

УДК 004:4'2

DOI: <https://doi.org/10.53920/ITS-2022-1-1>

**Ірина Анатоліївна ГЕТЬМАН,**

к.т.н., доцент, (доцент кафедри комп'ютерних інформаційних технологій Донбаської державної машинобудівної академії)

ORCID ID: 0000-0003-1835-4256

**Марина Анатоліївна ДЕРЖЕВЕЦЬКА,**

к.е.н. (викладач кафедри медичної фізики та інформаційних технологій Донецького національного медичного університету)

ORCID ID: 0000-0002-9952-4992

**Тетяна Володимирівна БАУЛІНА,**

д-р е.н., професор, (професор кафедри менеджменту Міжнародного науково-технічного університету

ім. академіка Ю. Бугая)

ORCID ID: 0000-0002-0110-849X

**Тамара Василівна КУХТІК,**

д-р т.н., професор, (професор кафедри комп'ютерних наук Міжнародного науково-технічного університету

ім. академіка Ю. Бугая) ORCID ID: 0000-0002-2000-3669

**Тетяна Юріївна СОЛОМКО,**

к.т.н., доцент, (доцент кафедри цифрових технологій, менеджменту та адміністративного управління Донбаського інституту техніки та менеджменту)

ORCID ID: 0000-0002-3029-7920

## **ПРОЕКТ ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ РЕАЛІЗАЦІЇ ДОДАТКУ ДЛЯ РОЗПІЗНАВАННЯ ЛІКАРСЬКИХ РОСЛИН**

***Розроблено програмний продукт призначений для поліпшення пошуку необхідних рослин та зменшення непередбачуваних випадків при використанні неправильних лікарських рослин. Зроблені аналізи існуючих аналогів, розглянути їх недоліки. За допомогою експертного оцінювання було обрано мову програмування, програмний продукт та нейронну мережу. Спроектвана базова архітектура та діяльність власного продукту. За допомогою методів проектування SADT було спроектоване базовий принцип роботи системи, який містить ідентифікацію лікарських рослин по зображенню. Структурована діаграма прецедентів, яка відображає систему ідентифікації лікарських рослин по зображенню. Побудовано діаграму послідовностей, яка містить систе-***

**му ідентифікації лікарських рослини по зображенню. Для побудови нейронної мережі використано 60 видів рослин – 38815 зображення, що надає чітку ідентифікацію, яка захистить людину від небезпечних помилок при використанні лікувальних рослин.**

**У даній роботі розглянуті такі завдання, як ідифікація і розпізнавання образів, в конкретному випадку образів рослин по фотографії. Створення архітектури на основі InceptionV3, для створення моделі переднавченої нейронної мережі. Класифікатор зображень рослин на основі переднавченої нейронної мережі. Зображення в навчальній мережі було поділено на категорії в залежності від частини рослини, зображеного на них: Entire (рослина цілком), Branch (гілка), Flower (квітка), Fruit (фрукт, або ягода), LeafScan (скан листа), Leaf (листя), Stem (стебло). Для кожної з цих категорій підібрано свій найбільш відповідний метод попередньої обробки. Навчено мережу в цілому за використанням аугментації та бібліотеки Imgaug. Для аугментації обиралися ті перетворення, які відбуваються в реальному житті. Використано топ-метрики для вимірювання здатності моделі видавати справжній клас рослини в списку найбільш вірогідних класів.**

**Ключові слова:** лікувальні рослини, ідентифікація, генератор зображень, нормалізація, нейрона мережа, розпізнавання образів.

**Iryna GETMAN,**

Candidate of Technical Sciences, Docent

**Maryna DERZHEVETSKA,**

Candidate of Economic Sciences

**Tetyana BAULINA,**

Doctor of of Economic Sciences, Professor

**Tamara KUKHTYK,**

Doctor of of Technical Sciences, Professor

**Tetiana SOLOMKO,**

Candidate of Technical Sciences, Docent

## **PROJECT OF A SOFTWARE COMPLEX FOR THE IMPLEMENTATION OF AN APPLICATION FOR THE IDENTIFICATION OF MEDICINAL PLANTS**

**A software product has been developed designed to improve the search for necessary plants and reduce unforeseen cases when using the wrong medicinal plants. Analyzes of existing analogues have been**

**made, and their shortcomings have been considered. A programming language, a software product, and a neural network were selected with the help of expert evaluation. The basic architecture and activities of the own product have been designed. With the help of SADT design methods, the basic principle of the system was designed, which includes the identification of medicinal plants by image. Structured chart of precedents, which reflects the system of identification of medicinal plants by image. A sequence diagram has been constructed, which contains a system for identifying medicinal plants by image. To build a neural network, 60 types of plants were used - 38,815 images, which provide a clear identification that will protect people from dangerous mistakes when using medicinal plants. In this work, such tasks as identification and recognition of images, in the specific case of images of plants from photographs, are considered. Creation of an architecture based on InceptionV3, to create a model of a pre-trained neural network. A plant image classifier based on a pre-trained neural network. The images in the learning network were divided into categories depending on the part of the plant depicted on them: Entire (plant as a whole), Branch, Flower, Fruit (fruit or berry), LeafScan, Leaf, Stem. For each of these categories, the most suitable method of pre-processing has been selected. Trained the network as a whole using augmentation and the Imgaug library. Those transformations that occur in real life were chosen for augmentation. Top-metrics were used to measure the ability of the model to give the true class of the plant in the list of the most probable classes.**

