МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ЗІНЕЦЬ ВЛАДИСЛАВ ВОЛОДИМИРОВИЧ

Допускається до захисту:
завідувач кафедри біофізики
і фізіології кандидат хім. наук, доцент
_____ О. І. Доценко
« ____»_____ 2024 р.

СЕМАНТИЧНА СЕГМЕНТАЦІЯ МАЛЮНКУ ЖИЛКУВАННЯ ЛИСТЯ CARPINUS BETULUS НА БАЗІ UNET ПОДІБНИХ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

Спеціальність 091 Біологія та біохімія

Кваліфікаційна (магістерська) робота

Керівник:
Міщенко А.М., к.б.н., старший викладач
кафедри біофізики та фізіології

Оцінка: ____/____/_____
(бал/за шкалою ЄКТС/за національною шкалою)

Голова ЕК:_____

Вінниця 2024

Анотація

Зінець В.В. Семантична сегментація малюнку жилкування листя Carpinus betulus на базі UNet подібних згорткових нейронних мереж. Кваліфікаційна робота магістра. Спеціальність 091 Біологія та біохімія. Донецький національний університет імені Василя Стуса, Вінниця, 2024.

Розробка методів автоматичної сегментації мережі жилок листків має важливе значення для ботаніки, сільського господарства. В магістерській роботі досліджуються можливості використаня U-Net подібних згорткових нейронних мереж для семантичної сегментації малюнку жилкування листя Carpinus betulus. В роботі було створено розмічений набір даних зображень листків Carpinus betulus для навчання нейромережі. Мовою Python з використанням платформим GoogleColab була реалізована нейромережа U-Net, було здійснено її навчання та оцінка якості. Було досліджено вплив на якість навчання: розміру та способів аугментації набору даних; вибору виду функцій втрат; архітектури нейромережі; кількості та типів класів сегментації. Підібрано оптимальні параметри нейромережі, що дозволяють заощадити обчислювальні ресурси не зменшуючи якість навчання. Показані проблеми, що виникають при класифікації великої кількості класів, запропоновані два сценарії практичного застосування нейромережі для сегментації окремих жилок листя граба.

Ключові слова: нейромережа, машинне навчання, сегментація, глибоке навчання, функції втрат, валідаційний набір, тренувальний набір, аугментація

Abstract

Zinets V.V. Semantic Segmentation of Leaf Venation Pattern of Carpinus betulus Based on U-Net-like Convolutional Neural Networks. Master's Thesis. Specialty 091 Biology and Biochemistry. Vasyl Stus Donetsk National University, Vinnytsia, 2024.

The development of methods for automatic segmentation of leaf venation patterns is of significant importance for botany and agriculture. This master's thesis explores the potential of using U-Net-like convolutional neural networks for the semantic segmentation of the leaf venation pattern of Carpinus betulus. A labeled dataset of Carpinus betulus leaf images was created for training the neural network. The U-Net neural network was implemented using Python and the Google Colab platform, with training and quality assessment carried out. The impact of various factors on training quality was studied, including the size and augmentation methods of the dataset, the choice of loss functions, neural network architecture, and the number and types of segmentation classes. Optimal network parameters were selected to save computational resources without reducing the quality of the training. Issues arising from classifying a large number of classes were discussed, and two practical application scenarios for segmenting individual veins of Carpinus betulus leaves using the neural network were proposed.

Keywords: neural network, machine learning, segmentation, deep learning, loss functions, validation set, training set, augmentation

ЗМІСТ

ВСТУП

Актуальність. Виходить все більше наукових статтей про застосування машинного навчання в біомедицині, біомедичній інженерії та біологічних досліджень. Воно і не дивно адже використання методів машинного навчання є найбільш точним і іноваційним в наш час. Біомедичний сигнал аналіз – це продукт застосування інженерних методів до проблеми медицини, охорони здоров'я та біології. Більшість методів машинного навчання є нелінійними відображення між входами та виходами моделі, що представляє роботу біомедичної системи.

Нейромережі, особливо глибокі нейронні мережі (глибоке навчання), знайшли широке застосування в біологічних дослідженнях через їх здатність до аналізу складних даних, розпізнавання патернів і здійснення прогнозувань на основі навчання з великою кількістю даних. Деякі з основних областей застосування нейромереж у біологічних дослідженнях включають:

Геноміка та генетика: Нейромережі можуть використовуватися для аналізу генетичних даних, включаючи передбачення функцій генів, виявлення змін у геномі, ідентифікацію мутацій, прогнозування вікових та інших залежностей, а також для розв'язання проблем в геномній медицині. Наприклад, навчити модель передбачати, чи є певна послідовність генома кодуючим регіоном, якщо він має певні характеристики.

Протеоміка: Нейромережі можуть допомагати у виявленні та аналізі білкових структур, виявленні біомаркерів, класифікації білків за їх функціями та взаємодіями, а також у розробці нових методів аналізу мас-спектрометрії. Наприклад, можемо використати алгоритми аналізу важливості функцій (наприклад, SHAP або permutation feature importance) для визначення, які мас-спектрометричні характеристики найбільш важливі для класифікації певних станів хвороби.

Медична діагностика: Нейромережі можуть використовуватися для аналізу зображень медичних зображень, таких як зображення рентгенів, КТ,

МРТ, а також діагностування за симптомами та історією пацієнта. Нейромережі можуть використовуватися для автоматичного виявлення ознак ракових пухлин на зображеннях МРТ та інших медичних зображеннях. Хоча це скоріше не як повна діагностика, а доповнення до неї.

