


Рисунок 2.4 – Порядок створення нейромережевої моделі

Для системи нейромережевого діагностування дерматологічних хвороб за зображеннями використовується архітектура згорткових нейронних мереж, яка навчена на великому датасеті зображень, адже область застосування такої моделі специфічна, а тому потребує точності діагностування. Вибір архітектури описано в схемі (рисунок 2.2).

Визначення структури моделі є непростим завданням і залежить від об'єму даних в датасеті, розділених вибірок, і складності зображень. Оскільки, в конкретному випадку, зображення дерматологічних захворювань, а тому може знадобитися багат шарова архітектура з глибокими фільтрами, тому згорткові НМ чудово розроблені для роботи з просторовою інформацією.

Навчання і тестування моделі відбувається поступово: навчання являє собою подачу в модель зображень і оновлення ваг мережі, щоб вона могла здійснювати передбачення, а тестування – перевірку, наскільки ефективно це навчання відбулося.

Одною з найефективніших архітектур є DenseNet [11] (рисунок 2.5) – створена на базі ResNet і чудово пристосована до завдань класифікації. DenseNet, налаштовуючи зв'язки між шарами, краще виділяє ознаки з зображень, що є необхідним аспектом через складність зображень дерматологічних хвороб.

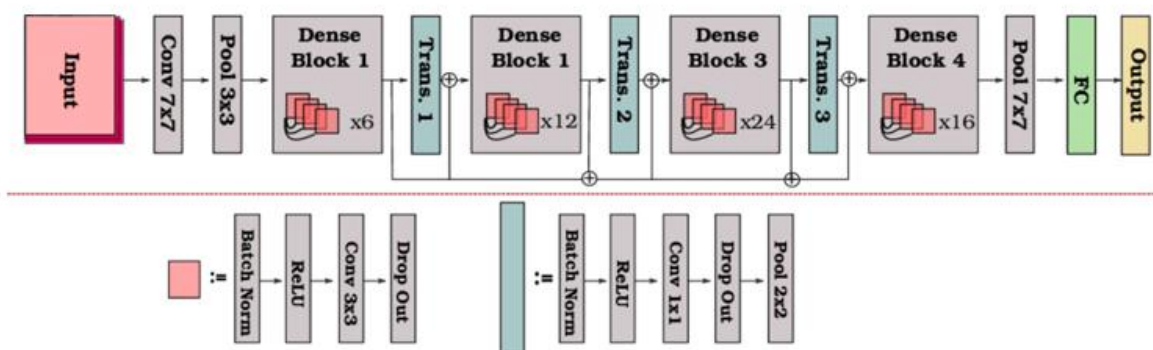


Рисунок 2.5 – Архітектура DenseNet [11]

Головною відмінністю від інших архітектур, таких як VGG або ResNet, є те, що DenseNet використовує принцип «пов'язаних» згорткових шарів. А значить це те, що кожен шар своїм входом має всі попередні шари. Завдяки цьому забезпечується інтенсивний обмін інформацією між шарами та поліпшується градієнтний потік в нейромережі. DenseNet є наступним кроком на шляху до збільшення глибини згорткових НМ.

DenseNet використовує блоки, які називаються Dense-блоками. Вони складаються з кількох згорткових шарів, які послідовно з'єднані між собою, і як було згадано, взаємодіють з попередніми шарами, отримуючи потрібну інформацію з них. Послідовно розташовані згорткові шари з однаковою кількістю фільтрів формують одиничний блок. Схематичний вигляд такого блоку зображено на рисунку 2.6

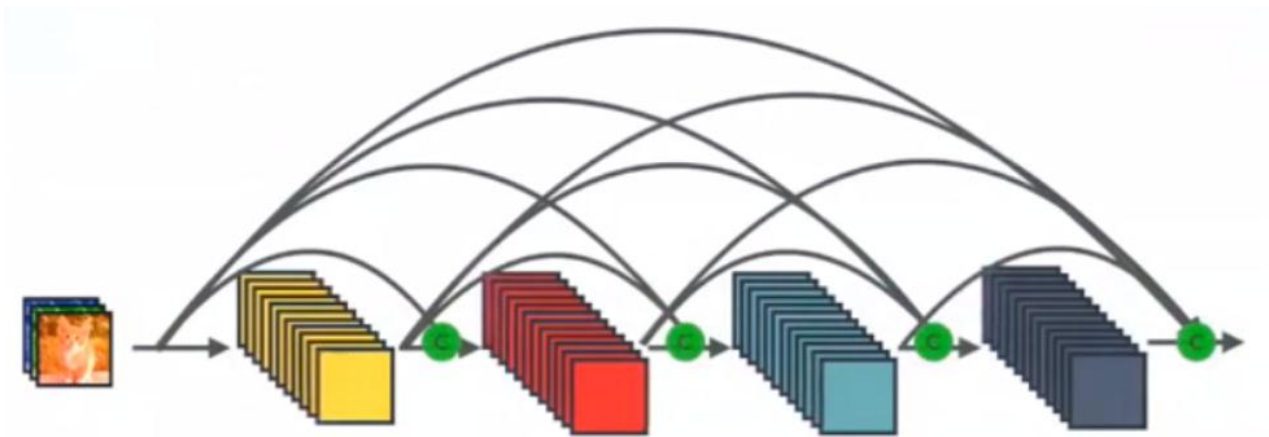


Рисунок 2.6– Dense-блок [12]

Обчислювальна складність моделі корегується за рахунок перехідних шарів, які зменшують кількість фільтрів і розмір зображення. Також між блоками зазвичай вставляють шари пулінгу або пакетної нормалізації для покращення продуктивності моделі. Останній Dense-блок архітектури представляє собою функцію пулінгу, яка сумарно збирає дані з усіх фільтрів, що запобігає перенавчанню. Зв'язуються блоки один між одним шляхом

конкатенації вихідних функцій. В якості функції активації використовують ReLU разом з пакетною нормалізацією задля стабільності навчання, а як для функції втрат використовують категоріальну перехресну ентропію. Для оптимізації підходять стохастичний градієнтний спуск або Adam, з відповідними гіперпараметрами.

Через описані особливості DenseNet є компактним нащадком ResNet, який у всьому кращий за нього в багато раз, в неї збільшується обчислювальна здатність і використання пам'яті. Завдяки його ефективності та здатності до використання невеликої кількості параметрів він може досягати високої точності класифікації та виявлення об'єктів, особливо при обробці медичних зображень, включаючи дерматологічні захворювання.

Розроблена архітектура нейронної мережі для діагностування дерматологічних хвороб з використанням Dense-блоків має вигляд, зображений на рисунку 2.7.

Дана архітектура містить в собі такі елементи:

- Вхідний шар: форма вхідних даних (зображення 100x100 пікселів, з трьома каналами кольору).
- 1 згортковий шар з 96 фільтрів, ядром 11x11, з функцією активації ReLU, після якого використовується пакетна нормалізація і максимальний пулінг 2x2.
- 2 згортковий шар з 256 фільтрів, ядром 5x5, з функцією активації ReLU.
- 3 згортковий шар з 384 фільтрів, ядром 3x3, з функцією активації ReLU.
- 4 згортковий шар з 384 фільтрів, ядром 3x3, з функцією активації ReLU.
- 5 згортковий шар з 256 фільтрів, ядром 3x3, з функцією активації ReLU, після якого використовується пакетна нормалізація і максимальний пулінг 2x2.

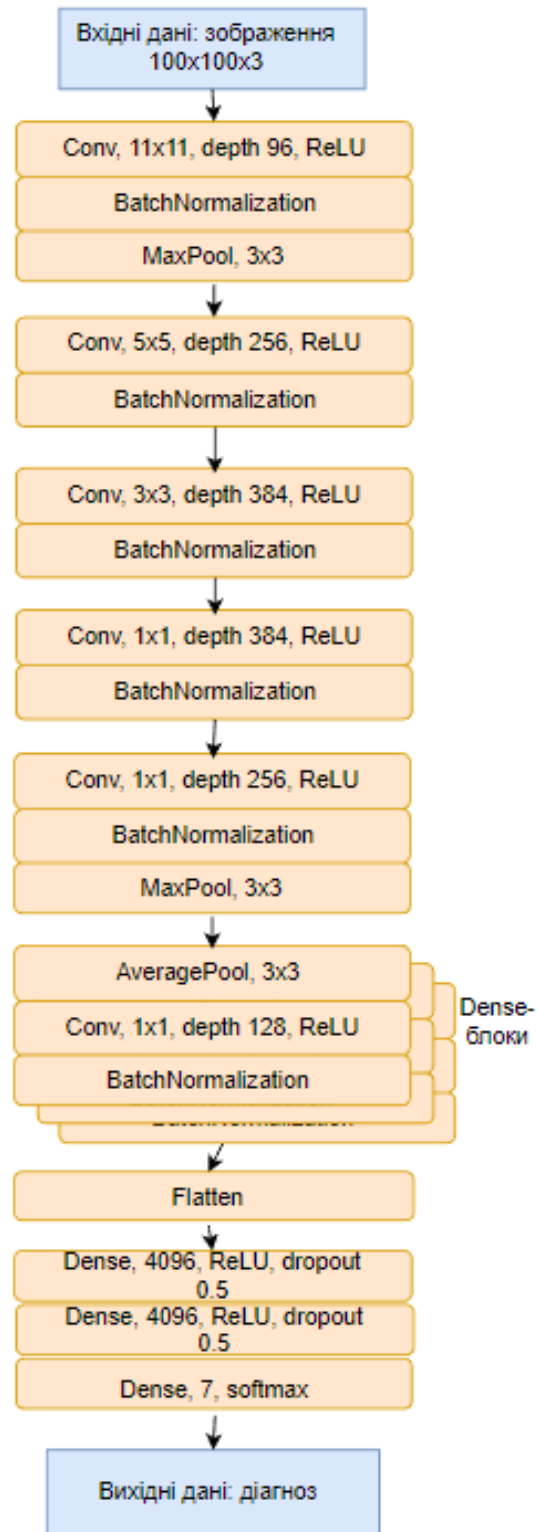


Рисунок 2.7 – Структура нейромережевої моделі

Також дана архітектура НМ має 3 DenseNet-блоки для поліпшення виконання мережі, кожен з яких складається з середнього пулінгу 3x3,

згорткового шару зі 128 фільтрів, ядром 1x1 і пакетною нормалізацією. З'єднуються блоки функцією конкатенації. Також наступними складовими архітектури є:

- Плоский шар для перетворення виходу з останнього DenseNet блоку в одномірний вектор.
- 2 Повнозв'язні шари з 4096 нейронів, функцією активації ReLU, та ймовірністю вимикання 0.5.
- Останній повнозв'язний шар з 7 нейронів, відповідно до класів захворювань, та функцією активації softmax.
- Оптимізатором є Adam.
- Функцією втрат є категоріальна перехресна ентропія.

Іншою чудовою архітектурою є InceptionV3 [13] (рисунок 2.8), яка є ідейним наслідником Inception і InceptionV2.

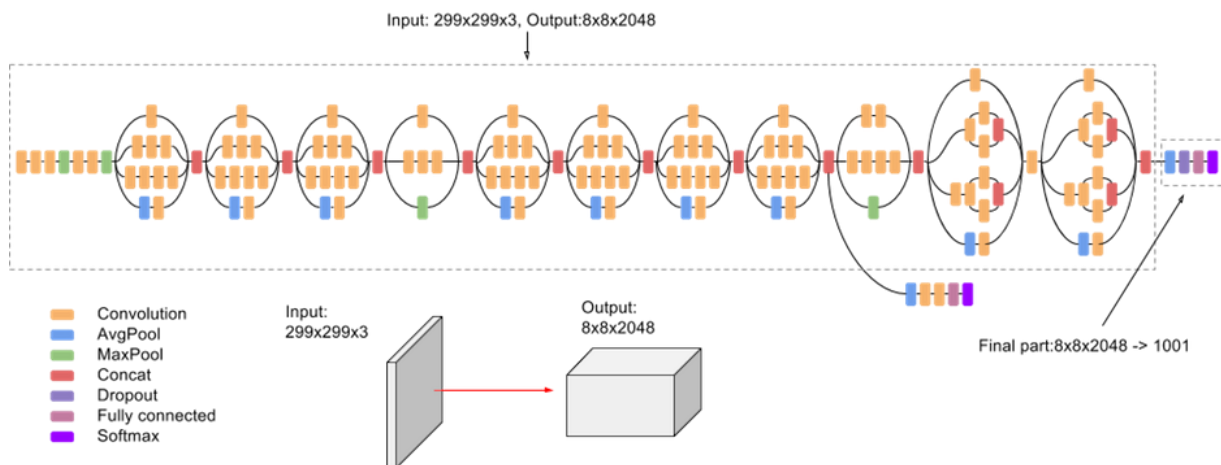


Рисунок 2.8 – InceptionV3 [13]

Зарекомендувала себе на змаганнях ILSVRC [14] 2015 року, справляючись зі складними зображеннями, виявляючи об'єкти і класифікуючи їх. Новий модуль даної архітектури, названий Inception-module (рисунок 2.9) [13], здатен паралельно виконувати згортки.

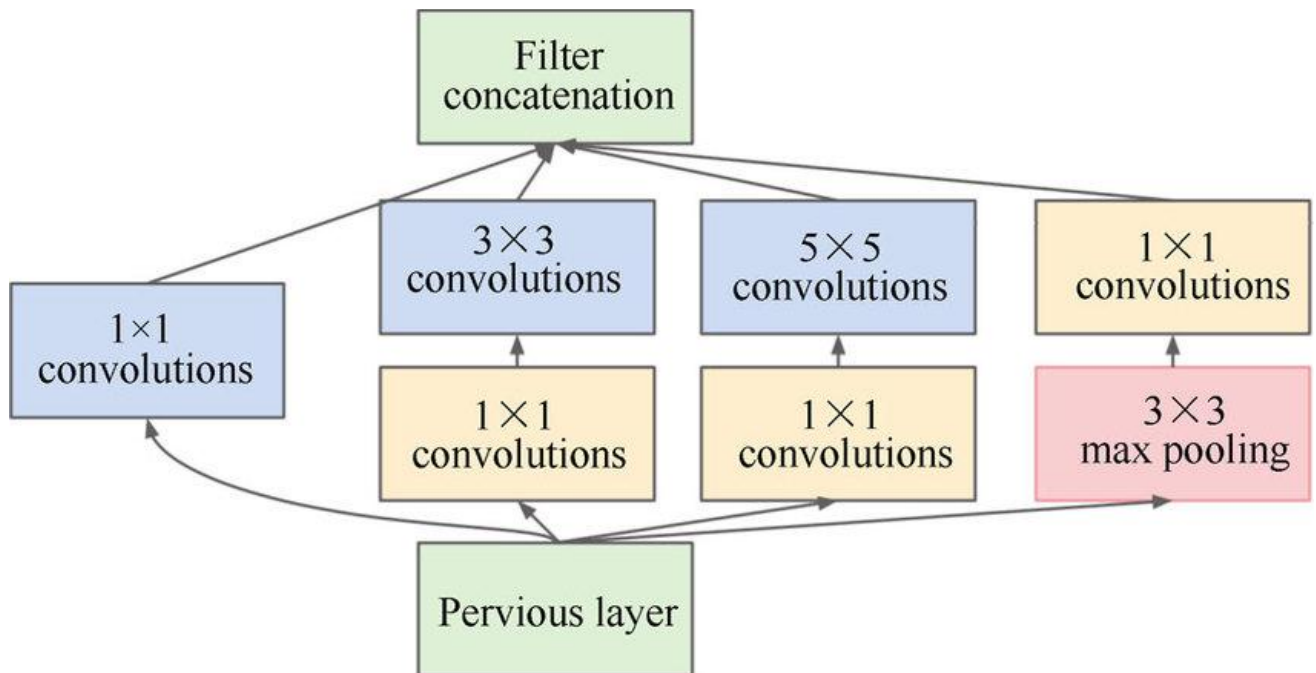


Рисунок 2.9 – Приклад Inception-модуля [13]

Inception-модуль є головною особливістю завдяки паралельному виконанні функцій згортки і пулінгу, що дозволяє захоплювати більше інформації з зображення. Він зазвичай складається з таких компонентів:

- 1x1 згортка на вхідних даних, яка вибирає з зображення взаємозв'язки між каналами та зменшує розмірність вхідних даних, покращуючи обчислювальну ефективність.

- 3x3 згортка, яка захоплює просторові зразки та надає більше широке рецептивне поле, на відміну від 1x1 згортки. Може використовуватися в зв'язці з кількома 3x3 згортками.

- 5x5 згортка, що захоплює найбільш складні особливості.

- Максимальний пулінг.

Створена архітектура НМ діагностування дерматологічних хвороб на основі InceptionV3 виглядає так (рисунок 2.10):

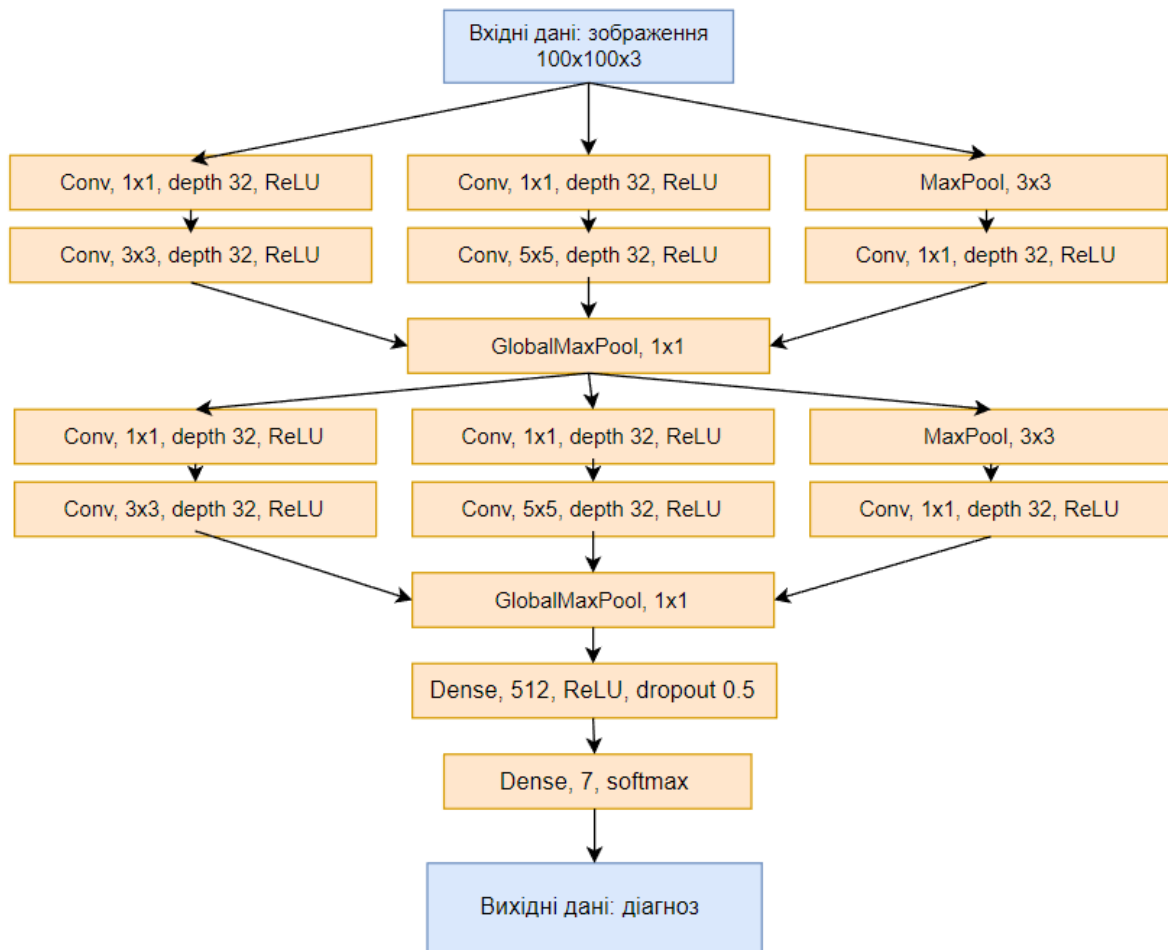


Рисунок 2.10 – Структура нейромережевої моделі

Дана архітектура містить наступні компоненти:

- Вхідний шар: форма вхідних даних (зображення 100x100 пікселів, з трьома каналами кольору).
- 2 Інсершн-модуля, які складаються з гілок згортки 1x1, 3x3, 5x5, і 3x3 пулінгу. Кожен згортковий шар містить 32 фільтри і використовує функцію активації ReLU. В кінці виходи гілок з допомогою конкатенації з'єднуються.
- Глобальний максимальний пулінг 1x1.
- 1 повнозв'язний шар з 512 нейронів і ймовірністю вимикання 0.5.
- 1 повнозв'язний шар з 7 нейронів, що відповідають за класи захворювань.