**Keywords:** medicinal plants, identification, image generator, normalization, neural network, pattern recognition.

**Постановка проблеми.** За все своє існування, людство вивчало навколишній світ. І перше, що потрапило людині на очі – різноманітні рослини. Їх різновид – колір, розмір і структура були безмежні. Поступово людина почала куштувати рослини, це могло проявитися, як в гарному так і поганому ключі. Велика кількість людей помирала від отрути, або неправильного використання тих або інших рослин. Надалі люди вивчали флору, запам'ятовували та передавали знання із покоління в покоління, але це не давало 100% результату, що хтось не отруїться або не сплутає схожі між собою рослини. Людство розвивалося і зробило безліч записів, відкрило майже всі існуючі рослини, вивчило не одне покоління лікарів, але навіть і зараз деякі люди через неувважність, або дитя-

чу цікавість та необережність куштують або використовують і надалі отруєні або неправильні лікарські рослини в лікуванні певних захворювань або поранень.

Також зараз людство перейшло на новий рівень лікування, і майже ніколи, на пряму, не застосовує лікарські рослини, що підвищує витрати, бо сучасні ліки виробляються в лабораторіях за допомогою передових технологій. Їх доступність і надійність вища ніж у рослин, але нажаль не кожен може витратити стільки коштів. Також є захворювання, які не під силу сучасній медицині, або можуть визивати сильні побочні ефекти через вміст отруйних речовин. Лікарські рослини навпаки, вони не містять ні металів, ні неприродних речовин, що робить їх більш безпечними у правильному використанні.

Тому в наслідок останніх подій почали з'являтися різні ідентифікатори і розпізнавачі рослин. Більшість з них присутні на системі Android і невелика кількість на РС. Але навіть така кількість програм, не призводить до найкращого результату: більшість з них або погано працюють, або видають неточну інформацію, яка може нашкодити здоров'ю. Навіть пошукова система Google видає лише частково правильні відповіді при пошуку певної рослини. І найголовнішою помилкою є те, що такі програмні забезпечення дають можливість користувачам вносити свої неправильні та некоректні данні, які йдуть в нейронну мережу і формують помилки, які нажаль у великій кількості зображень – або довго шукають, або навіть залишають все як є.

**Метою роботи** є розробка мультиплатформеного програмного комплексу, який за використанням нейронної мережи виконує розпізнання лікувальних рослин з метою допомоги людям і їх огороження від отруйних і не бажаних у використанні або вживанні рослин.

**Виклад основного матеріалу дослідження.** Головною частиною даного ідентифікатора є його набір зображень лікарських рослин. Після їх зібрання необхідно представити їх в такому виді, щоб ідентифікатор міг порівняти отримане зображення з тими що вже мають. Таким чином потрібно зробити нейронну мережу на основі розповсюдженої бібліотеки TensorFlow.

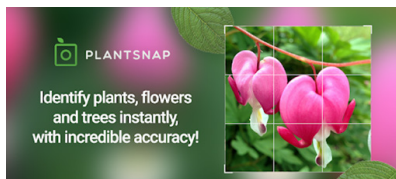
TensorFlow — відкрита програмна бібліотека для машинного навчання цілої низки задач, розроблена компанією Google для задоволення її потреб у системах, здатних будувати та тренувати нейронні мережі для виявлення та розшифрування образів та кореляцій, аналогічно до навчання й розуміння, які застосовують

люди [1]. Її наразі застосовують як для досліджень, так і для розробки продуктів Google.

Аугментація даних (data augmentation) – це методика створення додаткових навчальних даних з наявних даних. Для досягнення хороших результатів глибокі мережі повинні навчатися на дуже великому обсязі даних. Отже, якщо вихідний навчальний набір містить обмежену кількість зображень, необхідно виконати аугментацію, щоб поліпшити результати моделі [2]. На сьогоднішній день існує достатня кількість працюючих **ідентифікаторів рослин**. Розглянемо деякі з них.

Першим розглянемо непогану програму для Android «PlantSnap» (рис. 1) [3]. Цей додаток для ідентифікації рослин від Earth.com, який допомагає миттєво визначити будь-які рослини. У нього є карта місцевих рослин для допомоги в дослідженнях. Він працює на основі технології глибокого навчання і штучного інтелекту; це означає, що додаток дійсно поповнюється і вдосконалюється з кожним завантаженим зображенням. Він англомовний та платний.

Наступною програмою-ідентифікатором рослин, можна вважати «Flora Incognita» (рис. 2) [4]. Вона дозволяє швидко, легко і точно ідентифікувати рослини. Однак це тільки рослини Центральної Європи.



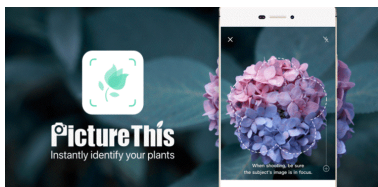
**Рис. 1. PlantSnap**

Джерело: [3]



**Рис. 2. Flora Incognita**

Джерело: [4]



**Рис. 3. PictureThis**

Джерело: [5]

PictureThis - захоплюючий додаток для ідентифікації рослин (рис. 3) [5]. Він потребує відправки фотографії рослини для мит-

тевого і точного його визначення. Додаток об'єднує технології візуального розпізнавання і глибокого навчання, а так само використовує досвід і знання всієї мережі фахівців з садівництва і ботаніків. Однак він не може поповнюватися новими зображеннями і потребує часу для отримання результату ідентифікації.