Фармакологія та дослідження ліків: Нейромережі можуть допомагати у прогнозуванні властивостей лікарських засобів, дослідженні їх ефективності та побічних ефектів, а також у відборі кандидатів на нові ліки.

Екологія та охорона навколишнього середовища: Нейромережі можуть бути використані для аналізу даних про вплив забруднення, моніторингу біорізноманіття, передбачення змін клімату та інших екологічних проблем.

Еволюційна біологія: Нейромережі можуть бути використані для моделювання процесів еволюції, аналізу генетичного різновиду та дослідження філогенетичних відносин між видами.

Нейронаука: У вивченні нейронаук, нейромережі можуть використовуватися для аналізу мозкових сигналів, моделювання нейродинаміки та вивчення когнітивних функцій.[1]

Морфометрія рослин — це наука, яка вивчає форми, розміри та структурні характеристики рослин. Вона використовує різноманітні методи вимірювання, щоб описати морфологічні ознаки рослин, такі як висота, ширина, об’єм, кількість листків і так далі. Морфометрія допомагає в дослідженні різноманітності видів, адаптацій до умов середовища, а також у селекції та генетичних дослідженнях. Цей підхід може включати як традиційні, так і сучасні методи, такі як комп’ютерне моделювання та аналіз зображень.

Листова морфометрія активно використовується в різних наукових дисциплінах, таких як таксономія, систематика, біологія розвитку, морфологія, агрономія та селекція рослин. Форма листя має велику варіативність, що робить її корисним показником для ідентифікації видів або генотипів, вивчення розвитку в межах та між особинами, оцінки здоров'я рослин і аналізу впливу навколишнього середовища на фенотип рослин.

Традиційні методи морфометрії потребують ручних інструментів та доступу до зразків, проте сучасні зусилля з цифровізації ботанічних колекцій роблять цифрову морфометрію більш доступною і науково обґрунтованою альтернативою.

Об’єкти досліджень. Здатність нейромережі аналізувати та обробляти морфологію рослин.

Предмет досліджень. Оброблене нейромережею жилкування листків.

Мета дослідження. Метою роботи є дослідження можливостей U-Net подібних згорткових нейронних мереж для мультіклассової семантичної сегментації малюнку жилкування листя Carpinus betulus.

Завдання. В рамках мети розв’язувались такі задачі:

• створення набору даних листя Carpinus betulus для семантичної сегментації малюнку жилкування;

• розробка відповідного програмного забезпечення;

• здійснити експериментальне налаштування та навчання низки моделей;

• дослідити вплив на результат роботи моделі: розміру вибірки, способів аугментації, виду функцій втрат, архітектури нейромережі, кількості та типів класів сегментації;

Методи дослідження. Є дослідження можливостей UNet подібних згорткових нейронних мереж для семантичної сегментації малюнку жилкування листя Carpinus betulus.

Наукова новизна дослідження. Полягає у розробленні більш точного та швидкого інструменту фіксації та обробки морфології рослин.

Практичне значення отриманих результатів. Данне дослідження може допомогти у ботанічних дослідженнях.

Апробація. Було прийнято участь у Науково-практичній конференції з міжнародною участю 2024 року на тему: «Медико-технічна співпраця заради перемоги: актуальні завдання медичної, біологічної фізики та інформатики». Тема тези: «Зв'язок між швидкістю відновлення серцевого ритму після фізичного

навантаження та варіабельністю серцевого ритму» де також використовувались python бібліотеки для біологічних досліджень.

Структура і обсяг магістерської роботи. Магістерська робота складається зі вступу, огляду літератури, об’єктів та методів дослідження, експериментальної частини, результатів, висновків та перспектив подальших досліджень викладених на сторінках друкованого тексту. Список використаних джерел включає 57 найменування. Загальний обсяг роботи 89 сторінок. Робота містить 29 рисунків.

Дослідження проводилось протягом п’яти місяців 2024 р. на кафедрі біофізики та фізіології Донецького національного університету імені Василя Стуса.

РОЗДІЛ 1. ОГЛЯД ЛІТЕРАТУРИ

Визначення рослин за допомогою польових спостережень потребує значних знань у ботаніці, що ускладнює цю задачу для багатьох природолюбів. Традиційні методи ідентифікації рослин часто є досить складними навіть для професіоналів, таких як природоохоронці, фермери, лісники та ландшафтні архітектори, які регулярно стикаються з ботанічними проблемами. Для ботаніків визначення видів може стати справжнім викликом через різноманітність форм і особливостей рослин.

У цьому документі пропонується нова модель на основі глибокого навчання, яка використовує архітектуру згорткових нейронних мереж для класифікації рослин на основі аналізу структури листя та форми жилок. Використання машинного навчання дозволяє автоматизувати процес ідентифікації, спрощуючи завдання для не тільки професіоналів, але й для любителів природи. Модель навчена на великій базі даних зображень листя різних видів рослин, що дозволяє їй точно визначати види за характерними ознаками, такими як форма, розмір та текстура жилок.

Цей підхід відкриває нові можливості для вивчення та збереження рослинності, роблячи ідентифікацію доступнішою і швидшою. Завдяки інтеграції технологій глибокого навчання в ботаніку, ми можемо значно поліпшити точність і ефективність досліджень у цій галузі, що особливо важливо в умовах зміни клімату та втрати біорізноманіття.