Отже, для системи нейромережевого діагностування дерматологічних хвороб було розроблено архітектури з використанням двох найсучасніших

згорткових НМ – InceptionV3 і DenseNet. Визначено оптимальні параметри для налаштування НМ і детально описані для більш чіткого ознайомлення. Тож, діагностику дерматологічних хвороб можна проводити використовуючи розроблені архітектури згорткових нейронних мереж.

2.3 Підготовка робочих даних для системи

Для навчання НМ потрібно набір даних, яким є HAM1000. Він дослівно перекладається як «людина проти машини з 10000 навчальними зображеннями». Складається з медичних зображень шкіри, призначених для завдань класифікації дерматологічних захворювань. Найбільшу популярність отримав у 2018 році, коли його повсюдно використовували для найбільшого в світі завдання з аналізу зображень шкіри [15], організатором якого була Міжнародна кооперація з обробки зображень шкіри (ISIC) [16], Вона ж і була засновником цього набору. До того ж на офіційному сайті представлено публічне сховище з найбільшою кількістю дермоскопічних зображень шкіри.

Є другим датасетом по кількості зображень шкіри з різними типами дерматологічних хвороб, на першому місці DermNet [17] з 23000 зображень, але відрізняється від нього більш доцільною організацією категорій захворювань, і спеціалізується на більш вузькій сфері захворювань, класифікуючи злоякісні та доброякісні ураження.

Зображення зібрані при звичайній дерматоскопії, але оскільки не всі захворювання викликають пігментацію, тобто забарвлення шкіри людини пігментом, то й у датасеті залишилися лише ті зображення на яких чітко видно уражену область з пігментованим ушкодженням. Зображення наведені на рисунку 2.11

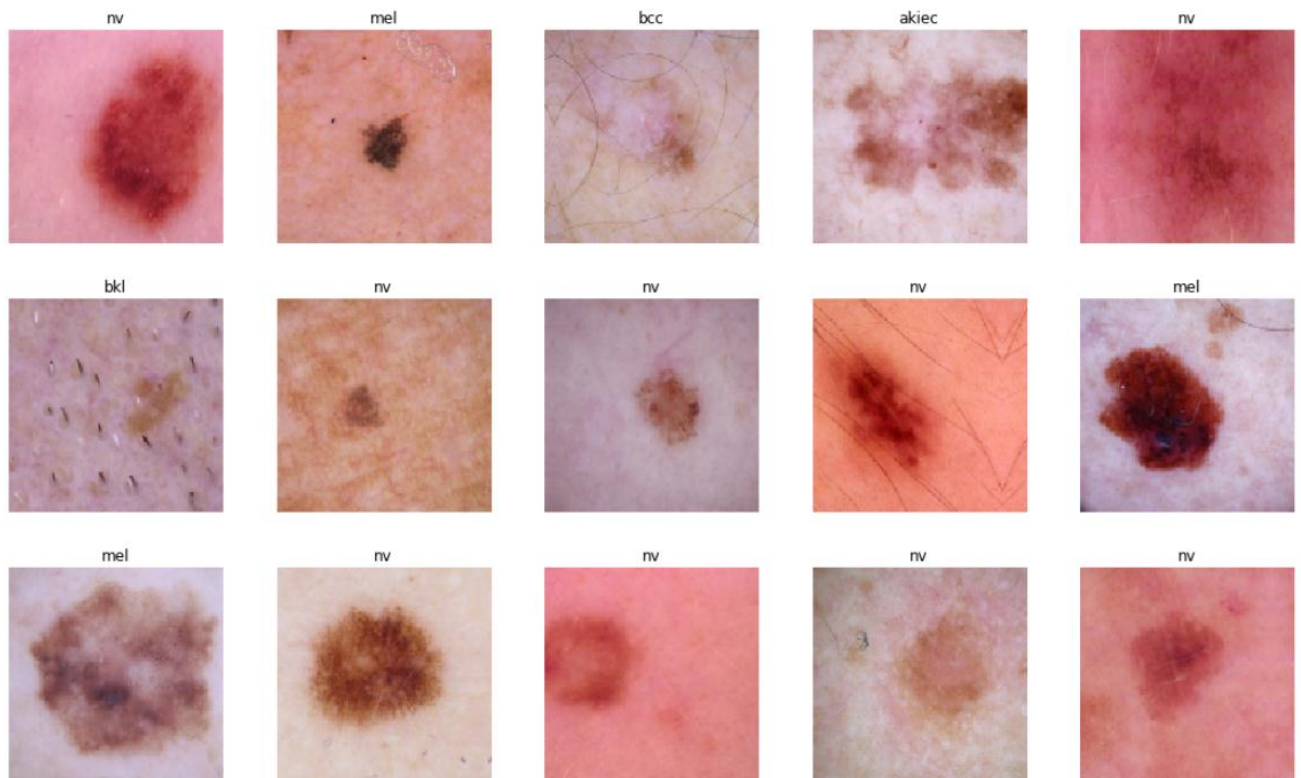


Рисунок 2.11 – Випадкові зображення, представлені в датасеті

Завантажити датасет можна, користуючись популярною платформою для розробників Kaggle [18], використовуючи їхній API, або зберігши на локальний пристрій. Завантажений архів складається з наступних файлів:

- `hmnist_28_28_RGB.csv`, який містить посилання на зображення в форматі PNG розміру 28x28 і дані про шлях до файлу, мітку класу, вік та стать пацієнта.

- `hmnist_28_28_L.csv` містить вже посилання ті самі зображення, але вже чорно-білі.

- `hmnist_8_8_RGB.csv` – посилання на зображення в форматі PNG розміру 8x8

- `hmnist_8_8_L.csv`: – як і `hmnist_8_8_RGB.csv`, але зображення чорно-білі

– HAM10000_metadata.csv – в файлі – детальна метадані про кожне зображення в наборі даних, включаючи діагноз, тип, вік, локалізації на тілі, і деякі характеристики.

– ham10000_images_part_1.zip и ham10000_images_part_2.zip – фактичні зображення уражених ділянок шкіри. В архіві – 5000 файлів.

7 різних класів представлено в HAM10000 (перша назва – скорочена, в якості мітки):

– akiec – Actinic Keratoses (актинічний кератоз) и intraepithelial carcinoma / Bowen's disease (внутрішньоепітеліальний рак / хвороба Боуэна). Груба шкіра, яка є наслідком тривалого впливу ультрафіолетового випромінювання. Викликає рак.

– bcc – Basal cell carcinoma (базаліома). Найбільш поширений тип раку шкіри, який уражає шар епідермісу.

– bkl – Benign Keratosis-like Lesions (бенджинові кератозоподібні утворення), кератози, які утворюються з ороговілих клітин шкіри.

– df – Dermatofibroma (дерматофіброма). Ураження, причиною якої є травми або укуси на шкірі. Має вигляд ущільненої і пігментованої області.

– melanoma – Melanoma (меланома) є злоякісною опухоллю і симптомом небезпечної форми раку шкіри, яку потребує вчасного лікування, утворюється з меланоцитів.

– nv – Melanocytic nevi (меланоцитові родинки), ті самі родинки або родимі плями, що утворюються з меланоцитів.

– vasc – Vascular Lesions (судинні утворення), гемангіоми або винні плями, утворюються з кров'яних судин.

Результати нейронних моделей, навчених на ньому, змогли перевершити оцінки міжнародних спеціалістів. Точність навчених моделей зображена на рисунку 2.12

Experiment	Method	ACC
Experiment 1 HAM10000 (80% training, 20% testing)	PNASNet [77]	76.00%
	ResNet-50 + gcForest [78]	80.04%
	VGG-16 + GoogLeNet	81.50%
	Ensemble [79]	
	Densenet-121 with SVM [80]	82.70%
	DenseNet-169 [80]	81.35%
	Bayesian DenseNet-169 [81]	83.59%
	Shifted MobileNetV2 [23]	81.90%
	Shifted GoogLeNet [23]	80.50%
Shifted 2-Nets [23]	83.20%	
	The proposed method	84.80%
Experiment 2 HAM 10000 (90% training, 10% testing)	HARTS [82]	77.00%
	FTNN [83]	79.00%
	CNN [84]	80.00%
	VGG19 [85]	81.00%
	MobileNet V1 [68]	82.00%
	MobileNet V2 [86]	84.00%
	MobileNet V2-LSTM [29]	85.34%
	The proposed method	86.33%
Experiment 3 ISIC 2019 (90% training, 10% testing)	VGG19 [85]	80.17%
	ResNet-152 [64]	84.15%
	Efficient-B0 [87]	81.75%
	Efficient-B7 [87]	84.87%
	The proposed method	82.56%

Рисунок 2.12 – Порівняння точності різних моделей на наборі даних HAM10000 [19]

Основним обмеженням в нейромережевому діагностуванні дерматологічних хвороб з допомогою HAM10000 є те, що єдиних дермоскопічних даних недостатньо для точності роботи системи. Медичні спеціалісти задля отримання коректного результату використовують крім зображень ще й інші виміри, як розмір ураження, огляд всього тіла тощо. А деякі деталі неможливо виявити стандартними техніками.

Отже, HAM10000 є найбільш вдало вибраним ресурсом для навчання нейромереж в завданнях класифікації шкірних захворювань. HAM10000 спеціалізується на більш вузькій сфері хвороб, має переваги над DermNet, надає можливість визначати злякисні, доброякісні ураження шкіри. Даний датасет

показав чудові показники при навчанні різних моделей. В цьому підрозділі також було детальніше описані дані з датасету і типи захворювань. Об'єму важливої інформації HAM10000 вистачає для точного класифікування означених категорій.

2.4 Проєктування структури інформаційної системи нейромережевого діагностування дерматологічних хвороб

Система, здатна діагностувати дерматологічні хвороби за їх зображеннями, матиме багато переваг, таких як:

- Сучасні нейромережеві моделі мають покращену точність діагностики, ніж при реальних клінічних умовах, що дозволить уникнути помилкових діагнозів.

- Штучний інтелект може проводити діагностику набагато швидше, це зекономить час лікарям на створення тактики боротьби з хворобою.

- Процес діагностики в лікарняному кабінеті може принести пацієнту біль, дискомфорт, чого не зустрічається при онлайн сеансі.

- Така система буде доступна в будь-якому місці і будь-коли, кожен користувач може дізнатися свій діагноз.

Щоб цього досягти, потрібно організувати структуру системи, створити модулі, які правильно взаємодітимуть між собою задля зручного користування даним застосунком. В даній ІС наявні такі модулі (рисунки 2.13):

- користувацький інтерфейс;
- модуль обробки зображень;
- модуль роботи з датасетом
- нейромережевий модуль;
- модуль обробки результату.

Опис функцій кожного модуля наведено на рисунку 2.14.

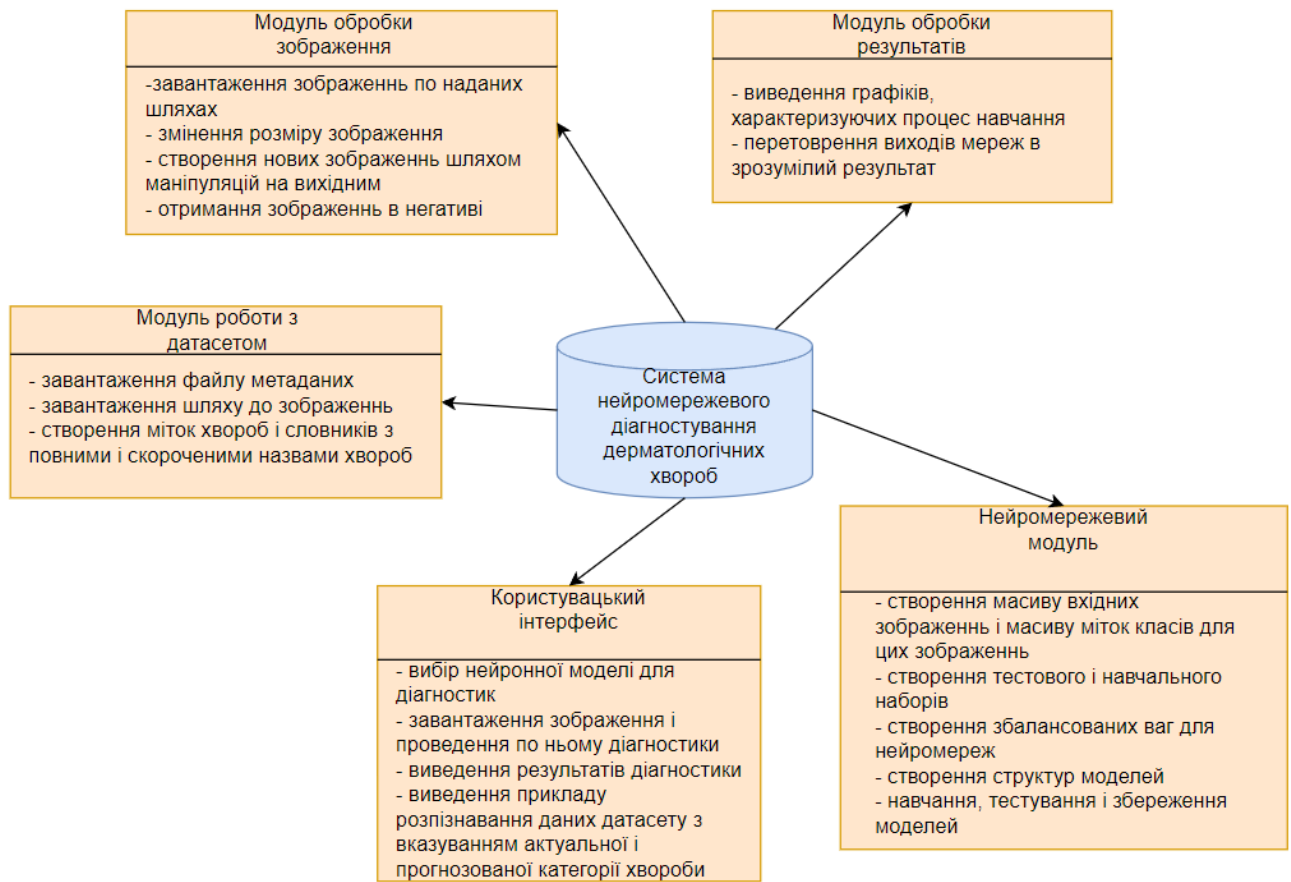


Рисунок 2.13 – Функції модулів

Користувацький інтерфейс є проміжним пунктом між системою і користувачем, дозволяючи останньому взаємодіяти з модулями через кнопки, форми, діаграми. Основна функція даного модулю – форма завантаження зображення з хворобою, коли користувач завантажує зображення безпосередньо з комп'ютера або мобільного пристрою, використовуючи стандартну форму завантаження файлів.

Користувач не надає ніякі додаткові дані системи, окрім зображень.

Діаграма функцій користувача зображена на рисунку 2.14

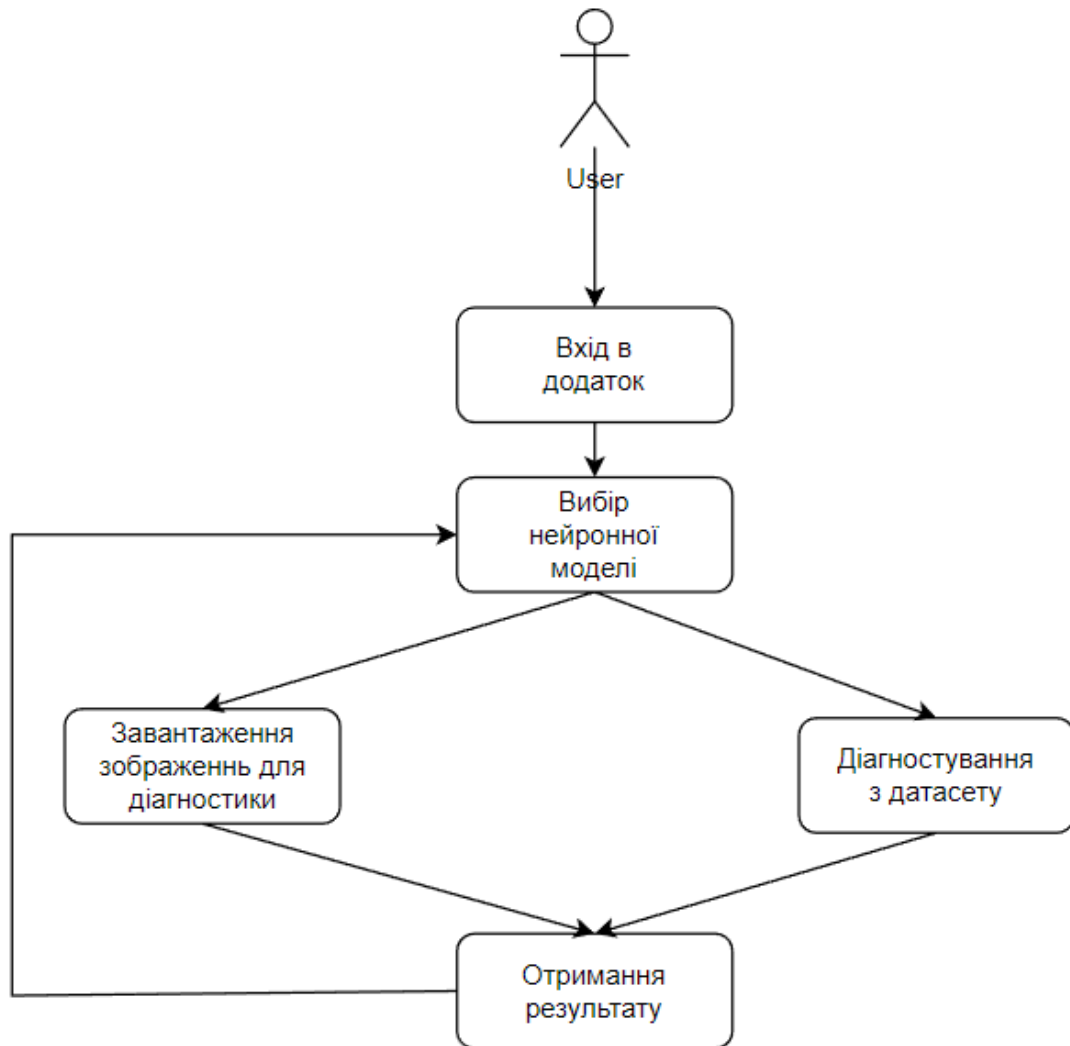


Рисунок 2.14 – Діаграма функцій користувача

Модуль роботи з датасетом загалом завантажує файли з датасету для подальшого опрацювання, яке включає створення міток класів для зображень і словників назв хвороб.