Аналіз існуючих додатків-ідентифікаторів показав, що більшість безкоштовних додатків має невелику базу рослин, яку неможливо поповнити. Ті які можна поповнити не передбачені для використання на телефоні. При чому у даних ідентифікаторів велика помилка розпізнавань. Тому в нашій моделі є можливість подавати зображення, яке завантажуються з інтернету або фотокамери, а також на Android за допомогою камери телефону. На виході модель ідентифікації видає результат порівняння рослин, тобто найбільш можливий результат у відсотках. Користувач може самостійно передавати зображення для класифікації (сканування) для отримання результату.

Для використання завдання роботи треба вибрати програмну середу для його реалізації, тому необхідно провести їх експертну оцінку. Експертна оцінка визначається за формулою (1):

$$Q_i = \sum_{j=1}^N K_j * a_{ij}, i = \overline{1,3}, n = 7 \quad (1)$$

де  $Q_i$  – загальна оцінка програмного продукту;  $n$  – кількість критеріїв;  $K_j$  – ваговий коефіцієнт критерію;  $a_{ij}$  – оцінка  $i$  – того програмного засобу по  $j$  – тим критерієм.

Порівняння програмних продуктів (мови програмування) (за десятибальною шкалою) наведено у таблиці 1 [6–9].

**Таблиця 1. Порівняння мови програмування**

Найменування критерію	Python	C++	Java	$K_j$
Динамічна типізація	9	8	7	0,10
Підтримка нейронної мережі	9	6	5	0,10
Інтерпретатор	7	6	9	0,20
Багатоплатформеність	9	8	5	0,15
Підтримка ООП	7	9	9	0,20
Зручність підключення сторонніх модулів	8	8	7	0,10
Простота застосування	9	7	7	0,15

Як показав розрахунок, найбільш доцільно вибирати в якості мови програмування Python.

Порівняння засобів розробки (за десятибальною шкалою) наведено у таблиці 2 [10-11].

**Таблиця 2. Порівняння програмних продуктів (засоби розробки)**

Найменування критерію	Android Studio	Bazel	Appery.io	$K_j$
Динамічна типізація	8	7	6	0,10
Підтримка нейронної мережі	7	8	5	0,10
Інтерпретатор	9	7	8	0,20
Багатоплатформеність	8	7	6	0,15
Підтримка ООП	7	8	9	0,20
Зручність підключення сторонніх модулів	6	6	7	0,10
Простота застосування	9	8	6	0,15

Як показав розрахунок, найбільш доцільно вибирати в якості засобу розробки програмний продукт Android Studio.

Порівняння нейронних мереж (за десятибальною шкалою) наведено у таблиці 3 [12-14].

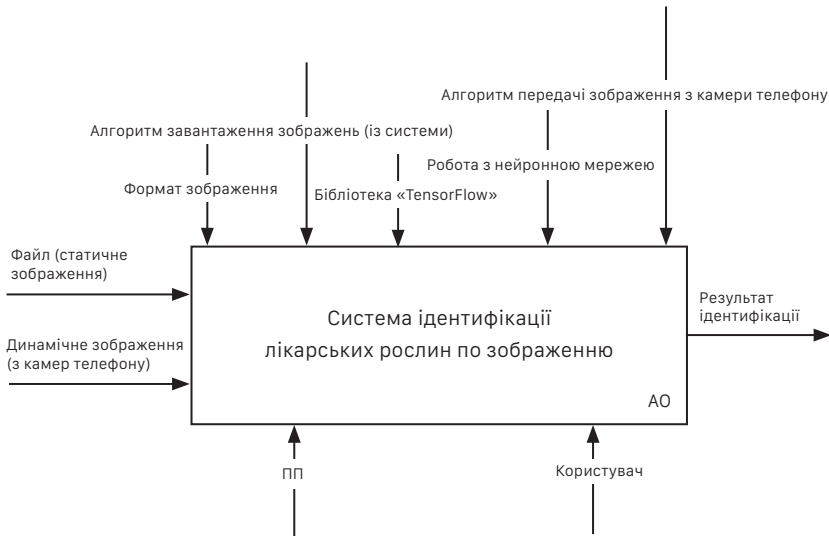
**Таблиця 3. Порівняння нейронних мереж**

Найменування критерію	TensorFlow	Keras	Torch	$K_j$
Підтримка реляційної моделі даних	7	9	6	0,2
Технологія створення нейронної мережі і об'єктів нейронної мережі	8	7	5	0,2
Можливість створення локальної нейронної мережі	9	8	7	0,1
Наявність вбудованої мови для розробки додатків	7	7	5	0,15
Підтримка стандарту SQL	7	6	9	0,1
Засоби для отримання звітів	7	8	3	0,1
Простота застосування	9	7	4	0,15

Як показав розрахунок, найбільш доцільно вибирати в якості нейронної мережі TensorFlow.

Отже, для реалізації ПК та Android «Система ідентифікації лікарських рослини по зображенню» можна використовувати програмні засоби Python, Android Studio и TensorFlow.

На рисунку 4 зображена контекстна SADT-діаграма нульового рівня для ідентифікатора в середовищі ПК та Android.