1.1. Набори даних листя рослин

За останні кілька років розвиток цифрових технологій, комп’ютерів і мережевих систем призвів до зростання популярності створення, обробки та обміну цифровими зображеннями. Внаслідок цього в Інтернеті з'явилася величезна кількість таких зображень. Разом із збільшенням їхньої кількості зросла потреба у складних системах пошуку зображень, адже традиційні методи,

які базуються на текстових описах, таких як імена файлів або ключові слова, стають малоефективними.

Щоб вирішити цю проблему, дослідники розробляють системи пошуку зображень на основі вмісту (CBIR), які можуть автоматично виявляти основні об’єкти на зображеннях і генерувати корисну інформацію про них, включаючи їхні форми, текстури та кольори. Такі методи мають широкий спектр застосувань, особливо в умовах поширення портативних пристроїв, що дозволяє користувачам здійснювати пошук інформації в будь-якому місці [2].

Наприклад, під час прогулянки в ботанічному саду люди можуть зустріти незнайому рослину. Замість того, щоб чекати повернення додому для пошуку інформації в книжках, вони можуть сфотографувати рослину за допомогою свого пристрою і відправити зображення на віддалену базу даних через бездротове з’єднання, отримуючи відповіді на свої запитання на місці. Аналогічно, рибалка може скористатися такою ж технологією для отримання інформації про тільки що спійману рибу, просто описуючи її особливості у відповідному додатку на своєму пристрої.

Для ефективної роботи систем CBIR важливо вміти визначати і представляти найбільш значущі ознаки зображень, які потім можуть бути використані для пошуку схожих зображень у базі даних. У випадку з листям, звернення до кольору чи текстури може бути недоцільним, оскільки ці характеристики можуть бути схожими у багатьох видах. Натомість форма листя може дати більш чітке уявлення для звуження кола пошуку. Досліджуючи жилкування листя, можна виявити, що різні види мають унікальні візерунки, які відображаються у структурі їхніх жилок.

У цьому дослідженні ми шукаємо зображення листя, та аналізуємо жилкування для початкової категоризації, також використовуємо форму листя для подальшого пошуку схожих зображень у базі даних. Жилкування представлене у вигляді ліній різного кольору, які визначаються вручну[3][4].

1.1.1. Морфологія листків

Типовий листок має складну будову, що включає кілька ключових елементів, які виконують різні функції. Основною частиною листка є листова пластинка, або пластинка, яка є найширшою частиною. Вона відповідає за фотосинтез, адже містить хлорофіл, що поглинає світло.

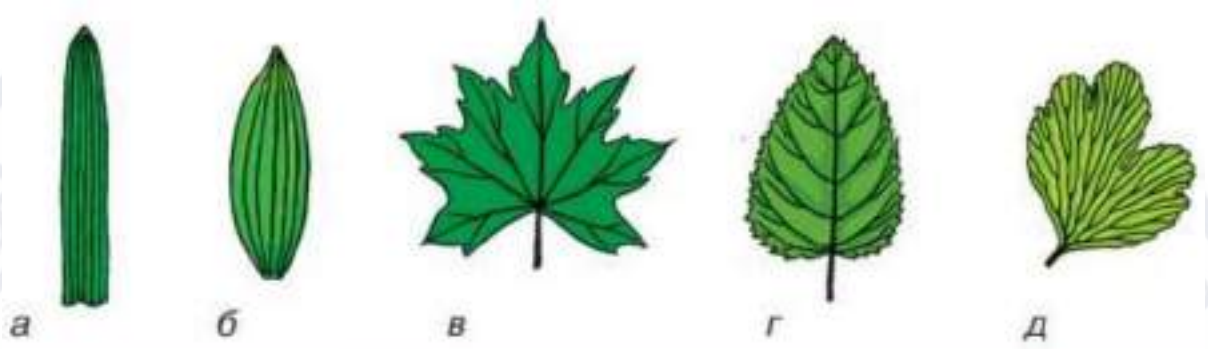
Багато листків прикріплені до стебла рослини за допомогою черешка — стеблинки, що з'єднує пластинку з основою. Листки без черешків, які прикріплюються безпосередньо до стебла, називаються сидячими. Біля основи черешка часто розташовані невеликі зелені придатки, відомі як прилистки, які можуть виконувати захисну функцію або допомагати у фотосинтезі.

У більшості листків є центральна жилка, яка проходить по всій довжині листка і розгалужується на менші жилки, формуючи судинну мережу. Ця судинна тканина відіграє важливу роль у транспортуванні води і поживних речовин, а також у відведенні продуктів фотосинтезу. Край листка, або край, може мати різні форми — бути гладким, зубчастим або хвилястим, що також може служити для ідентифікації рослин.[5]

Малюнок жилкування, тобто розташування жилок на листку, є важливим критерієм для класифікації рослин. Однодольні рослини, такі як злаки, зазвичай мають паралельне жилкування, де жилки розташовані паралельно одна до одної. У дводольних рослин, таких як бобові, жилки утворюють сітчастий візерунок, який може бути дуже різноманітним. Гінкго білоба є прикладом рослини з унікальним дихотомічним жилкуванням, яке також використовується для ідентифікації.