Модуль обробки зображення відповідальний за проведення певних змін на зображенні і відправку для подальшого виконання нейромережі. Головним чином, обробка включає різні техніки, як обертання, фільтрація, розмиття, збільшення або зменшення розміру зображення, налаштування контрастності та яскравості. Модуль бере на себе частину завдань по виділенню ознак на фотографії, окреслюючи розміри, текстури, форми.

Нейромережевий модуль містить алгоритми, які безпосередньо забезпечують проведення діагностики за зображеннями. Він включає нейромережеві моделі, попередньо навчені на зображеннях дерматологічних хвороб. Основна функція полягає у визначенні ймовірності наявності на зображенні аномалії, яка відноситься до одної категорії хвороби.

Модуль обробки результату отримує вихідні дані з нейромережевого модуля і на основі цих результатів формує власний, який містить виявлені проблеми і інформацію по кожній з них. Результати виводяться на екран, щоб користувач міг з ними ознайомитися.

Вищенаведені модулі в сукупності формують систему нейромережевого діагностування дерматологічних хвороб. Отже, було проаналізовано як повинна виглядати структура системи і які функції повинна реалізовувати, щоб ефективно виконувати поставлені завдання. Модулі обробки зображення, роботи з датасетом, і нейромережевий модуль, тісно пов'язані, беруть на себе завдання по підготовці даних для навчання і створенню нейромережевих моделей. Користувацький інтерфейс разом з модулем обробки результатів беруть на себе роботу по діагностуванню з використанням створених моделей.

2.5 Вибір комбінації засобів розробки інформаційної системи нейромережевого діагностування дерматологічних хвороб

Більшість мов програмування мають доступний функціонал для написання НМ будь-якого виду, але найпростіше і найзручніше створювати НМ з допомогою Python. Вона легка в освоєні, тому й стала настільки популярною, і на ній можна створювати додатки з різних сфер життєдіяльності людини. Python найбільше орієнтований на штучний інтелект і аналіз даних, а тому містить багатий функціонал для розробки НМ. До того ж, крос-платформенність Python дозволяє користуватися ним, працюючи з різними операційними системами.

Оскільки Python має відкритий вихідний код, для нього було створено велику кількість фреймворків, серед них найвідоміші: Django, CherryPy, Flask, TensorFlow. Це, насамперед, TensorFlow [20], який містить функції для навчання різних типів НМ. З допомогою використання графічних і тензорних процесорів значно прискорюється процес навчання згорткових НМ. TensorFlow був створений компанією Google, і спеціалізується в багатьох напрямках МН, дає доступ до процесу моніторингу навчання нейронних моделей, а також її просто освоювати через велику кількість доступних ресурсів для навчання. Додатком до TensorFlow використовують Keras [21], який має великий функціонал в написанні додатків комп'ютерного зору з використанням для навчання графічних процесорів. Keras з легкістю справляється з такими завданнями як класифікація, регресія, кластеризація і має вбудовані інструменти візуалізації процесу навчання. Таким чином, Keras та TensorFlow є тими засобами, з допомогою яких розробка НМ стає максимально простим і швидким завданням.

Чудовим редактором коду на мові Python може служити JupyterNotebook дистрибутива Anaconda [22]. Він дозволяє запускати код по частинах, перевіряючи кожен функцію окремо, має зрозумілий інтерфейс і дозволяє візуалізувати процес виконання коду у вигляді графіків та інших доступний форматів. В дистрибутиві Anaconda міститься графічний інтерфейс Anaconda Navigator, через який можна з легкістю завантажувати потрібні пакети фреймворків і створювати середовища з вже налаштованим вибором інструментів роботи. Цей навігатор містить великий набір програм: JupyterLab, JupyterNotebook, Orange, Spyder, Rstudio і VisualStudioCode.

Отже, вищеописаних засобів розробки достатньо задля створення ІС нейромережевого діагностування дерматологічних хвороб.

2.6 Висновки до розділу 2

Отже, в даному розділі було розглянуто проєктування нейромережевого методу діагностування дерматологічних хвороб, розподіленого по крокам, які включають завантаження датасету, попередню обробку, обробку вхідних даних згортковою нейромережею (на вибір користувача) та аналіз вихідних результатів.

Було розроблено нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями, який дозволяє одержати результат у вигляді діагностованої хвороби за фотозображеннями уражених ділянок шкіри.

Для реалізації нейромережевого методу діагностування дерматологічних хвороб обрано і розроблено структуру нейронних моделей на основі архітектур InceptionV3 та DenseNet з можливістю вибору користувачем потрібної, налаштовано їхні параметри, при яких вони будуть найефективніше проводити діагностики, дано повний опис їхніх особливостей в завданнях класифікації і опис компонентів, з яких складаються.

Для навчання моделей потрібно використати набори даних з відповідними зображеннями. Тому було підготовлено датасет, розподілено його на вибірки для навчання, валідації, і тестування, а також заповнення додатковими даними задля вирішення проблеми дисбалансу категорій.

Для взаємодії спроектованих частин системи нейромережевого діагностування потрібно було реалізувати функціональні модулі, кожен з яких відповідатиме за певний набір функцій і формуватимуть цілісну систему. Було спроектовано модулі, які відповідають за зчитування даних з датасету, обробку зображень, модуль навчання і налаштування НМ, і модуль створення діагностик. Описано функції кожного, і їхню взаємодію. Було проаналізовано як повинна виглядати структура системи і які функції повинна реалізовувати, щоб ефективно виконувати поставлені завдання. Модулі обробки зображення, роботи з датасетом, і нейромережевий модуль, тісно пов'язані, беруть на себе завдання

по підготовці даних для навчання і створенню нейромережових моделей. Користувацький інтерфейс разом з модулем обробки результатів беруть на себе роботу по діагностуванню з використанням створених моделей.

Було вибрано засоби розробки ІС, фреймворки, бібліотеки мови програмування Python. Для написання програмного коду обрано JupyterNoteBook дистрибутива Anaconda.

Розділ 3 Програмна реалізація інформаційної системи нейромережевого діагностування дерматологічних хвороб

3.1 Структура модулів інформаційної системи й їх взаємозв'язок

Структурою модулів системи нейромережевого діагностування дерматологічних хвороб є класи, які відповідають за наступний функціонал:

– «UpData» і «AddImages» – робота з датасетом і підготовка зображень для навчання.

– «training» – навчання і оцінка нейромережевої моделі.

– «gui» – інтерфейс користувача і обробка результатів.

Діаграму класів зображено нижче (рисунок 3.1)

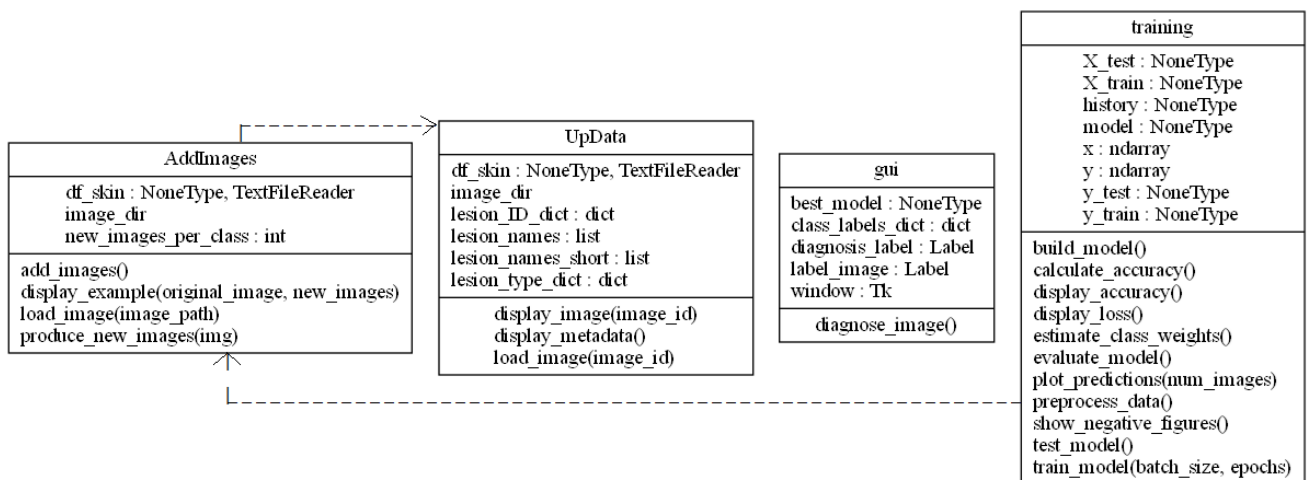


Рисунок 3.1 – Діаграма класів

Запуск програми здійснюється в класі «gui», в якій створюється інтерфейс користувача. Клас є незалежним від інших і опис його компонентів наведено в таблиці 3.1.

Таблиця 3.1– Клас «gui»

№	Компонент	Тип даних	Опис
1	best_model	Model	Завантаження попередньо навченої моделі
2	class_labels_dict	dict	Словник з індексами класів і відповідними мітками
3	diagnosis_label	Label	Віджет мітки, на якій відобразатиметься прогнозований клас захворювання
4	label_image	Label	Віджет з зображенням для діагностування
5	window	Tk	Об'єкт головного вікна застосунку
6	diagnose_image()		Функція для завантаження зображення і проведення діагностики

Клас «UpData» відповідальний за створення початкових даних системи, роботою з набором даних, зчитування файлів. Його компоненти коротко описані в таблиці 3.2.

Таблиця 3.2 – Клас «UpData»

№	Компонент	Тип даних	Опис
1	df_skin	DataFrame	DataFrame, який завантажує дані з csv-файлу з метаданими
2	image_dir	str	Шлях до директорії з зображеннями
3	lesion_ID_dict	dict	Словник, який співставляє типи захворювань з їхніми числовими ідентифікаторами
4	lesion_names	list	Список, що містить повні назви захворювань

Продовження таблиці 3.2

5	lesion_names_short	list	Список, що містить скорочені назви захворювань
6	lesion_type_dict	dict	Словник, який співставляє типи захворювань з їхнім описом
7	display_image (image_id)		Відображає зображення по заданому ідентифікатору
8	display_metadata()		Відображає інформацію про метадані, включаючи загальну кількість зображень і, окремо, різного
9	load_image (image_id)		Завантажує зображення по заданому ідентифікатору

Клас «AddImages» використовує дані, зчитані в «UpData» для доповнення даного датасет новими зображеннями, створеними завдяки маніпуляції з існуючими. Датасет є дуже нерівномірний по кількості зображень в кожній категорії, а тому, це компенсується новими для тих категорій, де їх не вистачає. Елементи класу «AddImages» можна розглянути в таблиці 3.3.

Таблиця 3.3 – Клас «AddImages»

№	Компонент	Тип даних	Опис
1	df_skin	DataFrame	DataFrame, який завантажує дані з csv-файлу з метаданими
2	image_dir	str	Шлях до директорії з зображеннями
3	new_images_per_class	int	Кількість нових зображень, згенерованих для кожного класу

Продовження таблиці 3.3

4	add_images()		Метод, який проходить по директорії з зображенням, і додає згенеровані в списки
5	display_example (original_image, new_images)		Виводить зображення, згенеровані на основі вихідного
6	load_image (image_path)		Завантажує зображення по заданому шляху
7	produce_new_images (img)		Метод, який генерує нові зображення, використовуючи техніку обертання і відзеркалення

Клас «training», в загальному, зосереджений на роботі з нейромережевою моделлю. На вхід йому подаються вже оброблені дані з попередніх класів. Його елементи наведено в таблиці 3.4.

Таблиця 3.4 – Клас «training»

№	Компонент	Тип даних	Опис
1	X_test	NoneType	Тестовий набір даних зі вхідними зображеннями
2	X_train	NoneType	Навчальний набір даних зі вхідними зображеннями
3	history	NoneType	Історія навчання моделі
4	model	NoneType	Об'єкт моделі нейронної мережі
5	x	ndarray	Масив вхідних зображень

Продовження таблиці 3.4

6	y	ndarray	Масив міток класів для кожного зображення
7	y_test	NoneType	Мітки класів у форматі бінарних матриць для тестового набору даних.
8	y_train	NoneType	Мітки класів у форматі бінарних матриць для навчального набору даних.
9	build_model()		Додавання шарів до моделі, оптимізування, компіляція моделі
10	calculate_accuracy()		Обчислює загальну і на кожному класі точність моделі
11	display_accuracy()		Відображає графік точності моделі на навчальному та тестовому наборах даних.
12	display_loss()		Відображає графік втрат моделі на навчальному та тестовому наборах даних.
13	estimate_class_weights()		Обчислює ваги класів, враховуючи їхній дисбаланс у навчальному наборі даних.

Отже, описана структура модулів є просто організована, кожен модуль розділяє логіку системи і виконує певний ряд функцій, що дозволяє маніпулювати даними для полегшення розробки. Описаних функцій достатньо для повного розуміння роботи системи.

3.2 Особливості реалізації інформаційної системи нейромережевого діагностування дерматологічних хвороб

Оскільки для реалізації ІС нейромережевого діагностування дерматологічних хвороб використовувалися бібліотеки, які в багато разів полегшують роботу з датасетами, нейронними моделями, візуалізацією даних, не виникало труднощів в реалізації задуманого. Серед них найбільш відомі:

- numpy для математичних обчислень;
- pandas для маніпулювання даними;
- matplotlib для побудови графіків;
- keras для побудови та тренування НМ;
- opencv для обробки зображень в задачах комп'ютерного зору;
- os для викликів операційної системи;
- PIL для додаткової обробки зображень;
- tkinter для створення інтерфейсу користувача.

Особливу увагу потрібно виділити підготовці даних. Отримавши метадані з датасету (рисунок 3.2), можна дійти висновку, що виникає дисбаланс між кількістю зображень, виділених до кожного типу захворювання, що може викликати неточність в характеристизації типу захворювання. Діагностика буде прогнозувати весь час одну й ту саму хворобу при непоганій точності навчання, адже вона найчастіше зустрічається в датасеті і призводить до його перенавчання.

Кількість зображень 10015

Проблема не вирішена, адже Melanocytic частіше зустрічається, ніж інші мітки

Melanocytic nevi	6705
Melanoma	1113
Benign keratosis-like lesions	1099
Basal cell carcinoma	514
Actinic keratoses	327
Vascular lesions	142
Dermatofibroma	115
Name: lesion_type, dtype: int64	

Рисунок 3.2 – Вивід кількості зображень кожної категорії

Для вирішення проблеми була створена функція, яка буде виконувати певні зміни в вихідних зображеннях і завантажувати на їх основі змінені у датасет. Блок-схема алгоритму маніпулювання зі зображенням і заповнення масиву вхідних даних знаходяться нижче (рисунок 3.3).

Функція «produce_new_img» зчитує зображення, робить 5 нових з тією ж самою міткою з вихідного, обертаючи його за часовою, проти часової стрілки, і перевертанням, а в блок-схемі наданий код, який використовує цю функцію для тих зображень, категорія захворювання якого будь-яка, окрім Melanocytic nevi. Потім новостворені зображення потрапляють в масив вхідних зображень, а їхні мітки – в масив міток. Ось приклад, як виглядає такий список (рисунок 3.4).

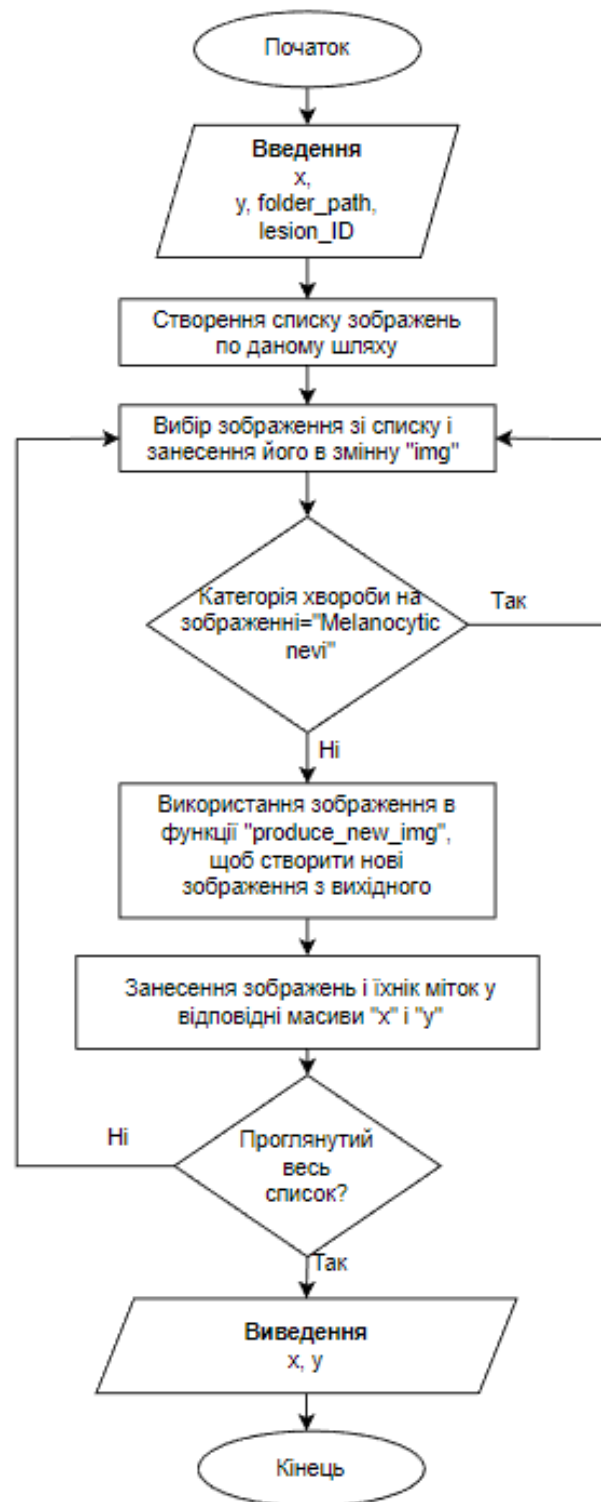


Рисунок 3.3 – Блок-схема алгоритму заповнення вхідного масиву зображень

Таким чином датасет розширюється до близько 25000 зображень, виділяючи 20000 на навчальну вибірку, і 5000 на тестувальну.