**Рис. 4. Контекстна SADT-діаграма нульового рівня для процесу AO «Система ідентифікації лікарських рослини по зображенню»**

Джерело: Розроблено авторами

На основі аналізу предметної області та виявлених вимог щодо розроблюваного програмного продукту розроблена діаграма прецедентів використання (рис. 5) та загальна схема використання ідентифікатора лікувальних рослин. Користувач запускає ПП та фокусує або завантажує зображення, в залежності від того запуск виконується на ПК або Android. Потім ПП відправляє зображення на класифікацію в ході якої надалі передаються данні для порівняння їх з нейронною мережею. Користувач має доступ до прецеденту: «Введення зображення», а Розробник до прецедентів «Робота нейронної мережі», «Сканування (Обробка сфотографованого зображення)», «Отримання результату (назви рослини і його лікувальних властивостей)».



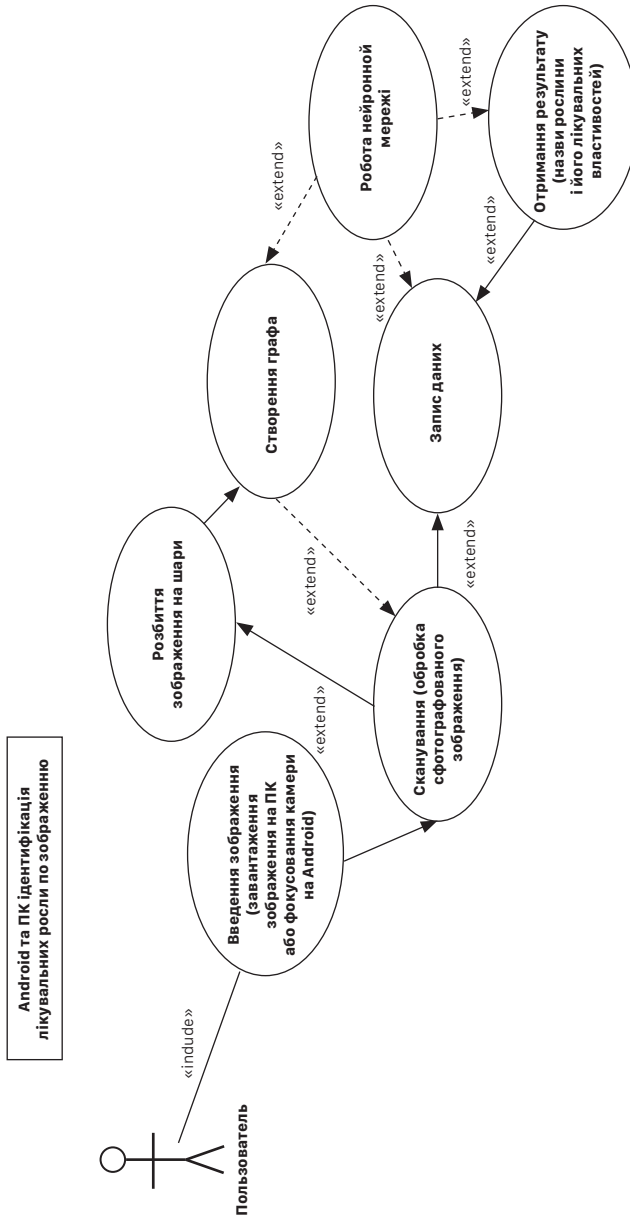


Рис. 5. Діаграма прецедентів ПК і Android «Система ідентифікації лікарських рослин по зображенню»

Джерело: Розроблено авторами

Діаграма послідовності для ПК та Android «Система ідентифікації лікарських рослини по зображенню» представлена на рисунку 6. Користувач робить вхід в клас «ПП», попадає в програму, де він повинен завантажити зображення на ПК, або сфокусувати зображення камерою телефону. Клас «Система завантаження зображення», або «CameraActivity: Система передачі зображення (камера)» вводить клас «Зображення», який передає інформацію о зображенні в динамічну пам'ять. Потім данні зображення потрапляють в клас «Класифікатор (сканер)» який розпізнає введені данні і на їх основі будує слої зображення, а потім граф, який записується в динамічну пам'ять. Надалі записаний граф знаходить до класу «Нейрона мережа», який порівнює існуючі графи зображень з введеним графом. Після того як користувач розташував усі вежі, починаються хвилі противників та їх взаємодія з вежами. Тобто йде звертання до класу «Противник». Останній клас «Ідентифікація зображення» робить аналіз на основі порівняння нейронної мережі і введеного в неї графа, і виводить у відсотках схожість введеного зображення. Таким чином користувач отримує інформацію о лікувальній рослині, яку сфотографував або знайшов в інтернеті, а також її застосування в лікуванні певних захворювання або травм.

Комп'ютерна реалізація розробленого ПК полягає в тому, щоб передбачити таксономічний клас рослини, рунтуючись на декількох його фотографіях.

Для навчання нейронної мережі використовувалося 38815 зображень рослин, кожне з яких належить до одного з класів. Необхідно було побудувати модель, яка буде повертати список найбільш ймовірних класів рослин. Позиція вірного класу рослин у списку передбачених класів (ранг) визначає якість системи.

Це завдання моделює реальний життєвий сценарій, де людина намагається ідентифікувати рослину, вивчаючи його окремі частини (стебло, лист, квітка та ін.)

Модель на вході отримує «спостереження» — одну найкращу фотографію, зроблену без зайвих об'єктів [15].

В якості базових архітектур були обрані VGG16, ResNet50 і InceptionV3. Ці мережі були натреновані на величезній кількості зображень і вже вміють розпізнавати найпростіші об'єкти, тому можна сподіватися, що вони допоможуть створити гідну модель для класифікації рослин.