Розташування листя на стеблі, відоме як філотаксія, може суттєво варіюватися залежно від виду рослини. Листя класифікуються як чергові, спіральні, супротивні або мутовчасті. У чергових та спіральних листках на одному вузлі розташовується лише один лист. Чергові листки чергуються з обох боків стебла, у той час як спіральні розташовані в спіральному порядку. У супротивному розташуванні два листки виникають з однієї точки, що дозволяє

їм зростати протилежно один одному. Якщо в одному вузлі є три або більше листків, таке розташування класифікується як мутовчасте. Це різноманіття форм і структур листя не лише виконує функціональні завдання, але й допомагає рослинам адаптуватися до різних умов середовища [6].



Мал. 106. Жилкування листків:
а — паралельне; *б* — дугове; *в* — пальчасте; *г* — пірчасте; *д* — вильчасте

Рисунок 1.1 – Типи жилкування листків

1.1.2 Листкова пластинка

Визначення виду рослини відбувається в тому числі за формою листкової пластинки. Листкова пластинка — це основна частина листка, що виконує критично важливу роль у фотосинтезі, процесі, який дозволяє рослинам перетворювати сонячну енергію на органічні речовини. Листкова пластинка забезпечує максимальну площу для захоплення світла, а черешок служить для оптимального розташування пластинки щодо сонячних променів. Завдяки черешку, лист може рухатися, уникаючи ушкоджень від вітру.

Листки поділяються на черешкові, які мають черешок, і сидячі, що безпосередньо прикріплені до стебла. Сидячі листки можуть охоплювати стебло, ставши стеблообгортними, або мати інші форми, такі як збігаючі чи пронизані. Наприклад, у деяких рослин, як-от ласкавець, листок може мати пронизану основу, що дозволяє стеблу проходити крізь нього.

У рослин, таких як злаки, осоки та орхідні, нижня частина листка розширена й утворює піхву. Вона захищає молоді бруньки і сприяє зміцненню стебла, особливо під час сильного вітру. У більшості рослин у базі листка можна знайти

прилистки — спеціальні вирости, які виконують різні функції: від захисту бруньок до участі в фотосинтезі.

Жилки, що проходять через листкову пластинку, називаються провідними пучками. Вони функціонують як "транспортні шляхи", постачаючи в листок воду та мінеральні солі з коренів і відводячи готові органічні речовини назад у рослину. Структура жилок також надає листковій тканині міцність: паренхіма, що складає основну частину листка, є м'якою, але жилки забезпечують їй потрібну підтримку.

Залежно від характеру жилкування виділяють кілька типів листків: паралельножилкові (типові для злаків і осок), дугожилкові (як у конвалії), пальчастожилкові (у клена) та пірчастожилкові (у груші та яблуні). Кожен тип жилкування має свої особливості, які допомагають рослинам адаптуватися до умов середовища.

Форми листкової пластинки варіюються від голчастих (як у сосни) до лінійних (як у злаків). Довгі й вузькі листки характерні для рослин, таких як пшениця, тоді як довгасті листки можна знайти в каштанів та евкаліптів. Листки можуть бути також ланцетними, еліптичними, овальними, округлими або навіть стрілоподібними, залежно від рослинного виду.

Край листка може бути цілокраїм, з гладкою лінією, або мати різні типи зубчастості: зарубчасті, пилчасті, зубчасті чи виїмчасті. Ці характеристики надають листкам унікальний вигляд і сприяють їх впізнаваності серед рослин.

Складні листки складаються з кількох окремих листочків, які прикріплені до спільного черешка. Вони можуть бути трійчастими, пальчастоскладними або пірчастоскладними. Це дозволяє рослинам з легкістю змінювати положення листочків, що є особливо корисним у несприятливих умовах.

Величина листкової пластинки також варіює: від дрібних листків у більшості рослин помірного клімату до гігантських листків тропічних видів, таких як банан або вікторія. Наприклад, листки лопуха можуть сягати значних розмірів, а в тропі рослини, як-от "дерево мандрівників" з Мадагаскару, мають величезні листки, які використовують для покриття дахів.

Розмаїття форм, розмірів і структур листків — це результат тривалих еволюційних процесів, які допомогли рослинам адаптуватися до різних умов середовища. Ці адаптації не лише дозволяють рослинам виживати, але й додають їм естетичної краси, що робить їх важливими як в природі, так і в культурі людини [9].

1.1.3 Алометричний показник

Співвідношення жилок і пластинок є алометричним показником розміру та пластичності листя. У вивченні різноманіття форм і розмірів листків важливу роль відіграють їх кореляції з кліматичними чинниками та фізіологічними обмеженнями. Більш ніж сто років тому було виявлено, що великі цілісні листки зустрічаються в тропіках, тоді як у помірних зонах переважають менші, розсічені листки. Це спостереження дало змогу використовувати зазубреність листків як індикатор давніх кліматичних умов.

Також встановлено, що співвідношення між площею листкової пластинки і жилками змінюється в процесі еволюції. У більших, зрілих листках первинні і вторинні жилки ширші, але їхня щільність нижча порівняно з меншими листками. Це має значення для фізіології листя, зокрема для гідравлічної провідності, що, в свою чергу, пов'язано з функцією і формою листя.

Щоб краще зрозуміти еволюційні патерни, важливо вивчати розвиток листя та реакцію рослин на навколишнє середовище. Необхідно розглянути генетичні, екологічні та взаємодіючі фактори, які впливають на форму і розмір листя. Дослідження показують, що програма розвитку рослин може адаптуватися до змін у середовищі, що призводить до різних морфологічних результатів навіть у рослин з однаковим генотипом.