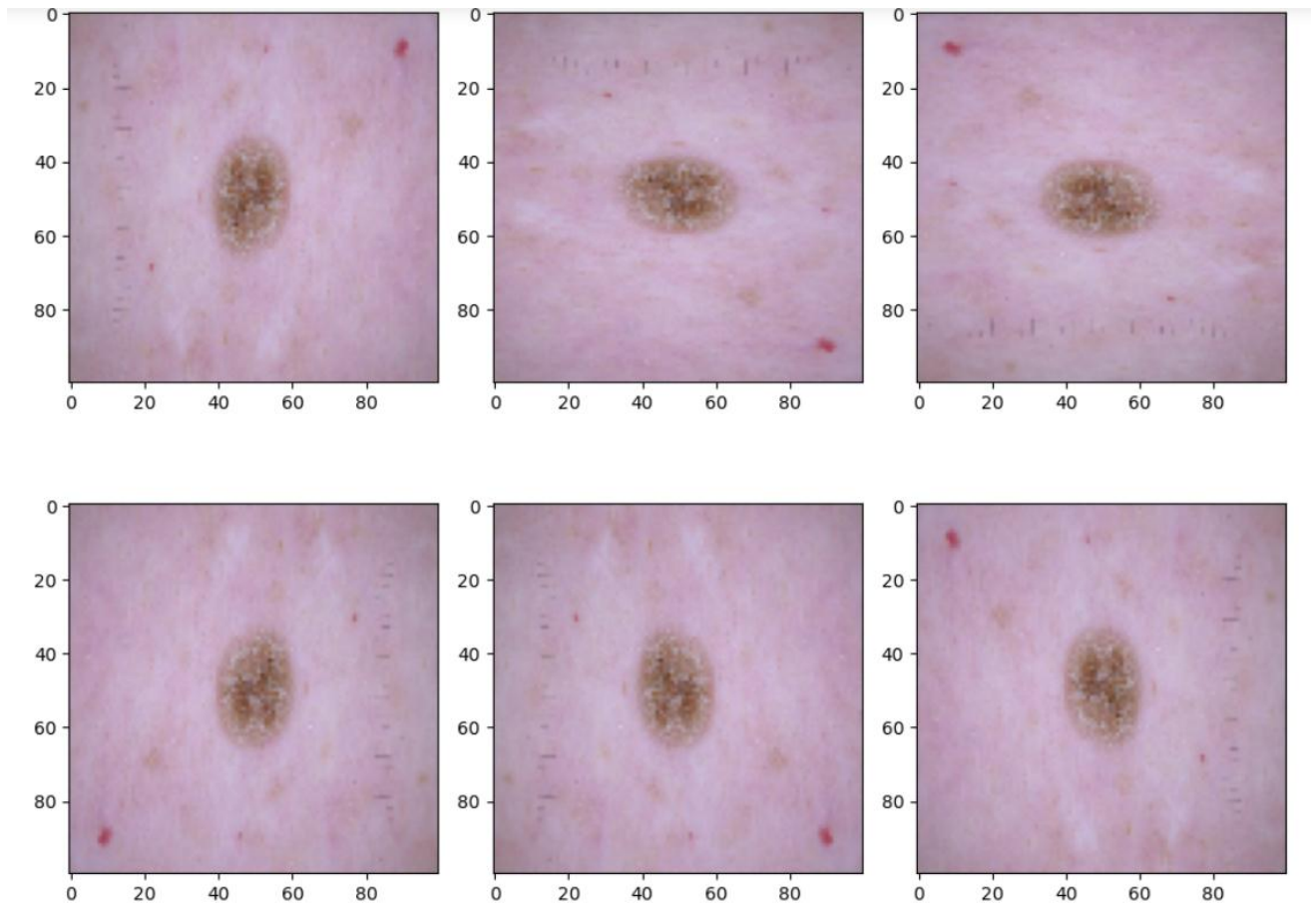


Рисунок 3.4 – Список нових зображень з вихідного

Також в системі є функція, яка для порівняння виводить 8 зображень з датасету, і робить діагностику по них. Блок-схема надана нижче (рисунок 3.5).

Дана схема показує вибір 8 випадкових зображень з датасету, діагностування їх з допомогою моделі DenseNet або InceptionV3, зберігає результат діагностики, актуальний діагноз на зображенні, і ймовірність правильного діагностування в відсотках.

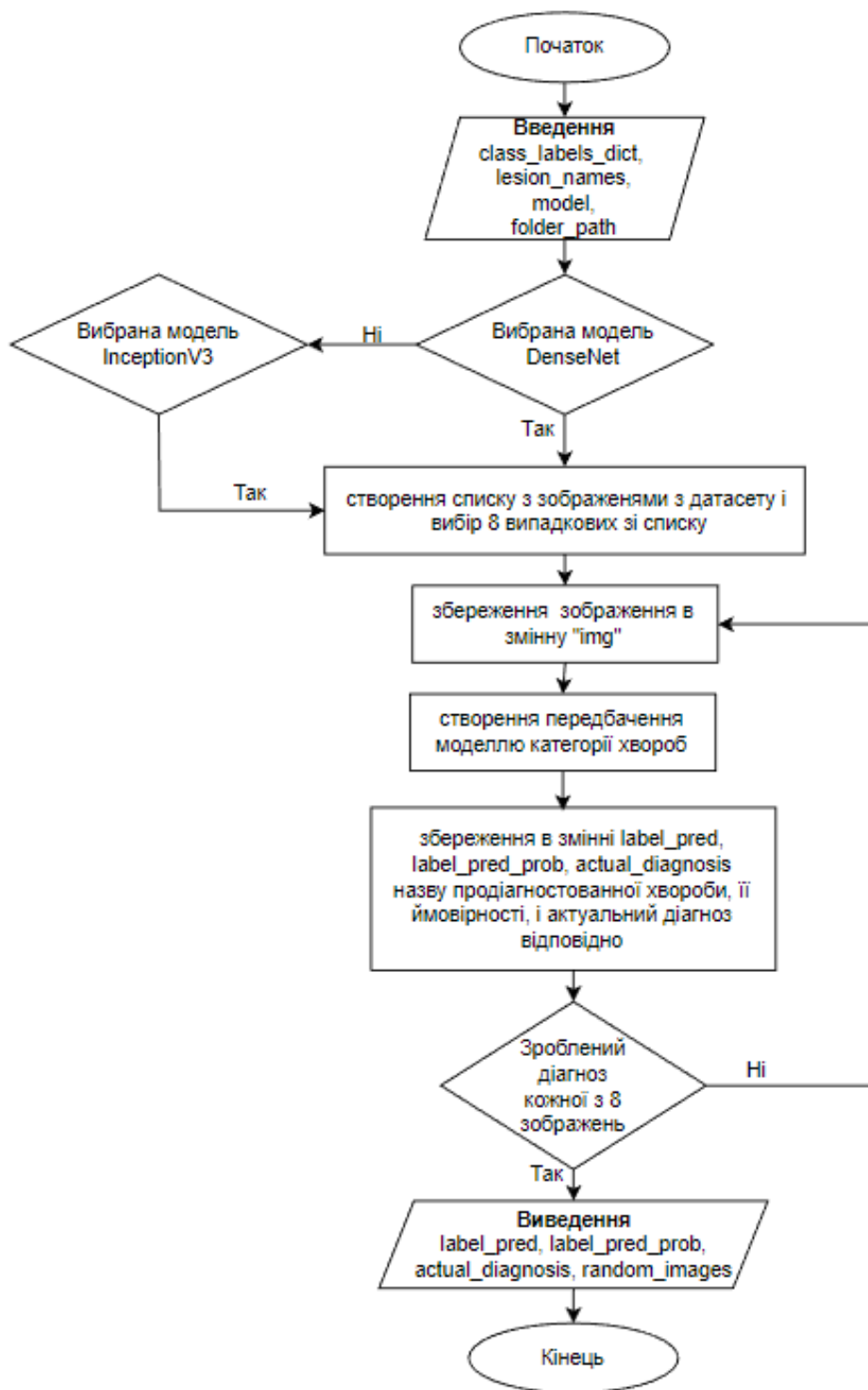


Рисунок 3.5 – Блок-схема «випадкового діагностування»

Отже, єдину проблему в реалізації ІС нейромережевого діагностування дерматологічних хвороб представляла незбалансованість даних для різних категорій захворювань. Її вирішення дало змогу налаштувати модель для більш

точного процесу діагностування. Також для більш наочного показу ефективності моделей, створена функція «випадкового діагностування».

3.3 Опис функціональних можливостей інформаційної системи

Головною функцією системи нейромережевого діагностування дерматологічних хвороб є визначення захворювання по його завантаженому зображенню. В інтерфейсі застосунку головний екран виглядає так (рисунок 3.6)

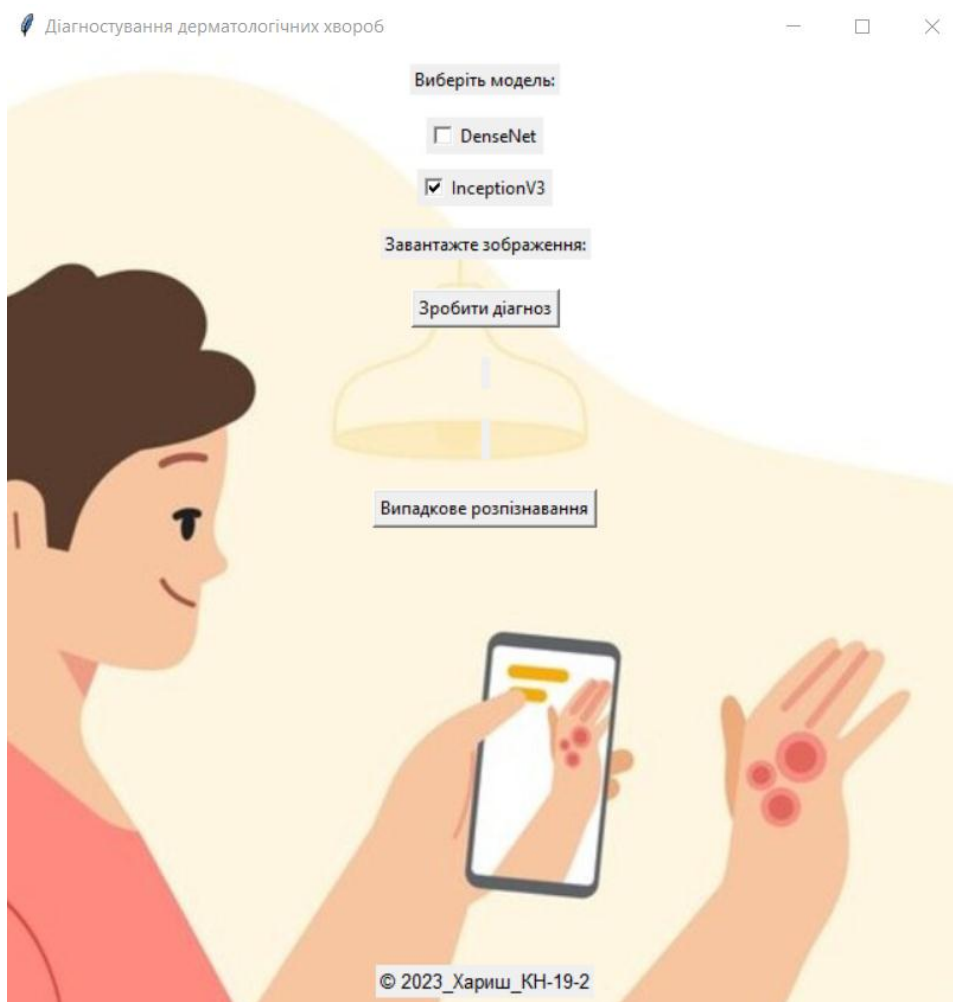


Рисунок 3.6 – Головний екран застосунку

В чекбоксі вибирається нейромережева модель, з допомогою якої проводитиметься діагностика. Натиснувши на кнопку «Зробити діагноз» ви

вибираєте зображення з ураженою ділянкою шкіри, яку хочете перевірити на наявність хвороби. Результатом буде зображення з вказуванням продіагностованої хвороби і відсотковим значенням. Ось як виглядає це в застосунку (рисунок 3.7)



Рисунок 3.7 – Проведена діагностика

Також користувач може переглянути, як модель розпізнає випадкові зображення з датасету, натиснувши на кнопку «Випадкове розпізнавання». Зображення будуть підписані їхніми актуальними і прогнозованими діагнозами для порівняння (рисунок 3.8)

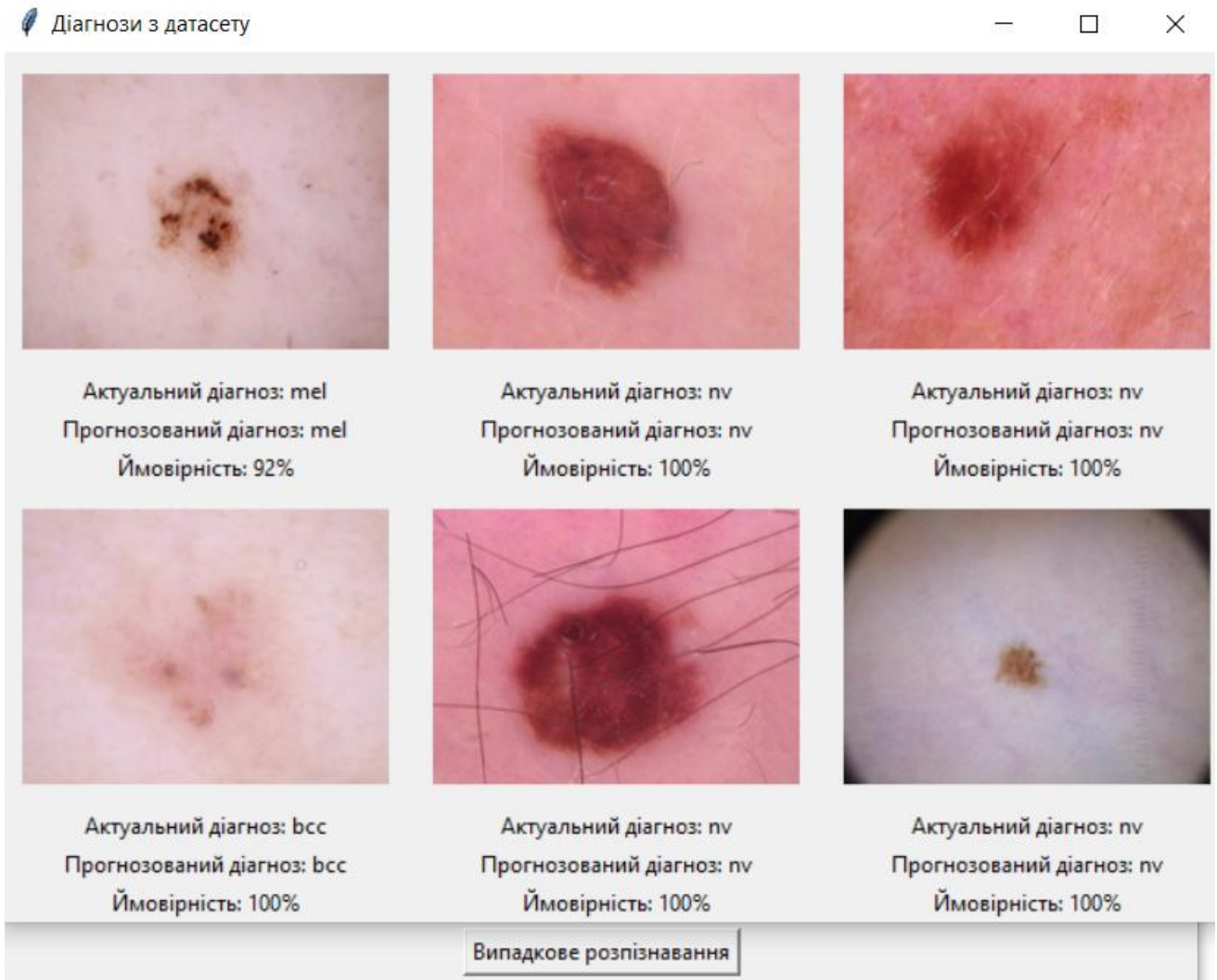


Рисунок 3.8 – Форма «Діагнози з датасету»

Щоб система працювала коректно, потрібно перевірити її на наявність помилок при виконанні. Для цього створюються тест-кейси, кожен з яких відповідає за окрему функцію.

При виконанні діагнозу може виникнути ситуація, що вибрана модель не існує, а тому має виводитися попередження. Тест-кейс цього випадку зображено в таблиці 3.5, вигляд попередження на рисунку 3.9.

Таблиця 3.5– Тест-кейс АТ0001

Тест-кейс ID: АТ0001	Пріоритет: 2	Створено: 15.05.2023, Хариш І. М.
Назва: Тестування завантаження обраної моделі Вхідні дані: checkbox з вибором нейромережевої моделі		
Кроки		Очікуваний результат
<p>Передумова: користувач вибрав модель, якої не створено або її не знайдено.</p> <ol style="list-style-type: none"> 1. Запустити програму 2. Вибрати в полі з чекбоксами варіант «VGG16» 3. Натиснути кнопку «Зробити діагноз» 4. Закрити діалог з попередженням 5. Вибрати варіант «DenseNet» 6. Натиснути кнопку «Зробити діагноз» 7. Отримати результат 		З'являється попередження «Нейронної моделі не знайдено»
Результат виконання тест-кейсу: пройдено успішно		

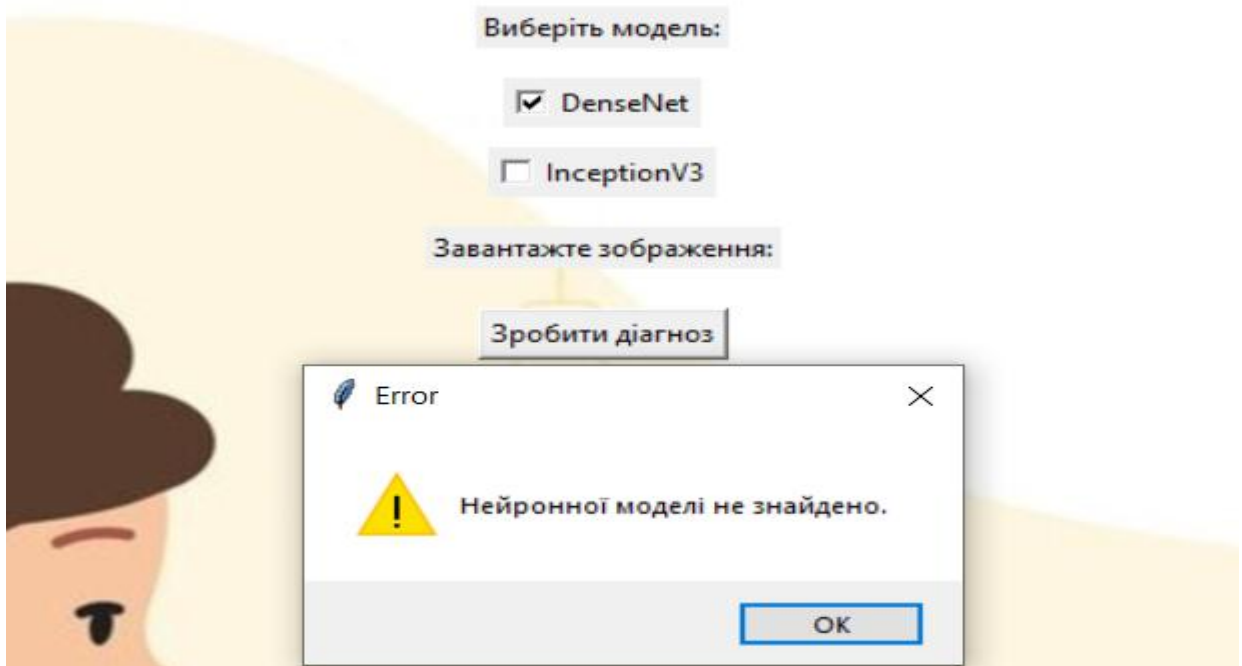


Рисунок 3.9 – Вигляд попередження про відсутність моделі

Користувач може завантажити в програму зовсім не зображення, а файли зовсім іншого типу, тому потрібно провести тестування правильності завантаження зображення для діагностики. На рисунку 3.10 зображено попередження, яке виникає при завантаженні неправильних вхідних даних.