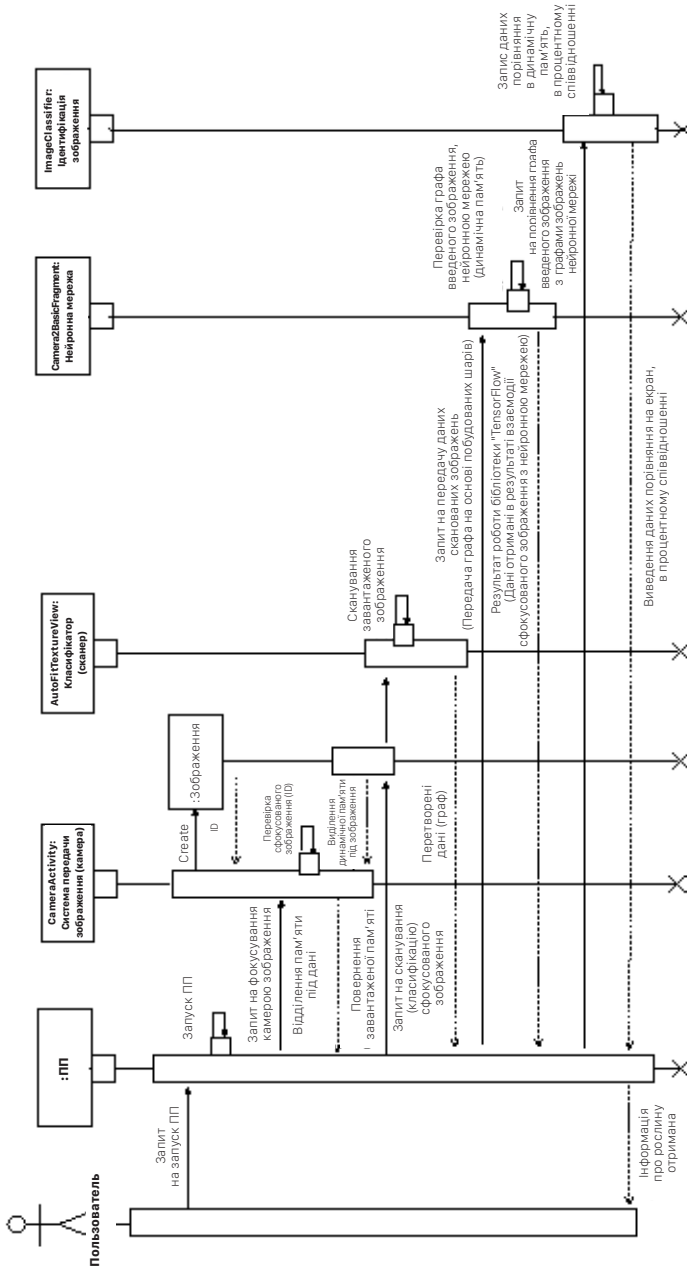


Рис. 6. Діаграма послідовностей ПК «Система ідентифікації лікарських рослини по зображенню»

Джерело: Розроблено авторами

Використовувалася модель Inception-v3, яку Google вже навчив на тисячі класів [16]. Також використовувалася transfer learning, який допомагає перенавчити фінальний шар вже навченої моделі Inception-v3 новим категоріям з нуля. Цей процес займає близько 30 хвилин і не вимагає будь-яких графічних процесорів. Навчання моделі на відкритій базі даних кольорів, яка є в TensorFlow і безкоштовна для скачування. Також використовувався Python 3 і TensorFlow 1.4 [1].

Tensorflow формуємо базу на основі репозиторію TensorFlow, потім створюємо скрипти для пошуку необхідної рослини. Для цього необхідна нейрона мережа лікарських рослин, яку за допомогою інтернету треба сформувати, і чим більше, тим краще.

Далі завантаживши базу лікарських рослин, в якій міститься 17 категорій: для горла; для очей; від жіночих захворювань; від інфекцій; для крові; для легких; від чоловічих захворювань; для зовнішніх травм; від захворювання нервів; для носа; для травлення; для нирок і сечового міхура; від психічних захворювань; для серця; для судин; для суглобів; для вуха.

Головна задача полягала в тому, щоб настроїти вже написану нейронну мережу, таким чином щоб вона здобула найкращу предобученість і могла ідентифікувати зображення без помилок.

Найважливішим етапом при реалізації програмного комплексу є препроцесинг зображень - попередня обробка зображень. Основною метою предобробки, в нашому випадку, є виявлення найбільш важливої частини зображення і видаленні непотрібного шуму.

Всі зображення в навчальній мережі можна розділити на категорії в залежності від частини рослини, зображеного на них: Entire (рослина цілком), Branch (гілка), Flower (квітка), Fruit (фрукт, або ягода), LeafScan (скан листа), Leaf (листя), Stem (стебло). Для кожної з цих категорій був підібраний свій найбільш відповідний метод попередньої обробки [17].

При обробки Entire і Branch зображень не треба їх змінювати, оскільки найчастіше велика частина зображення містить корисну інформацію, яку не треба втрачати (рис. 7,8).