Морфометричні методи, такі як еліптичні дескриптори Фур’є та топологічний аналіз даних, дозволяють порівнювати форми листя в рамках різних еволюційних відстаней. Наприклад, вивчення листя виноградної лози

може забезпечити важливу інформацію для класифікації та ідентифікації сортів [10].

Загалом, результати вказують на те, що співвідношення жилок до лопатей є обіцяючим індикатором, який відображає взаємозв'язок між морфологією листя та кліматом, і вимагає подальших досліджень для глибшого розуміння цих процесів.

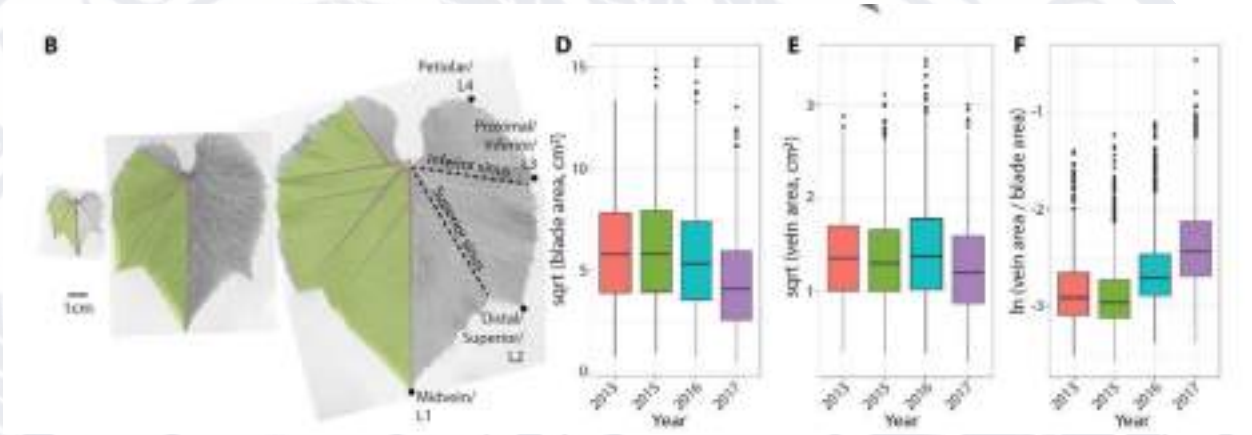


Рисунок 1.2 – Зміна алометричного показника з часом

1.1.4 Довжина жилки

Збільшення довжини жилки компенсує площу листя, втрачену через частки виноградної лози. Листки виноградної лози (Vitis) є одними з найбільш вивчених. Ампелографія, як наука, присвячена їхньому вивченню, виникла в 1902 році завдяки Луї Равазу і набула популярності завдяки П’єру Гале у 1979 році. Використання морфометричних технік, зокрема гомологічного маркування, продовжується до сьогодні.

Гомологічне маркування базується на спільній морфології листків виноградної лози, що включає п’ять основних жилок і шість основних розгалужених жилок. Незважаючи на численні дослідження, питання щодо лопатевості як основного джерела варіацій форми листків залишаються відкритими. Лопатевість сприяє проникненню сонячного світла в крону виноградної лози, що є корисним для виноградарів, але важливо, щоб більш лопатеві листки не зменшували загальну площу листкової пластинки, оскільки це могло б знизити фотосинтетичну здатність виноградної лози.

У досліджені 2022 року в American Journal of Botany використали техніки гомологічного маркування для вимірювання 2632 листків протягом двох років у 476 унікальних генетично відмінних виноградників, що походять з п’яти біпарентних гібридів, які відрізняються переважно ступенем лопатевості.[11]

Загальна площа листка, а також площі листкової пластинки і жилок були обчислені за допомогою алгоритму шнурків, який дозволяє розрахувати площу багатокутника, використовуючи опорні точки як вершини. Крім того, ми обчислили співвідношення площі жилок до площі листкової пластинки.

Ступінь дистальної та проксимальної лопатевості кожного листка визначався через довжину дистальних та проксимальних жилок, розраховуючи відстань між відповідними опорними точками. Значення дистальної лопатевості визначалося як відношення довжини дистального синуса до довжини дистальної лопаті, і аналогічні розрахунки проводилися для проксимальної лопатевості. Загальна довжина головних жилок обчислювалася як сума довжини середньої, дистальної та проксимальної вен.

Результати полягають в тому, що хоча лопатевість була основним джерелом варіації форми листків, площа листка змінювалася лише незначно в залежності від лопатевості. Натомість площа листка збільшувалася у зв'язку з загальною довжиною головних жилок, загальною довжиною розгалужених жилок та співвідношенням жилок до пластинки. Ці зв’язки були більш вираженими для листків з високою лопатевістю, причому залишки для кожної моделі відрізнялися залежно від дистальної лопатевості.

Листки з різним ступенем лопатевості, але однаковою площею, мають довші жилки та вищі співвідношення жилок до пластинки. Це дозволяє їм підтримувати однакову площу листка, незважаючи на збільшену лопатевість. Ці дані демонструють, як більш лопатеві листки можуть компенсувати потенційне зменшення площі листкової поверхні, що сприяє збільшенню фотосинтетичної здатності при збереженні однакового розміру листків [12].