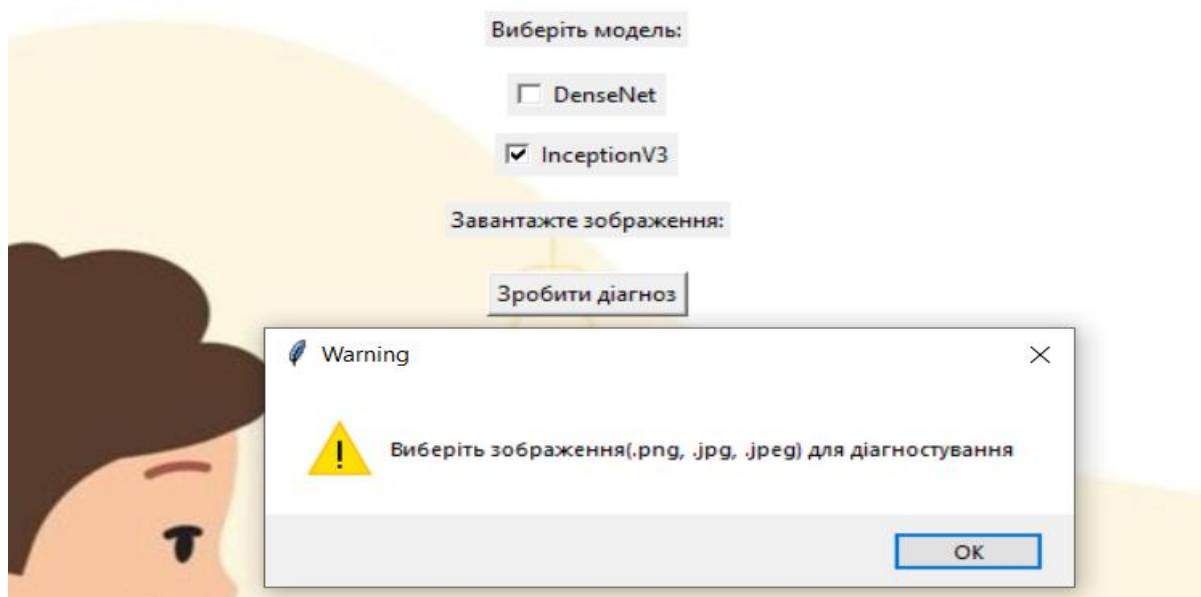


Рисунок 3.10 – Вигляд попередження про неправильний тип вхідних даних

Тест-кейс в даному випадку виглядає так (таблиця 3.6).

Таблиця 3.6– Тест-кейс АТ0002

Тест-кейс ID: АТ0001	Пріоритет: 1	Створено: 15.05.2023, Хариш І. М.
Назва: Тестування завантаження зображення		
Вхідні дані: натискання на кнопку «Діагноз» і вибір файлу		
Кроки	Очікуваний результат	
<p>Передумова: користувач вибрав файл з розширенням, відмінним від того, які використовуються для зображень.</p> <ol style="list-style-type: none"> 1. Запустити програму 2. Вибрати в полі з нейронну модель 3. Натиснути кнопку «Зробити діагноз» 4. Вибрати неправильний файл 5. Закрити діалог з попередженням 6. Натиснути кнопку «Зробити діагноз» 7. Вибрати зображення(.png, .jpg, .jpeg) 8. Натиснути кнопку «Зробити діагноз» 9. Отримати результат 	З'являється попередження «Виберіть зображення(.png, .jpg, .jpeg) для діагностування»	
Результат виконання тест-кейсу: пройдено успішно		

Якщо датасет не завантажено, то «Випадкове розпізнавання» не буде працювати, і видасть помилку Тест-кейс наведено нижче (таблиця 3.7). Результат виконання зображено на рисунку 3.11.

Таблиця 3.7 – Тест-кейс АТ0003

Тест-кейс ID: АТ0003	Пріоритет: 3	Створено: 15.05.2023, Хариш І. М.
Назва: Тестування розпізнавання з датасету Вхідні дані: натискання на кнопку «Випадкове розпізнавання»		
Кроки	Очікуваний результат	
<p>Передумова: користувач запустив випадкове розпізнавання, а датасет не завантажено.</p> <ol style="list-style-type: none"> 1. Запустити програму 2. Вибрати в полі нейронну модель 3. Натиснути кнопку «Випадкове розпізнавання» 4. Закрити діалог з попередженням 5. Завантажити датасет 6. Натиснути кнопку «Випадкове розпізнавання» 7. Отримати результат 	З'являється попередження «Датасет не знайдено»	
Результат виконання тест-кейсу: пройдено успішно		

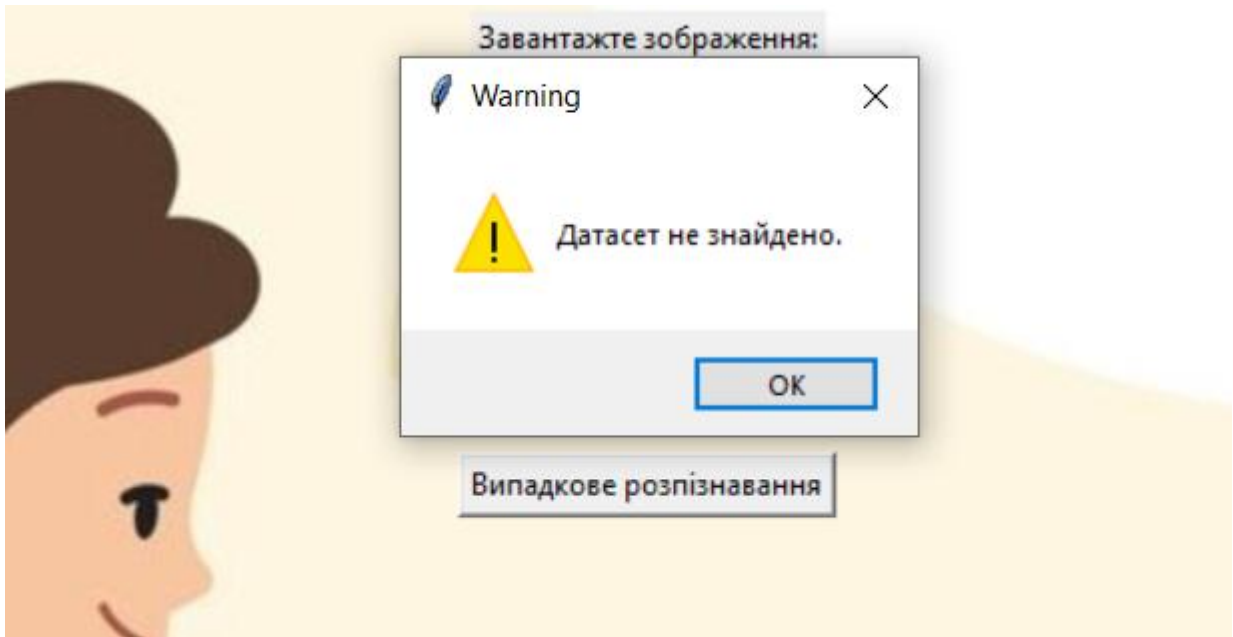


Рисунок 3.11 – Вигляд попередження про відсутність датасету

В даному розділі наведено та детально розглянуто функціональні можливості ІС нейромережевого діагностування дерматологічних хвороб. Наведено послідовність кроків, яку необхідно виконати користувачу для отримання результату. А також визначено потенційно вразливі місця в ІС, створено тест-кейси для доведення відсутності несправностей.

3.4 Дослідження ефективності застосування нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями

Дослідження нейромережевих методів діагностування дерматологічних хвороб проводилися і раніше, використовуючи датасет HAM10000 [23]. Результати таких експериментів зображено в таблиці 3.8.

Таблиця 3.8 – Точність нейронних моделей на датасеті HAM10000 [23]

Model Name	Peak Validation Accuracy(%)	Test Accuracy(%)
Basic VGG16	85.21	84.03
VGG16_Model1	86.38	85.47
VGG16_Model2	87.09	86.40
VGG16_Model3	83.94	83.32
VGG16_Model4(Proposed Method)	90.10	87.42
Basic VGG19	84.02	82.96
VGG19_Model1	85.82	84.84
VGG19_Model2	85.79	84.60
VGG19_Model3	82.95	82.37
VGG19_Model4(Proposed Method)	86.39	85.02
Basic MobileNet	87.09	85.87
MobileNet_Model1	87.90	86.76
MobileNet_Model2	86.84	85.62
MobileNet_Model3	85.57	84.42
MobileNet_Model4	88.73	87.31
MobileNet_Model5(Proposed method)	89.48	88.22

Щодо конкретної задачі класифікації, через використання найсучасніших, чудово структурованих архітектур НМ і вдалося досягти хорошої ефективності мереж. Також через збільшення розміру датасету майже втричі з допомогою маніпуляції з зображеннями мало свій вплив на навчання моделей. В кінцевому рахунку, були отримані наступні результати, які перевершують раніше описані (рисунки 3.12 – 3.13).

Для дослідження ефективності застосування нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями розраховано значення наступних метрик: точність, чутливість, F-міру.

Точність показує яка частка класифікованих об'єктів, є правильно класифікованими і її формула записується наступним чином.

$$Acc = \frac{TP}{(TP + FP)}, \quad (3.1)$$

де TP – правильно класифіковані об'єкти, FP – неправильно класифіковані об'єкти [24].

Чутливість визначає, наскільки система правильно ідентифікує справжні категорії захворювання. Її формула:

$$rec = \frac{TP}{(TP + FN)}, \quad (3.2)$$

де FN – об'єкти, які системи класифікує неправильно, що вони не належать до класу.

F-міра визначає баланс між чутливістю і точністю. Визначається так:

$$F = 2 \cdot (Acc \cdot rec) / (Acc + rec) \quad (3.3)$$

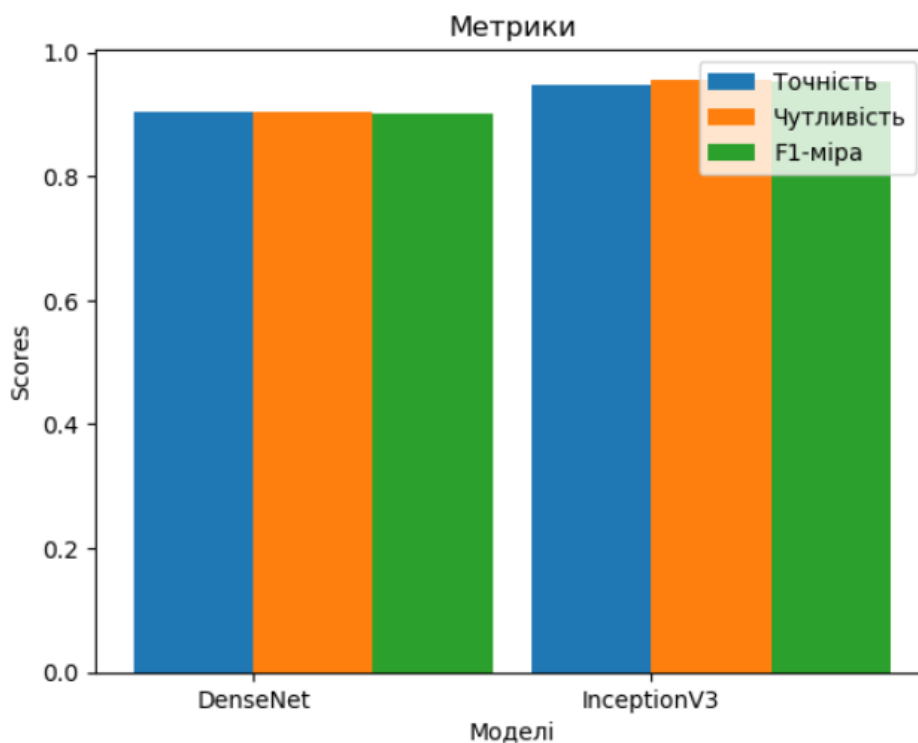


Рисунок 3.12– Метрики моделей

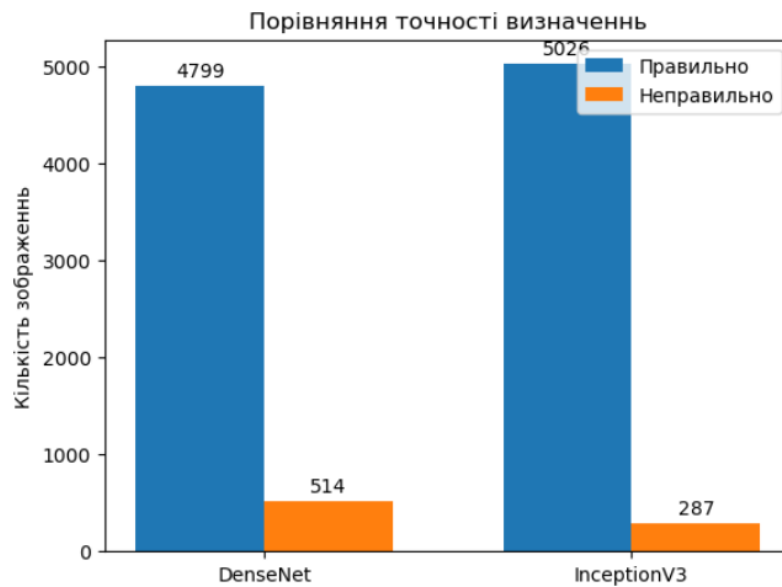


Рисунок 3.13 – Кількість правильно, неправильно створених діагнозів

З 5000 тестових зображень кількість неправильно продіагностованих знаходиться в районі 500(DenseNet) і 300(InceptionV3) зображень. Щодо точності визначення по категоріях захворювань, то моделі мають наступні результати (рисунок 3.14)

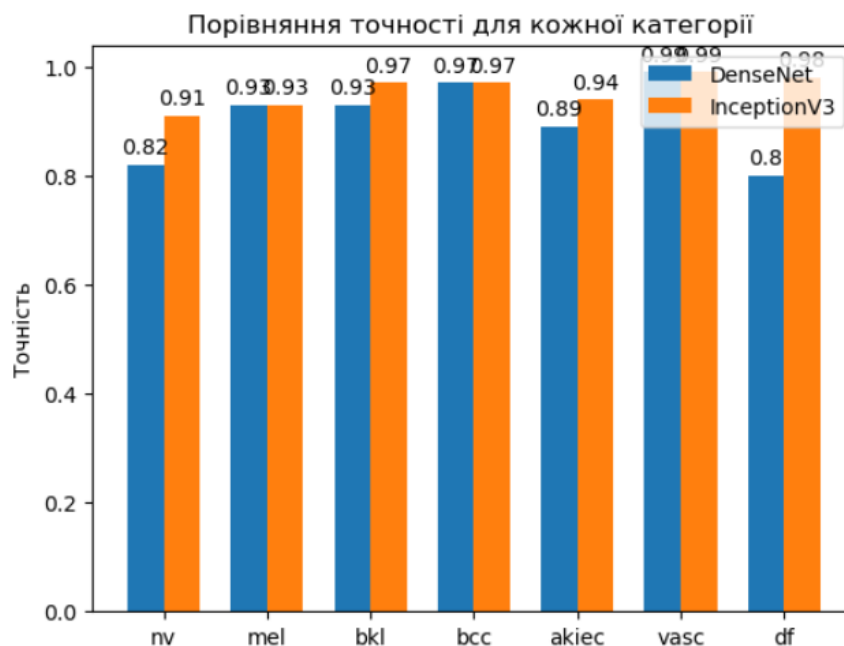


Рисунок 3.14 – Порівняння точності моделей для кожної категорії

Аналізуючи наведені результати, можна дійти висновку, що з допомогою МН можна досягти великої точності в діагностуванні дерматологічних хвороб, не покладаючись при цьому на оцінки лікарів-експертів. Застосовані архітектури нейронних мереж InceptionV3 та DenseNet для реалізації нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями дали високі показники ефективності, змогли показати високі результати з використанням підготовленого датасету, і тому можуть претендувати на використання в медичній сфері як помічники лікарям.

3.5 Висновки до розділу 3

Отже, в даному розділі описано програмну реалізацію ІС нейромережевого діагностування дерматологічних хвороб. Створення застосунку вимагало виконання етапів, описаних в розділі 2.

Розробка структури системи складалася з охарактеризування кожного модуля і реалізації задуманого функціоналу як робота з датасетом, попередня обробка зображень, навчання і тестування НМ, взаємодія з користувачем. Діаграми та таблиці класів змогли візуалізувати згадану структуру та описати функціонал кожної підсистеми і їхню взаємодію.

Під час реалізації програмного застосунку була вирішена основна проблема реалізації, яка заключалася в дисбалансі зображень до кожної категорії захворювань, були описані основні моменти, які максимально ефективно дозволили полегшити і покращити процес розробки ІС.

Сформовано інструкцію користувача з використання системи нейромережевого діагностування дерматологічних хвороб для проведення діагностик, як користувач взаємодіє з системою і ознайомлюється з можливостями нейромережевих моделей в медичній сфері.

Для порівняння ефективності сучасних нейронних моделей з раніше створеними моделями, було проведено експериментальне дослідження. Система була протестована на наборі даних HAM10000, які включають різні типи шкірних захворювань. Дослідження може включало оцінку метрик точності, чутливості, F-міри. Результати такого дослідження допомогли визначити, наскільки сучасні нейронні моделі дають прийнятні результати, і можуть слугувати підставою для рекомендацій щодо використання певної моделі в ІС діагностики дерматологічних хвороб [25].

Висновки

Кваліфікаційна робота бакалавра присвячена розробці нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями та ІС, яка використовує розроблений метод і дозволяє за зображеннями ураженого фрагменту шкіри людини визначати наявні дерматологічні хвороби за допомогою нейронної мережі.

Ознайомлення з предметною областю включало в себе аналіз зображень дерматологічних хвороб, створення їх опису для подальшого розуміння, як проектувати систему, що буде проводити діагностику, маючи на вході лише зображення з ураженими ділянками шкіри.

Проектування системи складалося з багатьох етапів, а тому потребувало багато інформації щодо МН, датасетів з потрібними даними в вигляді зображень ділянок шкіри з захворюванням, створенням структури модулів для майбутньої системи.

Розроблена ІС здатна встановлювати діагнози, визначаючи правильну категорію захворювання, використовуючи одні з сучасних нейромережевих моделей. Вона також надає можливість ознайомлення з ефективністю обраних моделей в діагностуванні даних з датасету.

Дослідження ефективності нейронних моделей в діагностуванні дерматологічних хвороб дає змогу оцінити надійність таким систем діагностування і підтвердити, наскільки сучасні архітектури НМ здатні справлятися з завданнями класифікації в медичній сфері.

Розроблений нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями дозволяє одержувати результат у вигляді діагностованої хвороби за фотозображеннями уражених ділянок шкіри.

Для покращення системи, ближньою перспективою є використання для діагностики не тільки зображень, а й метаданих, які йшли в комплекті датасету. Таким чином, в залежності від віку, статі, локалізації захворювання, можна отримати більше даних з процесу діагностики, а тому збільшити точність діагностування.