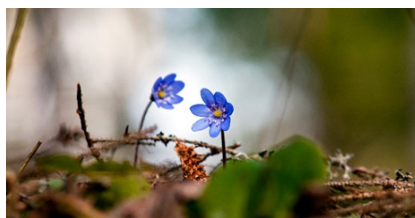
**Рис. 7. Зображення рослини цілком, з усіма її елементами**



**Рис. 8. Обробка певної частини рослини – гілка**

Джерело: Розроблено авторами

Для обробки Flower і Fruit зображень використовується один і той же метод: конвертуємо зображення в чорно-біле; застосуємо фільтр Гаусса з параметром  $\alpha = 2.5$ ; використовуємо метод активних контурів для пошуку найбільш важливої частини зображення; описуємо прямокутник навколо кордону (рис. 9, 10).



**Рис. 9. Обробка певної частини рослини – квітка**

Джерело: Розроблено авторами



**Рис. 10. Обробка певної частини рослини – фрукт, або ягода**

Джерело: Розроблено авторами

Переглянувши фотографії LeafScan, можна помітити, що в більшості випадків лист знаходиться на світлому фоні. Нормування зображення білим кольором: спершу конвертуємо зображення в чорно-біле і застосовуємо Otsu-метод, щоб розрахувати граничне значення; всі пікселі, значення яких менше порогового значення, фарбуємо в білий колір (рис. 11).



**Рис. 11. Сканування листка рослини**

Джерело: Розроблено авторами

Зазвичай в Leaf зображеннях лист знаходиться в центрі, а його контур трохи відступає від країв зображення. Для препроцесинга таких фото запровадяться наступні методи: вирізаємо по 1/10 зображення зліва, справа, знизу і зверху; конвертуємо зображення в чорно-біле; застосовуємо фільтр Гаусса з параметром  $\alpha = 2$ ; використовуємо метод активних контурів, щоб порахувати кордон найбільш важливої області; описуємо прямокутник навколо отриманої межі (рис. 12).



**Рис. 12. Обробка певної частини рослини – листок**

Джерело: Розроблено авторами

Обробка Stem зображень наступна. Стебло зазвичай знаходиться в центрі зображення: видаляємо по полотнище частини зображення зліва, справа, знизу і зверху; конвертуємо зображення в чорно-біле; застосовуємо фільтр Гаусса з параметром  $\sigma = 2$ ; використовуємо метод активних контурів, щоб порахувати кордон найбільш важливої області зображення; описуємо прямокутник навколо отриманої межі (рис. 13).



**Рис. 13. Обробка певної частини рослини – стебло**

Джерело: Розроблено авторами

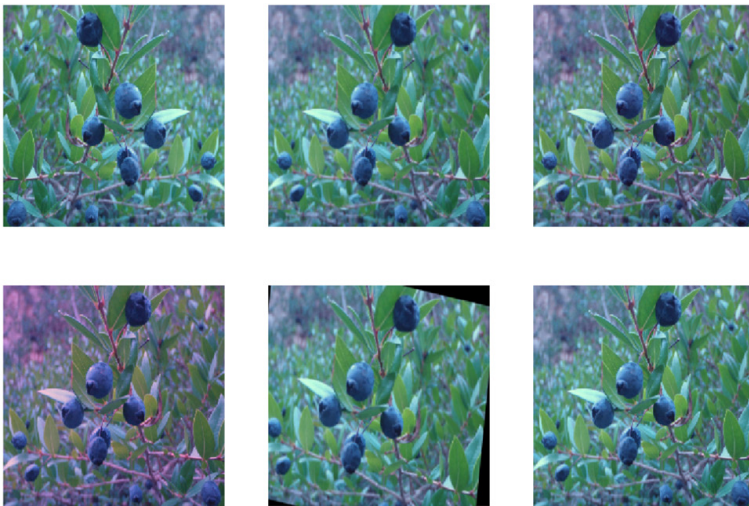
На заключному кроці навчаємо мережу цілком. Тут можемо використовувати аугментацію. Але замість стандартного ImageDataGenerator з Keras використовується Imgaug — бібліотека, яка призначена для аугментування зображень. Важливою особливістю Imgaug є те, що можна явно вказати, з якою ймовірністю перетворення має бути застосоване до зображення. Крім того, в цій бібліотеці є велика різноманітність перетворень, є можливість поєднувати перетворення в групи, і вибирати, яку з груп застосувати. Приклади можна знайти за посиланням вище [18].

Для аугментації вибираємо ті перетворення, які можуть відбуватися в реальному житті, наприклад, дзеркальне відображення фото (по горизонталі), повороти, збільшення, зашумлення, зміна яскравості і контрастності.

Розіб'ємо перетворення на кілька груп і застосовуватимемо кожну з них із заданою ймовірністю (у кожного може бути різна ймовірність). Потрібно аугментувати зображення у 80% випадках, тоді мережа зможе побачити і реальне зображення. З огляду на те, що навчання займає кількох десятків епох, є дуже великий шанс, що мережа побачить кожне зображення в оригіналі.

Ще один спосіб, який допоміг збільшити якість класифікатора передбачення на аугментованих даних (test-time augmentation, ТТА). Цей спосіб полягає в тому, щоб робити передбачення не тільки для зображень в тестовому наборі, але і для їх аугментації [19].

Наприклад, візьмемо п'ять найбільш реалістичних перетворень, застосуємо їх до зображень і отримаємо передбачення вже не для однієї картинки, а для шести. Після цього усереднім отриманий результат. Всі аугментовані зображення отримані в результаті одного перетворення (одне зображення — одне перетворення) (рис. 14).