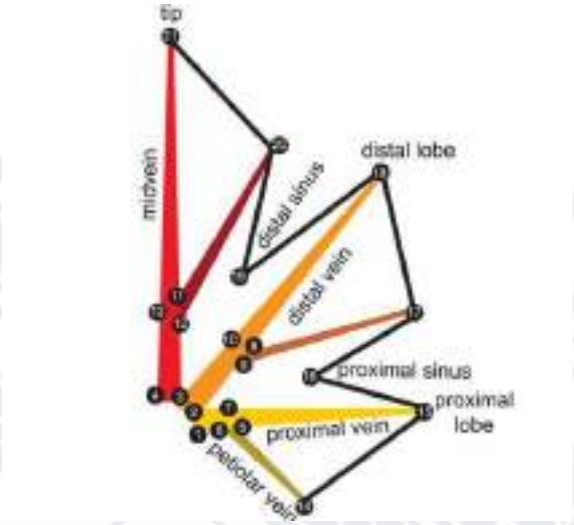


Рисунок 1.3 – Вимірювання форми та розміру листків

Схематична діаграма ілюструє 21 опорну точку, використану в дослідженні для вимірювання форми та розміру листків. Основні жилки позначені, а жилки, що розгалужуються, виділені темнішими відтінками. Відображені листки з різних популяцій демонструють варіації лопатевості [13].

1.1.5 Принцип золотого розподілу жилок

Листя дерев складається з тонкого шару мезофілу, який виконує фотосинтез за допомогою сонячного світла, і товстих жилок. Хоча відомо, що жилки забезпечують транспортування води, їхня роль у підтримці структури та створенні великої поверхні для поглинання сонячної енергії досліджувалася значно менше і залишається не до кінця зрозумілою. Дослідження з Scientific Reports використовує новий підхід, званий зворотною оптимізацією, щоб виявити принципи дизайну матеріалів, які лежать в основі особливого розподілу жилок.

Цікаво, що розподіл жилок, близький до золотого перетину, завжди демонструє оптимальні структурні властивості. Це дослідження вперше показує, що цей унікальний розподіл є корисним для підвищення жорсткості на вигин і створення великої площі, освітленої сонцем, що є важливим для фотосинтезу. Крім того, розподіл, схожий на золотий перетин, зустрічається у різних формах

і товщинах листя рослин, де наведені експериментальні докази для підтвердження оптимізації. Отже, результати цього дослідження не лише пояснюють, чому розподіл жилок листя слідує принципам золотого перетину, але й пропонують корисні рекомендації для дизайну м'яких структур з відмінними механічними властивостями [14].

Рисунок 1.4 - Золотий перетин у типових листках

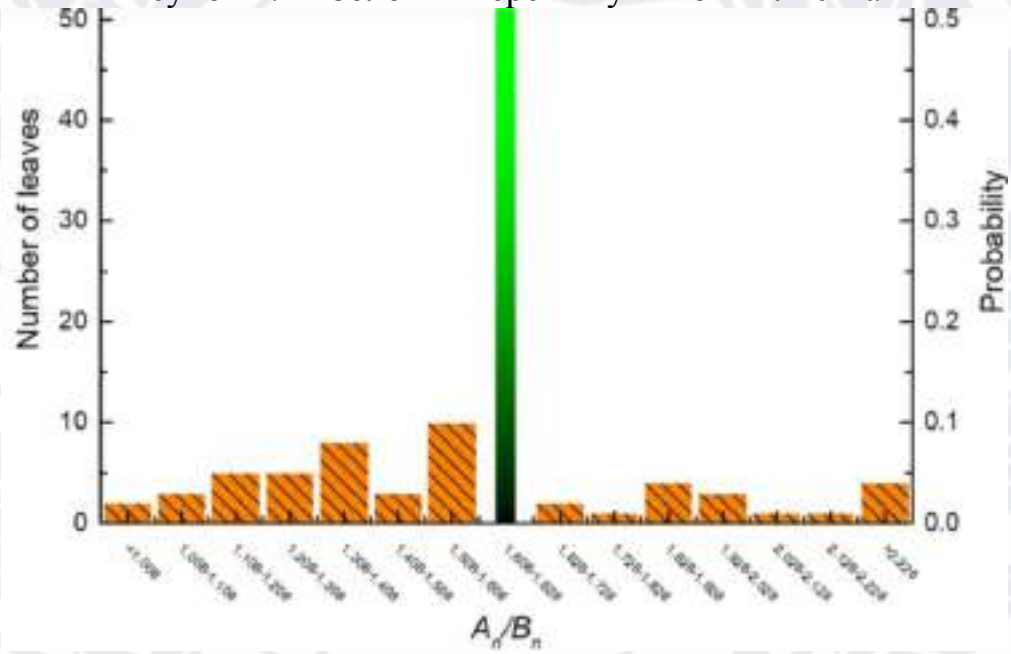


Рисунок 1.5 – Експериментальний розподіл A/B

1.1.6 Щільність жилок листя та клімат

У дослідженні з Research Gate зазначається, що характеристики жилок листя рослин частково є результатом адаптації до навколишнього середовища протягом тривалої еволюції. Проте не було встановлено загальних висновків щодо того, як щільність жилок змінюється вздовж кліматичного градієнта або як цей зв'язок впливає на зміни клімату.