Перелік посилань

1. Будова і функції шкіри. URL: <https://miyklas.com.ua/p/biologiya/8-klas/sistema-organiv-vidilennia-termoreguliatciia-362467/budova-i-funktciyi-shkiri-gigiyena-shkiri-353422/re-137c36dc-bae3-4ec2-a563-71a6608edd9c>
2. Шкірні симптоми внутрішніх хвороб. URL: <https://tsn.ua/lady/zdorovyе/aktualnaya-tema/shkirni-simptomi-vnutrishnih-hvorob-1949581.html>
3. Згорткова нейронна мережа. URL: <https://evergreens.com.ua/ua/articles/cnn>
4. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. URL: <https://arxiv.org/abs/1905.11946>
5. A convolutional neural network trained with dermoscopic images of psoriasis performed on par with 230 dermatologists. URL: <https://pubmed.ncbi.nlm.nih.gov/34688173/>
6. DermAssist. URL: <https://health.google/consumers/dermassist/>
7. Класифікація типів шкіри за шкалою Фіцпатріка. URL: <https://ua.cosmohit.ua/article/315>
8. Skin Health Platform for Patients and Providers. URL: <https://miiskin.com/>
9. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. URL: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>
10. Оцінка класифікатора (точність, повнота, F-міра). URL: <https://www.bazhenov.me/blog/2012/07/21/classification-performance-evaluation.html>
11. DenseNet Explained. URL: <https://paperswithcode.com/method/densenet>

12. Review: DenseNet. Dense Convolutional Network (Image Classification).
URL: <https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803>
13. Advanced Guide to Inception v3. URL: <https://cloud.google.com/tpu/docs/inception-v3-advanced>
14. ImageNet Large Scale Visual Recognition Challenge (ILSVRC) URL: <https://www.image-net.org/challenges/LSVRC/>
15. ISIC 2018 Challenge [Closed]. URL: <https://challenge.isic-archive.com/landing/2018/>
16. ISIC. URL: <https://www.isic.org/>
17. Kaggle. DermNet. URL: <https://www.kaggle.com/datasets/shubhamgoel27/dermnet>
18. Kaggle. HAM10000. URL: <https://www.kaggle.com/>
19. Atrous Convolution with Transfer Learning for Skin Lesions Classification. URL: https://www.researchgate.net/publication/335390357_Atrous_Convolution_with_Transfer_Learning_for_Skin_Lesions_Classification
20. What is TensorFlow? The machine learning library explained. URL: <https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html>
21. What Is Keras: The Best Introductory Guide To Keras. URL: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras>
22. Anaconda. URL: <https://www.anaconda.com/download>
23. Atrous Convolution with Transfer Learning for Skin Lesions Classification. URL: https://www.researchgate.net/publication/335390357_Atrous_Convolution_with_Transfer_Learning_for_Skin_Lesions_Classification

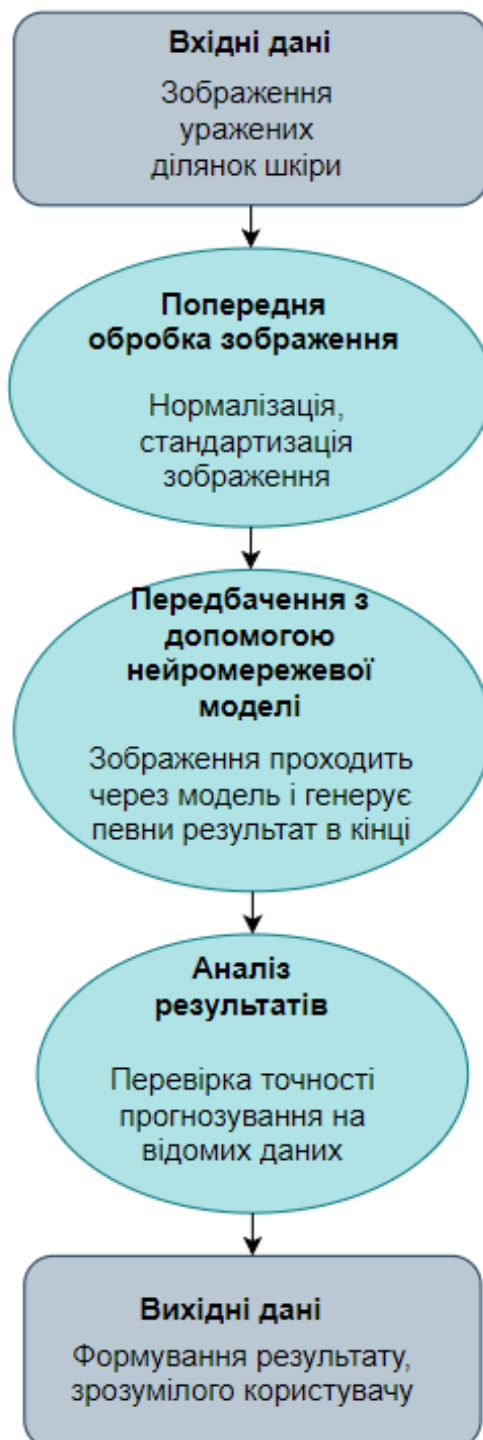
24. ResearchGate. Specificity, sensitivity and accuracy. URL: https://www.researchgate.net/figure/Performance-measurement-TP-TN-FP-FN-are-the-parameters-used-in-the-evaluation-of_fig3_347447352

25. ResearchGate. Clinical neurosciences – the spectrum of current implementation. URL: https://www.researchgate.net/publication/333264271_Clinical_neurosciences_-_the_spectrum_of_current_implementation

ДОДАТКИ

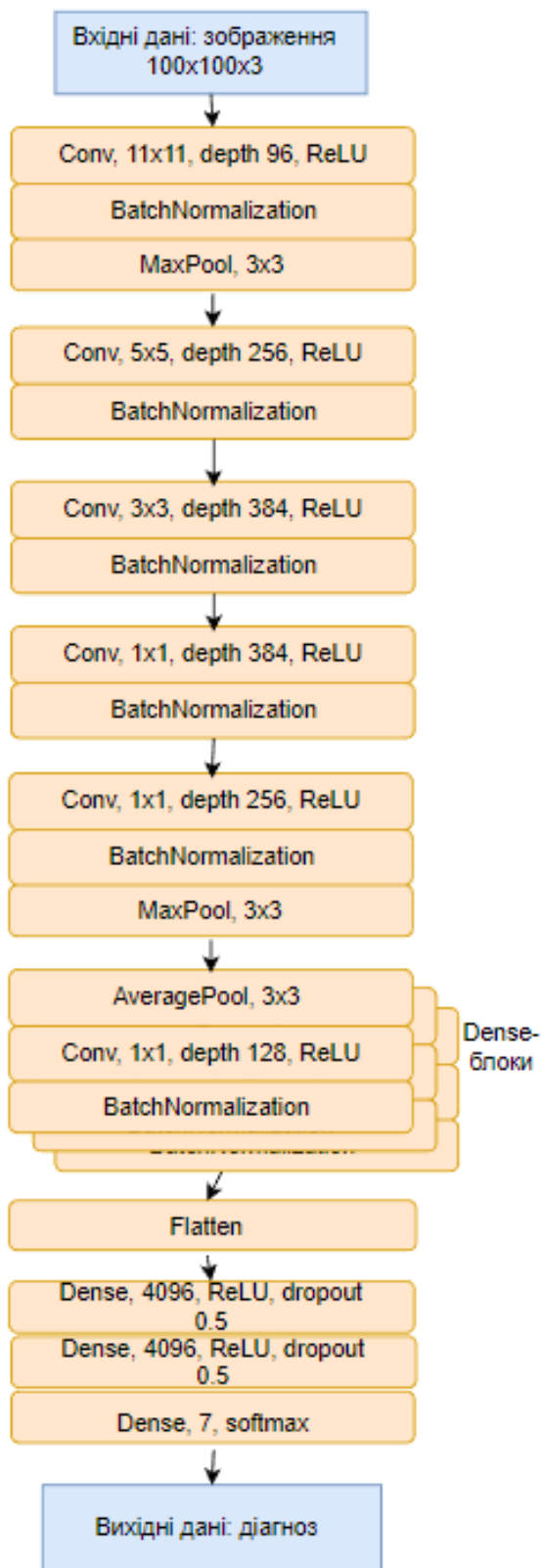
Додаток А

Схема неймережевого методу діагностування дерматологічних хвороб



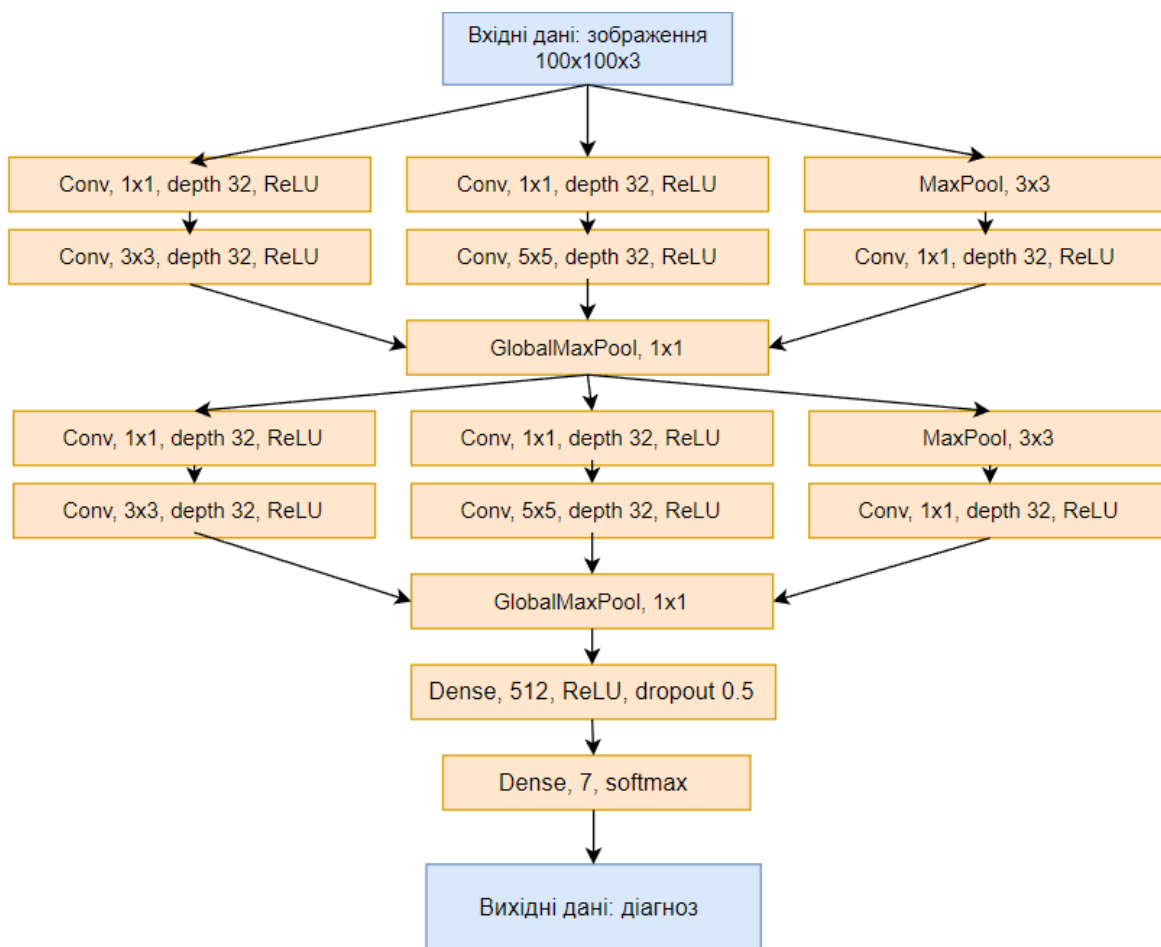
Додаток Б

Структура моделі з використанням архітектури DenseNet



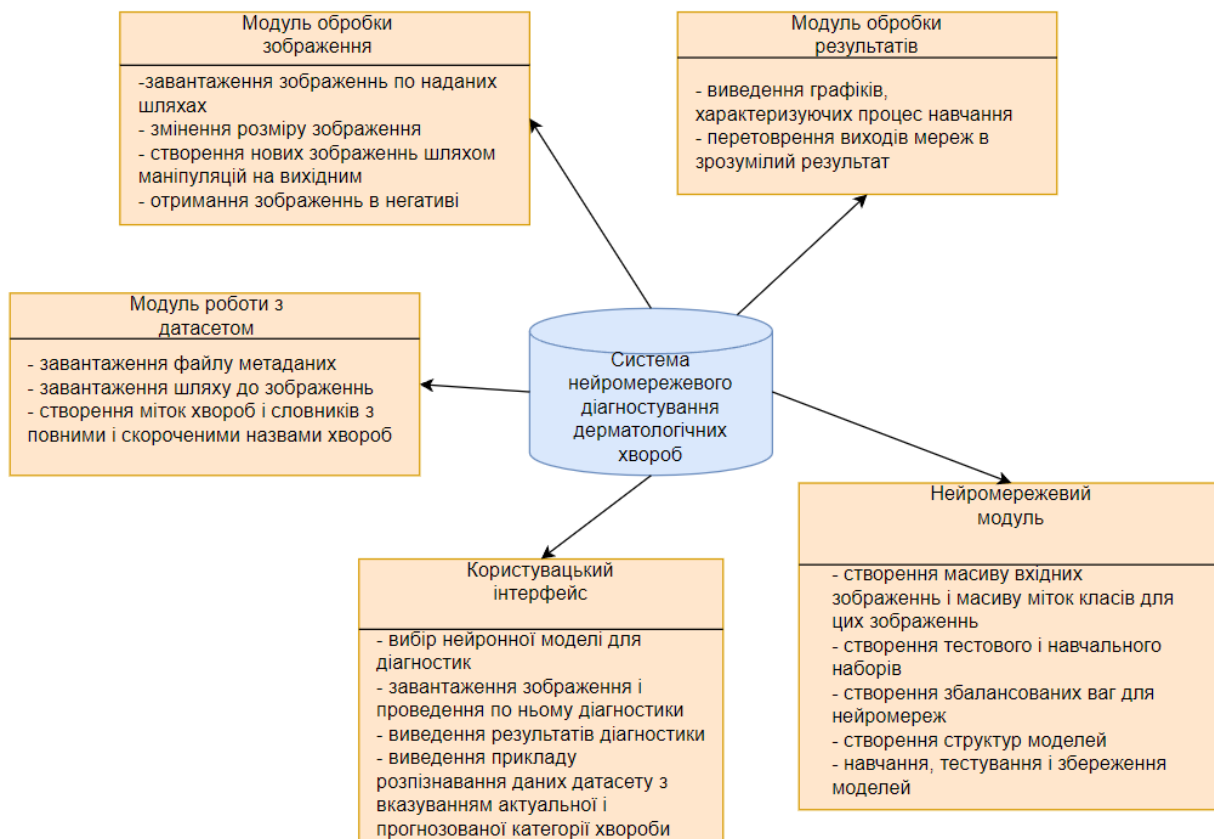
Додаток В

Структура моделі з використанням архітектури InceptionV3



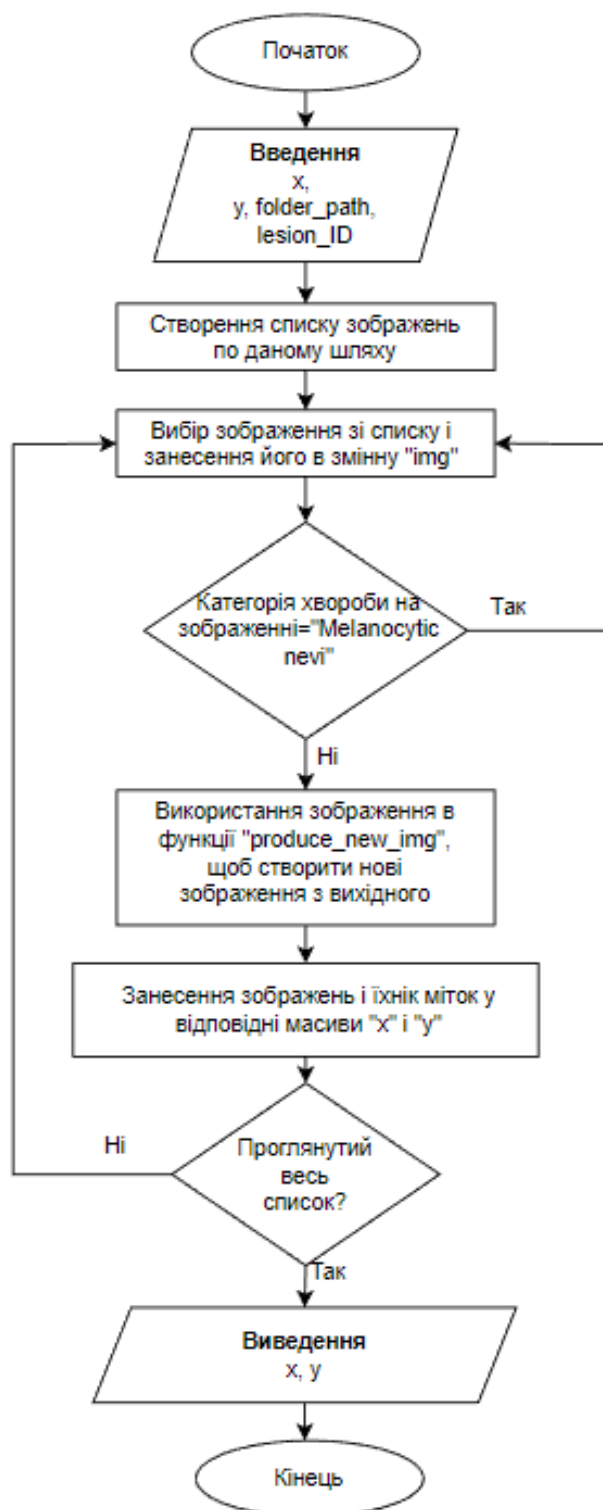
Додаток Г

Структура модулів ІС

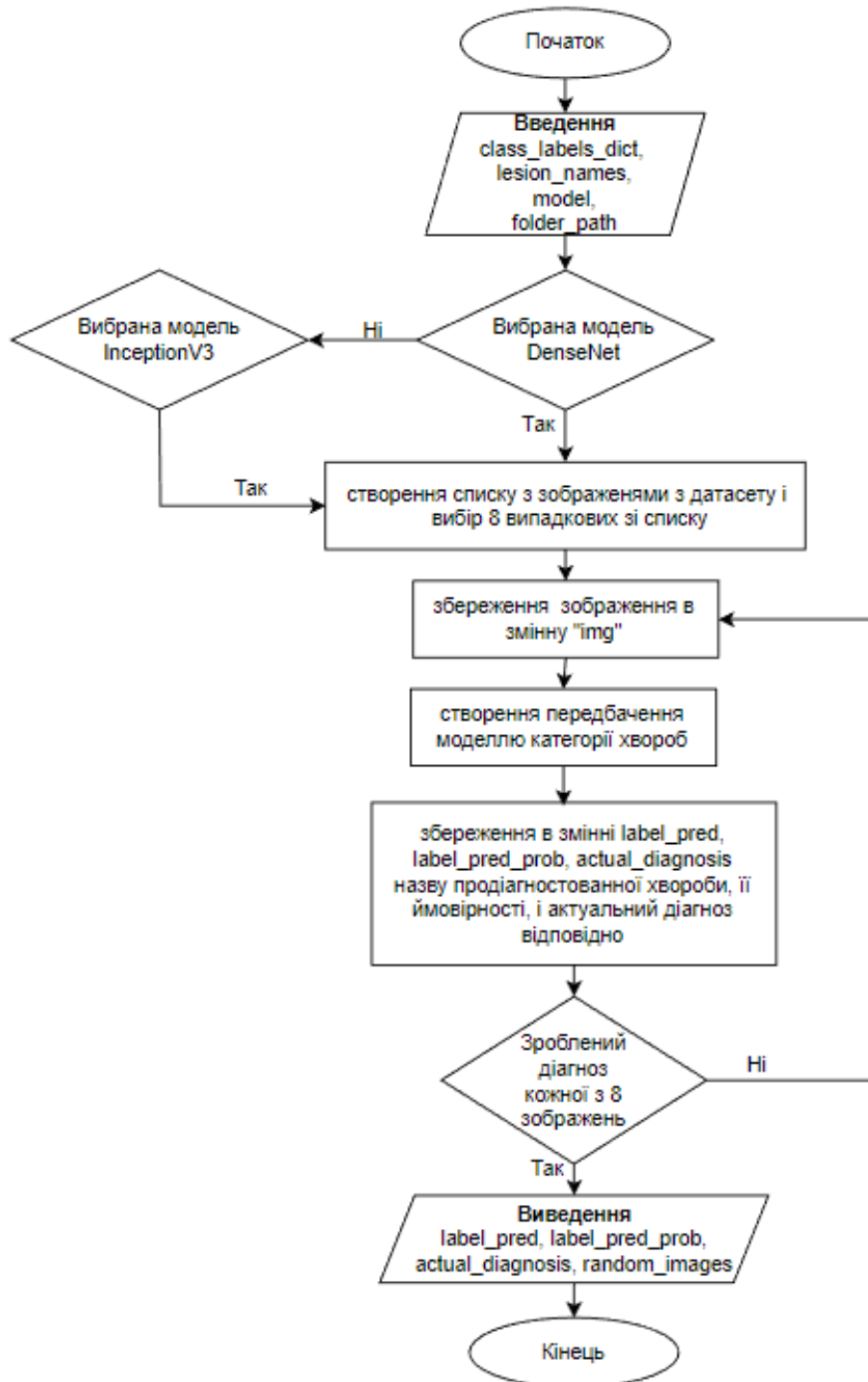


Додаток Д

Схема алгоритму заповнення вхідного масиву зображень



Додаток Е
Схема «випадкового діагностування»