**Рис. 14. Приклад аугментації на пророкуванні**

Джерело: Розроблено авторами

**Висновки та пропозиції.** Використано 4 метрики: основну метрику, запропоновану авторами, а також 3 топ-метрики — Top 1, Top 3, Top 5. Топ-метрики, також як і основна, застосовуються до спостереження (набору фотографій з однаковим Observation Id), а не окремого зображення.

У процесі роботи об'єднано результати декількох моделей, щоб ще більше покращити якість класифікатора (всі моделі бралися з однаковою вагою). Останні три рядки в таблиці, показують кращі результати, отримані при об'єднанні моделей (табл. 4).



**Таблиця 4. Порівняння ефективності моделей**

Мо- дель	Мережа	Цільова метрика (rank)	Тop 1	Тop 3	Тop 5	Епохи	ТТА
1	VGG16	0.549490	0.454194	0.610442	0.665546	49	Ні
2	VGG16	0.553820	0.458732	0.612600	0.666996	49	Так
3	VGG19	0.559978	0.468980	0.620219	0.676396	62	Ні
4	VGG19	0.563019	0.470534	0.619303	0.682585	62	Так
5	ResNEt50	0.573424	0.489943	0.627836	0.688938	46	Ні
6	ResNEt50	0.581954	0.495962	0.638806	0.716630	46	Так
7	Inception V3	0.528063	0.495962	0.666928	0.723992	38	Ні
8	Inception V3	0.615734	0.535675	0.671392	0.721084	38	Так

Модель переможців змагання показала результат 0.471 по цільовій метриці. Вона являє собою поєднання статистичних методів і нейронної мережі, навченої лише на тих зображеннях рослин, які були надані організаторами.

Дана модель, яка в якості основи використовує перед навчену нейронну мережу InceptionV3, досягає результату 0.60785 по цільовій метриці, покращуючи результат переможців конкурсу на 29%.

При використанні аугментації на тестових даних, результат по цільовій метриці збільшується до 0.615734, але в той же час швидкість роботи моделі падає приблизно в 6 разів [18].

Можливо піти ще далі і об'єднати результати роботи декількох мереж. Такий підхід дозволяє домогтися результату 0.635100 за цільовою метриці, але при цьому швидкість дуже сильно падає, і в реальному житті така модель може знайти застосування лише там, де швидкість роботи не є ключовим чинником, наприклад, в різних дослідженнях в лабораторіях.

Існуючі моделі не завжди можуть вірно визначити клас рослина, в цьому корисно знати список з найбільш ймовірних видів рослини. Для того щоб виміряти здатність моделі видавати справжній клас рослини в списку найбільш вірогідних класів, використовується top-метрики. Наприклад, за метрикою Top 5 довчена мережа InceptionV3 показала результат 0.716630. Якщо ж об'єднати кілька моделей і застосувати ТТА, то можна поліпшити результат до 0.730051 [16].

## ЛІТЕРАТУРА

1. Hello, TensorFlow! [Електронний ресурс]: – Режим доступу: <https://www.oreilly.com/content/hello-tensorflow/>.
2. Аугментация данных [Електронний ресурс]: – Режим доступу: <https://habr.com/ru/company/smartengines/blog/264677/>.
3. Flora Incognita [Електронний ресурс]: – Режим доступу: <https://floraincognita.com/>.
4. PictureThis [Електронний ресурс]: – Режим доступу: <https://www.picturethisai.com/>.
5. Ruder S. (2016) An overview of gradient descent optimisation algorithms <https://doi.org/10.48550/arXiv.1609.04747>.
6. The Making of Python [Електронний ресурс]: – Режим доступу: <https://www.artima.com/intv/pythonP.html>.
7. Python 3.0 Release [Електронний ресурс]: – Режим доступу: <https://www.python.org/download/releases/3.0/>.
8. Android Studio Overview [Електронний ресурс]: – Режим доступу: <http://developer.android.com/tools/studio/index.html>.
9. Кухтик Т. В., Гетьман І. А., Держевецька М. А. Дослідження методів, моделей та інформаційних технологій експертного оцінювання результатів технологічного процесу. –Збірник наукових праць Національного університету кораблебудування імені адмірала Макарова 3(481). 2020. – С.85-90.
10. Download Android Studio and SDK Tools [Електронний ресурс]: – Режим доступу: <https://developer.android.com/studio>.
11. Credits [Електронний ресурс]: – Режим доступу: <https://www.tensorflow.org/about>.
12. Google Offers Free Software in Bid to Gain an Edge in Machine Learning [Електронний ресурс]: – Режим доступу: <https://bits.blogs.nytimes.com/2015/11/09/google-offers-free-software-in-bid-to-gain-an-edge-in-machine-learning/>.
13. Large-Scale Machine Learning on Heterogeneous Distributed Systems [Електронний ресурс]: – Режим доступу: <http://download.tensorflow.org/paper/whitepaper2015.pdf>.
14. Програма генерации тестовых изображений для программных комплексов обработки снимков металлографического анализа /Л.В. Васильева, И.А. Гетьман, С.К. Добряк// Наук. пр. Донецького національного технічного університету. Серія: Обчислювальна техніка та автоматизація, № 1(29) – Покровськ (Красноармійськ): ДонНТУ, 2016. – С. 39–46. – ISSN 2075-4272. [Електронний ресурс]: –

Режим доступу: [https://science.donntu.edu.ua/wp-content/uploads/2017/06/%D0%9E%D0%A2%D0%90\\_129\\_2016.pdf](https://science.donntu.edu.ua/wp-content/uploads/2017/06/%D0%9E%D0%A2%D0%90_129_2016.pdf).