У цьому дослідженні аналізуються варіації щільності жилок та інших властивостей листя дуба східного (Quercus variabilis) у 10 популяціях, які ростуть

в природних умовах (in situ), а також у 7 популяціях, вирощених у звичайному саду.

Результати показали, що щільність дрібних жилок у популяціях in situ значно зменшувалася зі збільшенням широти ($r^2 = 0,44$, $P = 0,04$), а ця тенденція зберігалася і для садових популяцій ($r^2 = 0,67$, $P = 0,02$). Щільність незначних жилок позитивно корелювала зі середньою річною температурою (MAT) місць походження ($r^2 = 0,66$, $P = 0,03$ для садових популяцій; $r^2 = 0,37$, $P = 0,06$ для популяцій in situ), але кореляція з середньою річною кількістю опадів (MAP) не була суттєвою. В порівнянні з MAT і MAP, щільність жилок мала набагато нижчу кореляцію з кліматичними умовами поточного року або сезонами вирощування.

Також виявлено, що для садових популяцій щільність дрібних жилок зростала разом зі збільшенням сухої маси листя на площу та зменшувалася з довжиною черешка і довжиною листя. Ці результати свідчать про те, що щільність жилок листя є генотипною характеристикою, яка склалася в процесі тривалої еволюції, і вона не реагує на короткострокові зміни в умовах вирощування [15].

Для вимірювання щільності жилок використовували різні методи: для невеликих жилок взяли зразок розміром близько 1 $см^2$ з адиксальної поверхні листка. За допомогою мікроскопа (LEICA DM 2500) зробили п’ять зображень для кожного зразка, а потім виміряли довжину жилок за допомогою програмного забезпечення ImageJ. Щільність жилок виражалася як сума довжини всіх сегментів жилок на одиницю площі.

Таким чином, результати показують, що щільність жилок листя цього широко поширеного виду дуба негативно корелює з широтою, але позитивно з середньою річною температурою в природних популяціях, в той час як зв’язок між щільністю жилок і кліматом залишається незмінним для садових рослин. Також, щільність жилок не має істотної кореляції з концентрацією поживних речовин у грунті або кліматичними умовами.

1.1.7 Довжина жилки та щільність продихів

Судинні жилки в листках відіграють ключову роль у транспортуванні води та цукрів, а їх структура і щільність впливають на гідравлічну ефективність листка. Також важливою є щільність продихів, яка може впливати на газообмін під час фотосинтезу. Однак зв'язок між щільністю жилок і щільністю продихів (SD) залишається мало дослідженим.

У дослідженні 2024 року було проаналізовано 16 листків гібрида Photinia × fraseri та 16 листків одного з його батьків, P. serratifolia, з метою вивчення кореляції між щільністю жилок і SD. Вирізали квадрати з листків уздовж двох ліній — одну по середній жилці, іншу по краю. Потім зробили мікрофотографії продихів і жилок, щоб визначити загальну площу жилок на одиницю площі (VAA) та загальну довжину жилок на одиницю площі (VLA).

Результати показали, що між SD і VAA не було значної кореляції, але виявилася істотна позитивна кореляція між SD і VLA. Це свідчить про те, що майбутні дослідження повинні зосередитися на зв'язках між щільністю продихів і загальною довжиною жилок на одиницю площі, а не на площі жилок на одиницю площі [16].

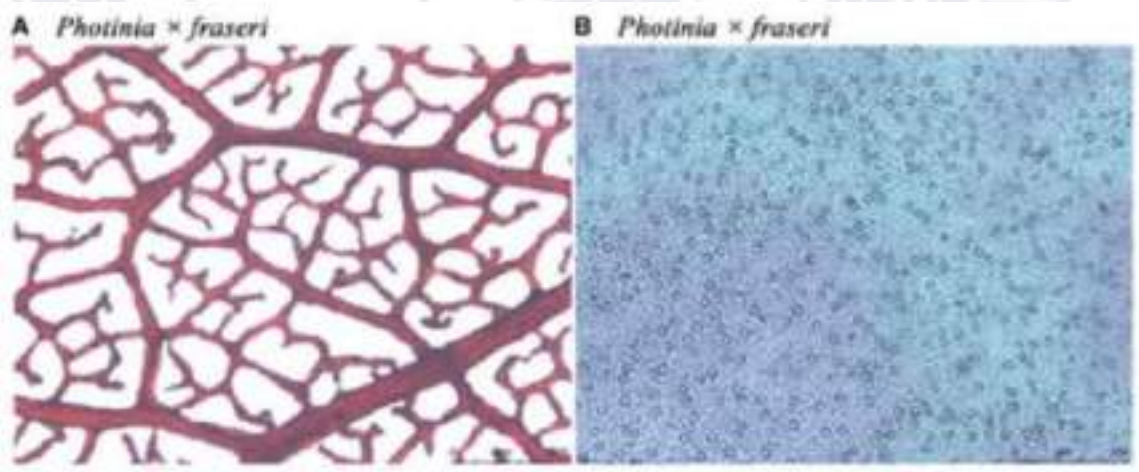


Рисунок 1.6 - Продихи та жилки рослини Photinia × fraseri

У іншому дослідженні було виміряно площу листка (LA), щільність малих жилок (MVD), щільність продихів (SD) та нову функціональну ознаку — «кількість стиків на довжину малої жилки» (SV) на арахісі (Arachis hypogaea),

вирощеному в різних умовах освітлення. Також були досліджені сонячні листки Dalbergia odorifera і Desmodium renifolium.