Додаток Ж

Програмний код

Лістинг gui.py

```

from tkinter import *
from tkinter import filedialog, messagebox
from PIL import Image, ImageTk
import cv2
from cv2 import imread, resize
import numpy as np
import os
from keras.models import load_model
import random
import pandas as pd

class gui:
    def __init__(self):
        self.class_labels_dict = {
            0: 'nv',
            1: 'mel',
            2: 'bkl',
            3: 'bcc',
            4: 'akiec',
            5: 'vasc',
            6: 'df'
        }

        self.lesion_names = ['Melanocytic nevi', 'Melanoma', 'Benign keratosis-like
lesions',
                            'Basal cell carcinoma', 'Actinic keratoses', 'Vascular
lesions',
                            'Dermatofibroma']

        self.df_skin =
pd.read_csv('D:/edu/bachelor_diploma/HAM10000/HAM10000_metadata.csv')

        self.window = Tk()
        self.window.title("Діагностування дерматологічних хвороб")
        self.window.geometry('650x650')

        self.use_densenet = IntVar() # Нет значення по умовчанию

        self.background_image =
ImageTk.PhotoImage(Image.open("D:/edu/bachelor_diploma/img.jpg"))
        self.background_label = Label(self.window, image=self.background_image)
        self.background_label.place(x=0, y=0, relwidth=1, relheight=1)

        self.upload_label = Label(self.window, text="Виберіть модель:")
        self.upload_label.pack(pady=10)

        self.densenet_checkbox = Checkbutton(self.window, text="DenseNet",
variable=self.use_densenet, onvalue=1)
        self.densenet_checkbox.pack(pady=5)
        self.vgg16_checkbox = Checkbutton(self.window, text="InceptionV3",
variable=self.use_densenet, onvalue=0, offvalue=1)
        self.vgg16_checkbox.pack(pady=5)

```

```

self.upload_label = Label(self.window, text="Завантажте зображення:")
self.upload_label.pack(pady=10)
self.diagnose_button = Button(self.window, text="Зробити діагноз",
command=self.diagnose_image)
self.diagnose_button.pack(pady=10)

self.label_image = Label(self.window)
self.label_image.pack(pady=10)

self.diagnosis_label = Label(self.window, font=('Arial', 14, 'bold'))
self.diagnosis_label.pack(pady=10)

self.button_show_images = Button(self.window, text="Випадкове розпізнавання",
command=self.show_images)
self.button_show_images.pack(pady=10)

self.bottom_label = Label(self.window, text="© 2023_Хариш_КН-19-2",
font=('Arial', 10))
self.bottom_label.pack(side=BOTTOM, padx=10, pady=10)

def is_image_file(self, file_path):
    try:
        image = Image.open(file_path)
        image.verify()
        return True
    except (IOError, SyntaxError):
        return False

def diagnose_image(self):
    if self.use_densenet.get() == 1:
        model_path = 'D:/edu/bachelor_diploma/models/best_model.h5'
    else:
        model_path = 'D:/edu/bachelor_diploma/models/best_model1.h5'

    if not os.path.isfile(model_path):
        messagebox.showwarning("Error", "Нейронної моделі не знайдено.")
        return
    best_model = load_model(model_path)

    file_path = filedialog.askopenfilename()
    if file_path and os.path.isfile(file_path) and self.is_image_file(file_path):
        img = imread(file_path)
        img = resize(img, (100, 100))
        img = np.reshape(img, (1, 100, 100, 3))

        y_pred_prob = best_model.predict(img)
        y_pred = np.argmax(y_pred_prob, axis=1)
        label_pred = self.lesion_names[y_pred[0]]
        label_pred_prob = round(np.max(y_pred_prob[0]) * 100)

        image = Image.open(file_path)
        image.thumbnail((200, 200))
        photo = ImageTk.PhotoImage(image)

        self.label_image.configure(image=photo)
        self.label_image.image = photo

        self.diagnosis_label.configure(text="Діагноз: " +
self.lesion_names[y_pred[0]] + '(' + str(label_pred_prob) + '%)')
    else:

```

```

        messagebox.showwarning("Warning", "Виберіть зображення(.png, .jpg, .jpeg)
для діагностування")

def show_images(self):
    window1 = Toplevel(self.window)
    window1.title("Діагнози з датасету")

    row = 0
    col = 0

    if self.use_densenet.get() == 1:
        model_path = 'D:/edu/bachelor_diploma/models/best_model.h5'
    else:
        model_path = 'D:/edu/bachelor_diploma/models/best_model1.h5'

    if not os.path.isfile(model_path):
        messagebox.showwarning("Error", "Нейронної моделі не знайдено.")
        return
    best_model = load_model(model_path)

    try:
        folder_path = "D:/edu/bachelor_diploma/HAM10000/HAM10000/"
        all_images = os.listdir(folder_path)
        random_images = random.sample(all_images, 8)
    except:
        messagebox.showwarning("Warning", "Датасет не знайдено.")
        return

    for image_file in random_images:
        file_to_read = os.path.join(folder_path, image_file)
        img = cv2.imread(file_to_read)
        img_resized = cv2.resize(img, (100, 100))
        img_resized = np.reshape(img_resized, (1, 100, 100, 3))

        y_pred_prob = best_model.predict(img_resized)
        y_pred = np.argmax(y_pred_prob, axis=1)
        label_pred = self.lesion_names[y_pred[0]]
        label_pred_prob = round(np.max(y_pred_prob[0]) * 100)

        image = Image.open(file_to_read)
        image.thumbnail((200, 200))
        photo = ImageTk.PhotoImage(image)

        label_image = Label(window1, image=photo)
        label_image.image = photo
        label_image.grid(row=row, column=col, padx=10, pady=10)

        image_id = os.path.splitext(image_file)[0]
        actual_diagnosis = self.df_skin.loc[self.df_skin['image_id'] == image_id,
'dx'].values[0]

        label_actual = Label(window1, text=f"Актуальний діагноз:
{actual_diagnosis}")
        label_actual.grid(row=row + 1, column=col, padx=10)

        label_pred = Label(window1, text=f"Прогнозований діагноз:
{self.class_labels_dict[y_pred[0]]}")
        label_pred.grid(row=row + 2, column=col, padx=10)

        label_prob = Label(window1, text=f"Ймовірність: {label_pred_prob}%")

```

```

        label_prob.grid(row=row + 3, column=col, padx=10)

        col += 1
        if col == 4:
            col = 0
            row += 4

    def create_gui(self):
        self.window = Tk()
        self.window.title("Діагностування дерматологічних хвороб")
        self.window.geometry('650x650')

        background_image =
ImageTk.PhotoImage(Image.open("D:/edu/bachelor_diploma/img.jpg"))
        background_label = Label(self.window, image=background_image)
        background_label.place(x=0, y=0, relwidth=1, relheight=1)

        self.use_densenet = IntVar()

        upload_label = Label(self.window, text="Виберіть модель:")
        upload_label.pack(pady=10)

        densenet_checkbox = Checkbutton(self.window, text="DenseNet",
variable=self.use_densenet, onvalue=1)
        densenet_checkbox.pack(pady=5)

        vgg16_checkbox = Checkbutton(self.window, text="InceptionV3",
variable=self.use_densenet, onvalue=0,
                                offvalue=1)
        vgg16_checkbox.pack(pady=5)

        upload_label = Label(self.window, text="Завантажте зображення:")
        upload_label.pack(pady=10)

        diagnose_button = Button(self.window, text="Зробити діагноз",
command=self.diagnose_image)
        diagnose_button.pack(pady=10)

        self.label_image = Label(self.window)
        self.label_image.pack(pady=10)

        self.diagnosis_label = Label(self.window, font=('Arial', 14, 'bold'))
        self.diagnosis_label.pack(pady=10)

        button_show_images = Button(self.window, text="Випадкове розпізнавання",
command=self.show_images)
        button_show_images.pack(pady=10)

        bottom_label = Label(self.window, text="© 2023_Хариш_КН-19-2", font=('Arial',
10))
        bottom_label.pack(side=BOTTOM, padx=10, pady=10)

        self.window.mainloop()

```

Лістинг UpData.py

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import keras
from keras.models import Sequential, load_model
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras.layers.core import Dropout, Activation
from keras.layers import Conv2D, BatchNormalization, MaxPool2D, Flatten, \
    Dense, Input, Activation, Dropout, GlobalAveragePooling2D, AveragePooling2D
from keras.utils.np_utils import to_categorical
from sklearn.model_selection import train_test_split
from keras.models import Model
from sklearn.utils.class_weight import compute_class_weight
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
import cv2
from cv2 import imread, resize
from tensorflow.keras.optimizers import Adam
import os

class UpData:
    def __init__(self, metadata_file, image_dir):
        self.df_skin = pd.read_csv(metadata_file)
        self.image_dir = image_dir
        self.lesion_type_dict = {
            'nv': 'Melanocytic nevi',
            'mel': 'Melanoma',
            'bkl': 'Benign keratosis-like lesions',
            'bcc': 'Basal cell carcinoma',
            'akiec': 'Actinic keratoses',
            'vasc': 'Vascular lesions',
            'df': 'Dermatofibroma'
        }
        self.lesion_ID_dict = {
            'nv': 0,
            'mel': 1,
            'bkl': 2,
            'bcc': 3,
            'akiec': 4,
            'vasc': 5,
            'df': 6
        }
        self.lesion_names = [
            'Melanocytic nevi', 'Melanoma', 'Benign keratosis-like lesions',
            'Basal cell carcinoma', 'Actinic keratoses', 'Vascular lesions',
            'Dermatofibroma'
        ]
        self.lesion_names_short = ['nv', 'mel', 'bkl', 'bcc', 'akiec', 'vasc', 'df']

        self.df_skin['lesion_type'] = self.df_skin['dx'].map(self.lesion_type_dict)
        self.df_skin['lesion_ID'] = self.df_skin['dx'].map(self.lesion_ID_dict)

    def load_image(self, image_id):
        file_path = os.path.join(self.image_dir, f'{image_id}.jpg')
        image = imread(file_path)
        return image

    def display_metadata(self):

```

```

    print('Total number of images:', len(self.df_skin))
    print('The problem is unbalanced, since Melanocytic nevi is much more frequent
than other labels')
    print('Count of each lesion:')
    print(self.df_skin['lesion_type'].value_counts())

    def display_image(self, image_id):
        image = self.load_image(image_id)
        plt.imshow(image)
        plt.axis('off')
        plt.show()

```

Лістинг AddImages.py

```

class AddImages:
    def __init__(self, metadata_file, image_dir):
        self.df_skin = pd.read_csv(metadata_file)
        self.image_dir = image_dir
        self.new_images_per_class = 5

    def load_image(self, image_path):
        img = imread(image_path)
        img = resize(img, (100, 100))
        return img

    def produce_new_images(self, img):
        imga = cv2.rotate(img, cv2.ROTATE_90_CLOCKWISE)
        imgb = cv2.rotate(img, cv2.ROTATE_90_COUNTERCLOCKWISE)
        imgc = cv2.rotate(img, cv2.ROTATE_180)
        imgd = cv2.flip(img, 0)
        imge = cv2.flip(img, 1)
        new_images = imga, imgb, imgc, imgd, imge
        return new_images

    def display_example(self, original_image, new_images):
        plt.figure(figsize=(10, 8))
        plt.subplot(2, 3, 1)
        plt.imshow(original_image[:, :, :-1])
        for i in range(self.new_images_per_class):
            plt.subplot(2, 3, 2 + i)
            plt.imshow(new_images[i][:, :, :-1])
        plt.tight_layout()
        plt.show()

    def add_images(self):
        x = [] # Hold resized images
        y = [] # Hold image lesion ID from the data set

        # Handling images from part 1 directory
        lista1 = os.listdir(os.path.join(self.image_dir, 'HAM10000_images_part_1'))
        for fname_image in lista1:
            fname_ID = fname_image.replace('.jpg', '')
            file_to_read = os.path.join(self.image_dir, 'HAM10000_images_part_1',
fname_image)
            img = self.load_image(file_to_read)
            x.append(img)

```

```

        output = np.array(self.df_skin[self.df_skin['image_id'] ==
fname_ID].lesion_ID)
        y.append(output[0])

        if output != 0:
            new_images = self.produce_new_images(img)
            for i in range(self.new_images_per_class):
                x.append(new_images[i])
                y.append(output[0])

        if len(x) % 100 == 0:
            print(len(x), 'images loaded')

        # Handling images from part 2 directory
        lista2 = os.listdir(os.path.join(self.image_dir, 'HAM10000_images_part_2'))
        for fname_image in lista2:
            fname_ID = fname_image.replace('.jpg', '')
            file_to_read = os.path.join(self.image_dir, 'HAM10000_images_part_2',
fname_image)
            img = self.load_image(file_to_read)
            x.append(img)

        output = np.array(self.df_skin[self.df_skin['image_id'] ==
fname_ID].lesion_ID)
        y.append(output[0])

        if output != 0:
            new_images = self.produce_new_images(img)
            for i in range(self.new_images_per_class):
                x.append(new_images[i])
                y.append(output[0])

        if len(x) % 100 == 0:
            print(len(x), 'images loaded')

    return x, y

```

Лістинг training.py

```

class training:
    def __init__(self, x, y):
        self.x = np.array(x)
        self.y = np.array(y)
        self.y_train = None
        self.X_train = None
        self.X_test = None
        self.y_train = None
        self.y_test = None
        self.model = None
        self.history = None

    def preprocess_data(self):
        self.y_train = to_categorical(self.y, num_classes=7)
        self.X_train, self.X_test, self.y_train, self.y_test = train_test_split(self.x,
self.y_train, test_size=0.20, random_state=50, stratify=self.y)

    def show_negative_figures(self):
        _, ax = plt.subplots(1, 7, figsize=(30, 30))

```

```

for i in range(7):
    ax[i].set_axis_off()
    ax[i].imshow(self.X_train[i])
    ax[i].set_title(lesion_names[np.argmax(self.y_train[i])])
plt.show()

def estimate_class_weights(self):
    class_weights = np.around(compute_class_weight(class_weight='balanced',
classes=np.unique(self.y), y=self.y), 2)
    class_weights = dict(zip(np.unique(self.y), class_weights))
    return class_weights

def build_model(self):
    self.model = Sequential()
    # Add layers to the model

    # [+] 1st convolutional layer
    self.model.add(Conv2D(filters=96, kernel_size=(11, 11), strides=(4, 4),
activation='relu', input_shape=(100, 100, 3)))
    self.model.add(BatchNormalization())
    self.model.add(MaxPool2D(pool_size=(3, 3), strides=(2, 2)))

    # Add more layers

    # [+] 8th output layer
    self.model.add(Dense(7, activation='softmax'))

    # Compile the model

    optimizer = Adam(lr=0.0001, beta_1=0.9, beta_2=0.999, epsilon=1e-3)
    self.model.compile(optimizer=optimizer, loss='categorical_crossentropy',
metrics=['accuracy'])

    def train_model(self, batch_size=32, epochs=100):
        datagen = ImageDataGenerator(zoom_range=0.2, horizontal_flip=True,
shear_range=0.2)
        datagen.fit(self.X_train)

        early_stopping_monitor = EarlyStopping(patience=100, monitor='val_accuracy')
        model_checkpoint_callback = ModelCheckpoint(filepath='best_model.h5',
save_weights_only=False,
monitor='val_accuracy', mode='auto',
save_best_only=True,
verbose=1)

        self.history = self.model.fit(datagen.flow(self.X_train, self.y_train),
epochs=epochs, batch_size=batch_size,
shuffle=True, callbacks=[early_stopping_monitor,
model_checkpoint_callback],
validation_data=(self.X_test, self.y_test),
class_weight=self.estimate_class_weights())

    def display_accuracy(self):
        plt.plot(self.history.history['accuracy'])
        plt.plot(self.history.history['val_accuracy'])
        plt.title('model accuracy')
        plt.ylabel('accuracy')
        plt.xlabel('epoch')
        plt.legend(['accuracy', 'val_accuracy'], loc='upper left')
        plt.show()

```

```

def display_loss(self):
    plt.plot(self.history.history['loss'])
    plt.plot(self.history.history['val_loss'])
    plt.title('model loss')
    plt.ylabel('loss')
    plt.xlabel('epoch')
    plt.legend(['train', 'test'], loc='upper left')
    plt.show()

def test_model(self):
    y_pred = self.model.predict(self.X_test)
    total = 0
    accurate = 0
    accurateindex = []
    wrongindex = []
    for i in range(len(y_pred)):
        if np.argmax(y_pred[i]) == np.argmax(self.y_test[i]):
            accurate += 1
            accurateindex.append(i)
        else:
            wrongindex.append(i)
            total += 1
    return total, accurate

def evaluate_model(self):
    best_model = load_model('best_model.h5')
    scores = best_model.evaluate(self.X_test, self.y_test, verbose=1)
    return scores[1] * 100

def plot_predictions(self, num_images=16):
    best_model = load_model('best_model.h5')
    y_pred_prob = np.around(best_model.predict(self.X_test), 3)
    y_pred = np.argmax(y_pred_prob, axis=1)
    y_test2 = np.argmax(self.y_test, axis=1)

    plt.figure(figsize=(16, 16))
    for i in range(num_images):
        plt.subplot(4, 4, i + 1)
        index = i + 100
        plt.imshow(self.X_test[index, :, :, :-1])
        label_exp = lesion_names[y_test2[index]]
        label_pred = lesion_names[y_pred[index]]
        label_pred_prob = round(np.max(y_pred_prob[index]) * 100)
        plt.title('Expected:' + str(label_exp) + '\n Pred.:' + str(label_pred) + ' (' + str(label_pred_prob) + '%)')
        plt.ylabel('')
    plt.tight_layout()
    plt.show()

def calculate_accuracy(self):
    acc_tot = []
    for i in range(7):
        acc_parz = round(
            np.mean(self.y_test2[self.y_test2 == i] == self.y_pred[self.y_test2 ==
i]), 2)
        lab_parz = lesion_names[i]
        print('accuracy for', lab_parz, '=', acc_parz)
        acc_tot.append(acc_parz)

```