15. Оцінка зображення [Електронний ресурс]: – Режим доступу: <https://www.imageclef.org/2014/lifeclef/plant>.

16. Inception-v3 модель [Електронний ресурс] ]: – Режим доступу: <https://habr.com/ru/post/302242/>.

17. Попередня обробка зображень [Електронний ресурс]: – Режим доступу: [https://ru.bmstu.wiki/предварительная\\_обработка\\_изображений](https://ru.bmstu.wiki/предварительная_обработка_изображений).

18. Tensorflow [Електронний ресурс] ]: – Режим доступу: <https://habr.com/ru/post/305578/>.

19. Класифікатор на основі нейронної мережі [Електронний ресурс]: – Режим доступу: <http://mechanoid.kiev.ua/neural-net-art1.html>.

20. Л.В. Васильева, А.Ф. Тарасов, И.А. Гетьман. Разработка алгоритмического и программного обеспечения сегментации изображений / Вимірювальна та обчислювальна техніка в технологічних процесах. – Хмельницький, 2016. № 3 (56) – С. 117–122. [http://journals.khnu.km.ua/vottp/pdf/pdf\\_full/2016/vottp-2016-3.pdf](http://journals.khnu.km.ua/vottp/pdf/pdf_full/2016/vottp-2016-3.pdf).

## REFERENCES

1. „Hello, TensorFlow!“, available at: <https://www.oreilly.com/content/hello-tensorflow/> (Accessed 20 July 2022).

2. „Auhmentatsiia danykh“ [Data augmentation], available at: <https://habr.com/ru/company/smartengines/blog/264677/> (Accessed 20 July 2022).

3. „Flora Incognita“, available at: <https://floraincognita.com/> (Accessed 21 July 2022).

4. „PictureThis“, available at: <https://www.picturethisai.com/> (Accessed 18 July 2022).

5. Ruder, S., (2016) „An overview of gradient descent optimisation algorithms“, available at: <https://doi.org/10.48550/arXiv.1609.04747> (Accessed 20 July 2022).

6. „The Making of Python“, available at: <https://www.artima.com/intv/pythonP.html> (Accessed 20 July 2022).

7. „Python 3.0 Release“, available at: <https://www.python.org/download/releases/3.0/> (Accessed 22 July 2022).

8. „Android Studio Overview“, available at: <http://developer.android.com/tools/studio/index.html> (Accessed 22 July 2022).

9. Kukhtyk, T., Getman I., Derzhetska M., (2020), „Doslidzhennia metodiv, modelei ta informatsiinykh tekhnolohii ekspertnoho otsiniu-

vannia rezultativ tekhnolohichnoho protsesu" [Research of methods, models and information technologies of expert evaluation of technological process results].

10. „Download Android Studio and SDK Tools“, available at: <https://developer.android.com/studio> (Accessed 22 July 2022).

11. „Credits“, available at: <https://www.tensorflow.org/about> (Accessed 20 July 2022).

12. „Google Offers Free Software in Bid to Gain an Edge in Machine Learning“, available at: <https://bits.blogs.nytimes.com/2015/11/09/google-offers-free-software-in-bid-to-gain-an-edge-in-machine-learning/> (Accessed 19 July 2022).

13. „Large-Scale Machine Learning on Heterogeneous Distributed Systems“, available at: (Accessed 22 July 2022).

14. Vasilyeva, L., Getman, I., Dobriak, S., (2016), „Prohramma heneratsyy testovykh yzobrazheniy dlia prohrammnykh kompleksov obrabotky snymkov metallohrayficheskoho analiza“ [The program for generating test images for software complexes for processing images of metallographic analysis], available at: [https://science.donntu.edu.ua/wp-content/uploads/2017/06/%D0%9E%D0%A2%D0%90\\_129\\_2016.pdf](https://science.donntu.edu.ua/wp-content/uploads/2017/06/%D0%9E%D0%A2%D0%90_129_2016.pdf) (Accessed 23 July 2022).

15. „Otsinka zobrazhennia“ [Image evaluation], available at: <https://www.imageclef.org/2014/lifeclef/plant> (Accessed 23 July 2022).

16. „Inception-v3 model“, available at: <https://habr.com/ru/post/302242/>.

17. „Poperednia obrobka zobrazhen“ [Pre-processing of images], available at: [https://ru.bmstu.wiki/предварительная\\_обработка\\_изображений](https://ru.bmstu.wiki/предварительная_обработка_изображений) (Accessed 22 July 2022).

18. „Tensorflow“, available at: <https://habr.com/ru/post/305578/> (Accessed 21 July 2022).

19. „Poperednia obrobka zobrazhen“ [A classifier based on a neural network], available at: <http://mechanoid.kiev.ua/neural-net-art1.html> (Accessed 20 July 2022).

20. Vasilyeva, L., Tarasov, A., Getman, I., (2016), „Razrabotka alhorytmicheskoho y prohrammnoho obespecheniya sehmentatsyy yzobrazheniy“ [Development of algorithmic and image segmentation software], available at: [http://journals.khnu.km.ua/vottp/pdf/pdf\\_full/2016/vottp-2016-3.pdf](http://journals.khnu.km.ua/vottp/pdf/pdf_full/2016/vottp-2016-3.pdf) (Accessed 20 July 2022).

**СТАТТЯ НАДІЙШЛА ДО РЕДАКЦІЇ 25.03.2022**