```
acc_tot = np.array(acc_tot)
freq = np.unique(self.y_test2, return_counts=True)[1]
accuracy = np.sum(acc_tot * freq) / np.sum(freq)
return accuracy
```

Додаток И
Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

**НЕЙРОМЕРЕЖЕВИЙ МЕТОД ДІАГНОСТУВАННЯ
ДЕРМАТОЛОГІЧНИХ
ХВОРОБ ЗА ЇХ ЗОБРАЖЕННЯМИ**



Виконав:
студент 4 курсу, групи КН-19-2
Хариш Іван Миколайович



Керівник:
викладач кафедри КН
Собко Олена Віталіївна

Актуальність

Шкіра – це надзвичайно важливий орган в організмі людини, адже він є частим індикатором того, які процеси відбуваються в людському тілі і як це відображається на її функціональності. Тому різні негативні прояви на ділянках шкіри можуть викликати незворотні зміни в організмі і призвести до важких наслідків. Нейромережі можуть бути навчені впізнавати характерні ознаки різних дерматологічних захворювань, використовуючи зображення шкіри. Актуальність шкіри в системі нейромережевого діагностування полягає в тому, що шкіра надає важливу інформацію про стан здоров'я пацієнта. Зображення шкіри може містити показники, такі як колір, текстура, рельєф, розмір, форма та розташування плям, висипань, ранок або інших змін.

Використовуючи навчену нейромережу, можна провести автоматичний аналіз цих ознак і порівняти їх з базою даних дерматологічних захворювань для діагностики конкретного стану шкіри пацієнта. Нейромережеві системи можуть надати швидку, точну та об'єктивну оцінку стану шкіри, допомагаючи лікарям у ранньому виявленні та лікуванні дерматологічних захворювань.

Мета і задачі роботи

Метою роботи є розробка нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями та інформаційної системи, яка використовує розроблений метод і дозволяє за зображеннями ураженого фрагменту шкіри людини визначати наявні дерматологічні хвороби за допомогою нейронної мережі. Для досягнення мети потрібно вирішити наступні задачі:

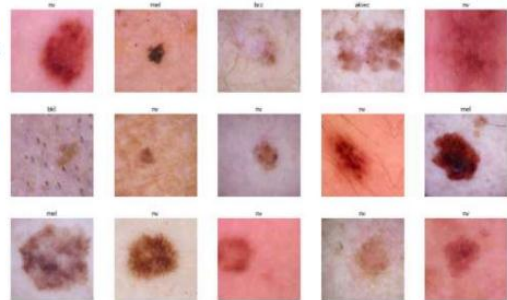
- розробити нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями;
- розробити архітектуру нейронної мережі для діагностування дерматологічних хвороб в людини за їх зображеннями;
- спроектувати структуру інформаційної системи, яка використовує розроблений метод і дозволяє за зображеннями ураженого фрагменту шкіри людини визначати наявні дерматологічні хвороби за допомогою нейронної мережі;
- виконати вибір засобів розробки інформаційної системи нейромережевого діагностування дерматологічних хвороб за їх зображеннями;
- виконати програмну реалізацію інформаційної системи нейромережевого діагностування дерматологічних хвороб;
- провести тестування створеної інформаційної системи.

Схема нейромережевого методу діагностування дерматологічних хвороб

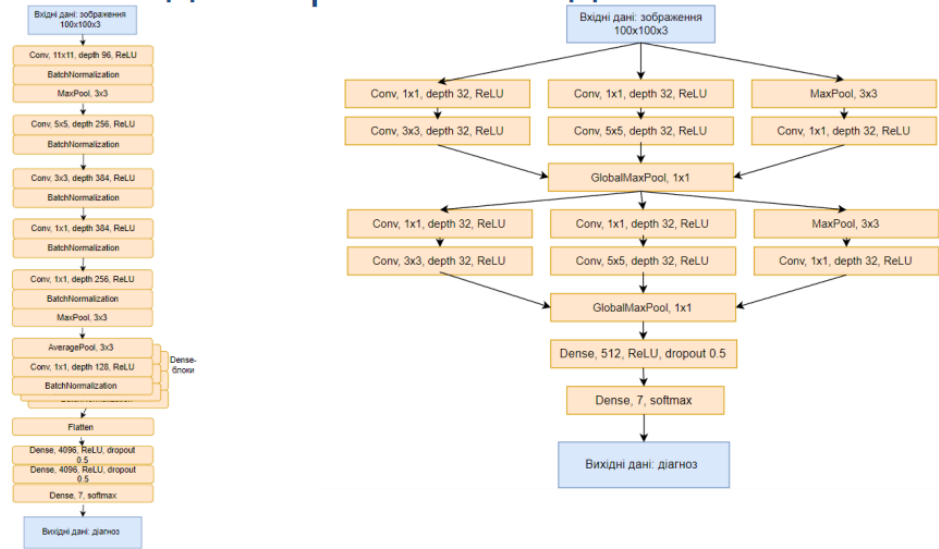


Зображення для навчання з датасету HAM10000

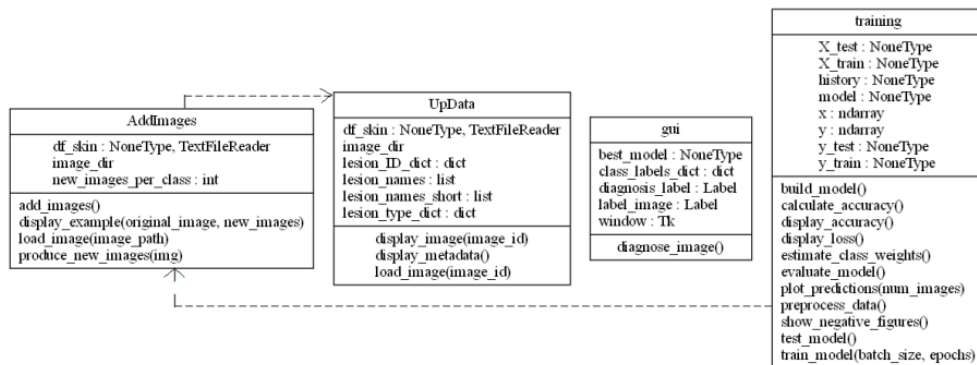
- akiec – Actinic Keratoses (актинічний кератоз) и intraepithelial carcinoma / Bowen's disease (внутрішньоепітеліальний рак / хвороба Боуэна).
- bcc – Basal cell carcinoma (базаліома).
- bkl – Benign Keratosis-like Lesions (бенджинові кератозоподібні утворення)
- df – Dermatofibroma (дерматофіброма).
- melanoma – Melanoma (меланома)
- nv – Melanocytic nevi (меланоцитові родинки)
- vasc – Vascular Lesions (судинні утворення)



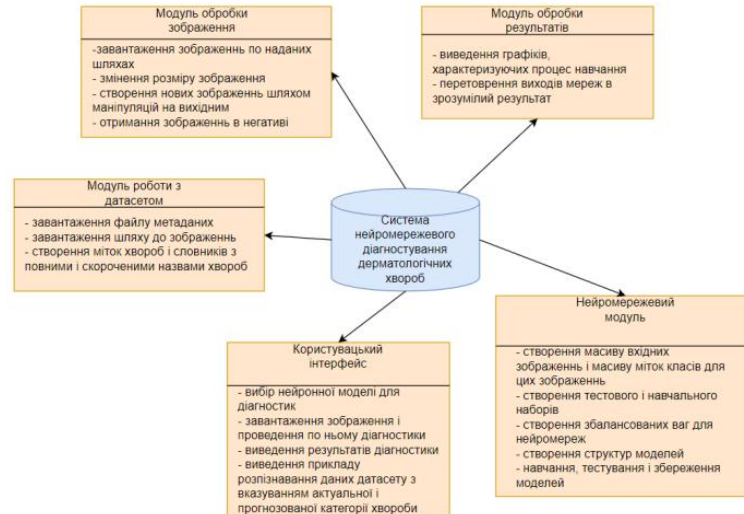
Розроблені нейронні моделі (DenseNet, InceptionV3) для вирішення завдання



Діаграма класів програмної реалізації методу нейромережевого методу діагностування дерматологічних хвороб



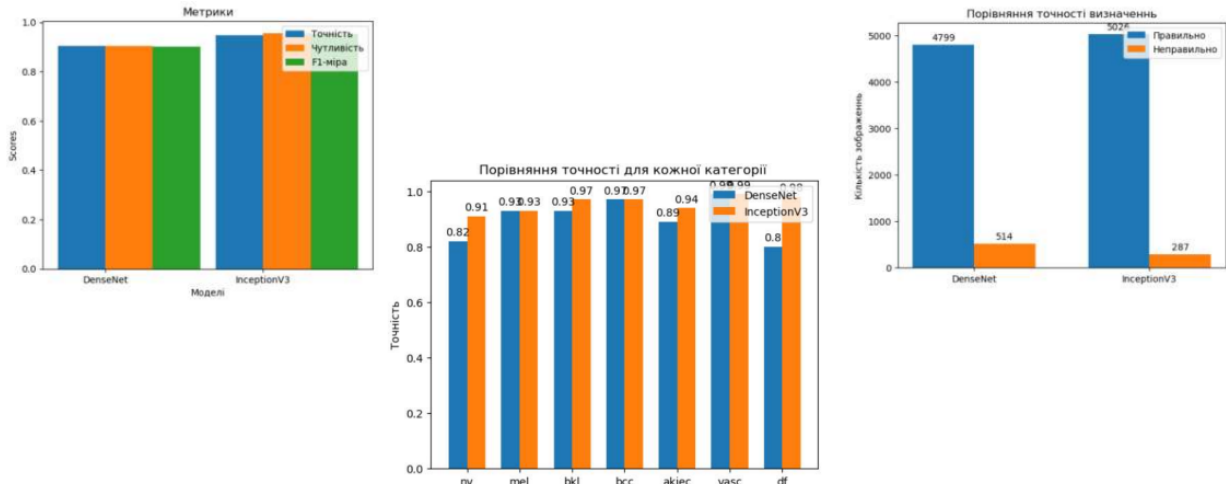
Структура інформаційної системи нейромережевого діагностування дерматологічних хвороб



Програмна реалізація методу нейромережевого діагностування дерматологічних хвороб



Дослідження ефективності розроблених моделей InceptionV3 та DenseNet для методу нейромережевого діагностування дерматологічних хвороб



Висновки

Кваліфікаційна робота бакалавра присвячена розробці нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями та інформаційної системи, яка використовує розроблений метод і дозволяє за зображеннями ураженого фрагменту шкіри людини визначати наявні дерматологічні хвороби за допомогою нейронної мережі.

Розроблено нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями, який дозволяє одержувати результат у вигляді діагностованої хвороби за фотозображеннями уражених ділянок шкіри.

Розроблена інформаційна система здатна створювати діагнози, з великою точністю визначаючи правильну категорію захворювання, використовуючи одні з найсучасніших нейромережевих моделей. Вона також надає можливість ознайомлення з ефективністю обраних моделей в діагностуванні даних з датасету.

Дослідження ефективності нейронних моделей в діагностуванні дерматологічних хвороб дає змогу оцінити надійність таким систем діагностування і підтвердити, наскільки сучасні архітектури НМ здатні справлятися з завданнями класифікації в медичній сфері.

Для покращення системи може знадобитися багато часу і ресурсів. Близькою перспективою є використання для діагностики не тільки зображень, а й метаданих, які йшли в комплекті датасету. Таким чином, в залежності від віку, статі, локалізації захворювання, можна отримати більше даних з процесу діагностики, а тому збільшити точність діагностування.

Ім'я користувача:
Кафедра КН

ID перевірки:
1015408649

Дата перевірки:
03.06.2023 20:56:11 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
03.06.2023 20:57:48 EEST

ID користувача:
100005671

Назва документа: КН-19-2 Хариш

Кількість сторінок: 68 Кількість слів: 8523 Кількість символів: 65699 Розмір файлу: 2.00 MB ID файлу: 1015072093

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

4.85% Схожість

Найбільша схожість: 1.28% з Інтернет-джерелом (<https://www.biorxiv.org/content/10.1101/746388v8.full.pdf>)

4.12% Джерела з Інтернету 210 Сторінка 70

2.01% Джерела з Бібліотеки 61 Сторінка 71

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 10

Підозріле форматування 14 сторінок

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 16%

ID: 114635 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА Додано в БД: 2023-06-03 Автора: І.М. Хариш Керівники: О.В. Собко Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	53043	809	1888 (4%)	26 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями

Автор: студент групи КН-19-2 Хариш Іван Миколайович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: викладач Собко О.В.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

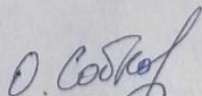
Запозичення, виявлені в роботі Хариша І.М., не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; серед запозичень знаходяться загальновідомі терміни, скорочення та матеріали статей.

Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:

- за системою Anti-Plagiarism: 1%;

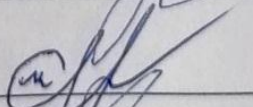
- за системою Unichек: 4.85 %.

Керівник роботи



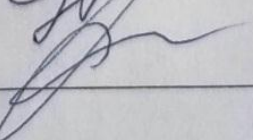
Олена СОБКО

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студента гр. КН-19-2 Харіша Івана Миколайовича

за темою: Нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями

1. Актуальність теми

Актуальним завданням, що розглядається в даній роботі, є розробка та реалізація нейромережевого методу діагностики дерматологічних хвороб за їх зображеннями. Враховуючи зростання випадків захворювань шкіри та потребу в точній та швидкій діагностиці, застосування нейромережевих методів може забезпечити швидку та точну діагностику, полегшуючи процес лікування захворювання. Дана тема має великий практичний потенціал, особливо в контексті автоматизації та покращення точності діагностики дерматологічних хвороб.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

За описом предметної області, об'єктом дослідження є процес діагностування дерматологічних хвороб в людини за зображеннями фрагменту шкіри на базі архітектури нейронної мережі. Предметом дослідження є моделі, методи, алгоритми та засоби для автоматизованого діагностування дерматологічних хвороб в людини за їх зображеннями. Метою роботи є розробка нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями та інформаційної системи, яка використовує розроблений метод і дозволяє за зображеннями ураженого фрагменту шкіри людини визначати наявні дерматологічні хвороби за допомогою нейронної мережі. Під час вирішення даного завдання були використані математичні моделі, методи та алгоритми для розв'язання теоретичних і практичних проблем, які виникають при розробці інформаційних технологій. Таким чином, виконана кваліфікаційна робота бакалавра відповідає стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

Під час написання пояснювальної записки та розробки прикладного програмного забезпечення студент продемонстрував вміння та компетентність фахівця комп'ютерних наук. Продемонстрував достатній рівень володіння професійними навичками в спеціальності "Комп'ютерні науки" та ефективно використовував свої знання в розробці програмного забезпечення. Студент проявив значний рівень знань і вмінь у розробці програмного забезпечення. В цілому, Харіш Іван Миколайович успішно впорався з поставленими завданнями своєї кваліфікаційної роботи бакалавра і продемонстрував високу професійну компетентність.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Студент Харин Іван Миколайович усі поставлені завдання своєї кваліфікаційної роботи бакалавра виконував самостійно. Проявив себе як студент, що здатен самостійно вирішувати завдання, знаходити рішення проблем та досягати поставлених цілей.

5. Ступінь оволодіння методами дослідження

Під час виконання кваліфікаційної роботи студент Харин Іван Миколайович продемонстрував достатній рівень компетентності та володіння необхідними інструментами, методами, методиками та технологіями.

6. Повнота та якість розкриття теми роботи

Тема роботи була повністю та якісно розкрита. Проведено аналіз предметної області з досить високою деталізацією та глибиною, що дозволило зрозуміти сутність та важливі аспекти діагностики дерматологічних хвороб за зображеннями.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Матеріал в роботі викладено логічно, послідовно та аргументовано. Літературна грамотність тексту роботи на високому рівні.

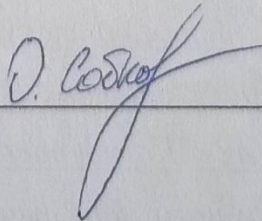
8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Практичне застосування розробленого методу може бути корисним для лікарів-дерматологів та медичних спеціалістів, дозволяючи їм швидше та більш точно встановлювати діагнози, а також розпізнавати різні типи хвороб на ранніх стадіях. Таке застосування нейромережевого методу може покращити якість діагностики та допомогти в розробці ефективних терапевтичних підходів для пацієнтів з дерматологічними захворюваннями.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «добре».

Керівник _____



викладач каф. КН Олена СОБКО



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-19-2 Хариша Івана Миколайовича

за темою: Нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями

1. Актуальність обраної теми

Актуальність теми "Нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями" полягає в потенційній можливості покращення швидкості, точності та ефективності діагностики, що сприяє покращенню медичної практики та забезпеченню кращого лікування пацієнтів з дерматологічними захворюваннями.

2. Повнота розкриття мети та завдань роботи

Мета та завдання роботи за темою "Нейромережевий метод діагностування дерматологічних хвороб за їх зображеннями" розкриті повністю.

3. Зміст кожного розділу роботи

Розділ 1 включає докладний огляд теоретичних та практичних аспектів, пов'язаних з діагностикою шкірних захворювань та аналізом існуючих рішень для їх діагностики. У розділі 2 проводиться огляд нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями. Автор надає схему та кроки нейромережевого методу, наводить архітектури нейронних мереж для його реалізації. Також в цьому розділі спроектовано інформаційну структуру програмного забезпечення. У розділі 3 описано архітектуру системи, вказано основні функціональні можливості та детально їх описано. Наведено ряд схем, що описують роботу нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями. Також в даному розділі проаналізовано ефективність розробленого нейромережевого методу діагностування дерматологічних хвороб за їх зображеннями.

4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблена інформаційна система має велику практичну цінність. Оцінка ефективності цієї системи, що наведена в розділі 3 кваліфікаційної роботи бакалавра засвідчує, що вона може бути використана для швидкого та точного виявлення та класифікації різних типів дерматологічних хвороб на основі зображень шкіри. Завдяки нейромережевому підходу, система може надати об'єктивну та надійну оцінку стану шкіри пацієнтів, забезпечуючи виявлення хвороб та вчасне лікування. Це може значно зменшити час, затрачений на діагностику, та допомогти лікарям приймати кваліфіковані рішення щодо лікування та догляду за пацієнтами.

5. Якість оформлення кваліфікаційної роботи бакалавра

Кваліфікаційна робота студента Хариша Івана Миколаєвича оформлена згідно вимог до структури та нормоконтролю.

6. Недоліки кваліфікаційної роботи бакалавра

Суттєвих недоліків не виявлено. Проте інтерфейс програмної системи має звужену функціональність, а результатом роботи системи є визначення тільки одної хвороби в той час як на зображенні можуть бути ділянки шкіри з кількома одночасно наявними захворюваннями.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка « *добре* ».

Рецензент

Григоренко О.М., доц. кафедри ІТБ 