Результати показали, що SV залишався стабільним незалежно від листового вузла або LA в умовах різного освітлення, тоді як затінення призводило до значного зниження SV. Позитивна кореляція між SD і MVD була виявлена в арахісі при всіх режимах освітлення. У сонячних листках D. odorifera і D. renifolium також спостерігалася стабільна SV уздовж листового вузла з позитивною кореляцією між MVD та SD.

Ці результати вказують на те, що стабільний SV, разом із позитивною кореляцією між MVD та SD, може свідчити про ефективну координацію між водопостачанням і потребами листка за певних умов освітлення. Висновки підкреслюють важливість SV та відкривають нові перспективи для розуміння зв'язку між кількістю продихів і довжиною малих жилок [17].

1.2 Методи машинного навчання

Вилучених ознак медичної проблеми недостатньо для повного пояснення патології в багатьох випадках. Особливо під час опису проблеми використання надлишкового або неоптимального набору функцій викликає проблему. Краще припустити, що існує нелінійний зв’язок між входом і виходом біомедичної системи, замість того, щоб шукати кращі характеристики. Наприклад, автоматична діагностична система для виявлення нейропатичних і міопатичних захворювань за допомогою сигналів ЕМГ використовує оброблені дані ЕМГ як вхідні дані. Мета полягає в тому, щоб дізнатися про зв’язок між інформацією, наданою системі, та пов’язаною патологією. Після навчання, коли ми подаємо нові дані в систему, вона розпізнає правильну патологію. Таке завдання можна легко виконати за допомогою техніки машинного навчання [18].

Нечітка логіка, контрольоване навчання та еволюційний алгоритм є основою сучасних прикладних технологій машинного навчання. Існують деякі інші технології, які все ще досліджуються, такі як символічне навчання, навчання з підкріпленням, неконтрольоване навчання та когнітивні науки.

Поняття механізму навчання приваблювало філософів ще з часів греків. Такі філософи, як Декарт, і психологи, такі як Спенсер, Джеймс і Торндайк, зробили свій внесок у розуміння механізму навчання.

Використовуючи розробки інженерно-математичних наук, були винайдені швидкі обчислювальні пристрої на основі архітектури фон Неймана. Невдовзі комп'ютери почали використовувати для навчання. Халл і Бернштейн сказали, що якщо м'язові реакційні системи можна відтворити за допомогою якогось неорганічного методу, то, безсумнівно, здатність мислення і висновків можна буде аналогічно відтворити в машині. Теорія коннекціонізму, яка була сформована на основі теорій Арістотеля (400 р. до н. е.), була реалізована та перевірена розробкою електронних комп'ютерів протягом 1950-х років. Таким чином, народилася сфера «машинного навчання», а пізніше термін було змінено на «обчислювальний інтелект».

Існує багато методів СІ, таких як контрольоване навчання, неконтрольоване навчання, символічне навчання та GA. Серед цих парадигм навчання найбільш дослідженим був метод навчання під керівництвом, який ґрунтувався на оцінці функцій. Зовнішній супервізор надає набір прикладів формули навчання під наглядом із позначкою класу. Система розпізнає приховані зв'язки між набором зразків і бажаним результатом. Після цього етапу навчання можна передбачити результат для невідомих прикладів. Навчання з підкріпленням, мінімізація ризику та стохастичне навчання є деякими парадигмами навчання під керівництвом. GA використовуються для оптимізації, яка базується на еволюційній теорії біологічних систем. Новий метод, символічне навчання, все ще розробляється, і дослідження зосереджені на пошуку способу навчити машину знаходити зв'язок між символами та перцептивним навчанням [19].

Здається можливим об'єднати концепцію навчання людини з розвитком методів машинного навчання після появи штучних нейронів. Сьогодні комп'ютери мають достатньо потужності, щоб обробляти числа та символи швидше, ніж люди. Тим не менш, сучасних комп'ютерів на основі архітектури

фон Неймана все ще недостатньо для вирішення проблем, які вимагають комплексного сприйняття.

Зрештою, комп’ютерна реалізація показує, що можна розробити програми, які добре працюватимуть із сучасними комп’ютерними технологіями. У біомедицині для візуалізації необхідні математичні моделі живих систем. Зараз методи машинного навчання все частіше використовуються для моделювання біологічних систем. Сьогодні ми знаємо, що комп’ютери можна використовувати як електронну та віртуальну лабораторію [20].

1.3 Штучні нейронні мережі

Мозок служить джерелом натхнення для моделей ШНМ. Він представляє собою пристрій, який обробляє інформацію та має деякі надзвичайні здібності, які виходять за межі поточних біомедичних застосувань. Якщо було б можливо зрозуміти спосіб, яким мозок працює і виконує ці функції, алгоритмічні рішення для цих завдань можна було б визначити та, в кінцевому рахунку, реалізувати на комп'ютерах. Є досить значна різниця між комп'ютером та людським мозком. Хоча у комп'ютера є лише один процесор, людський мозок складається з величезної кількості обчислювальних одиниць, зокрема нейронів. Вважається, що ці обчислювальні одиниці набагато простіші та повільніші, ніж процесор комп'ютера. Однак нейрони в мозку мають з'єднання, тобто синапси, з іншими нейронами, і всі вони працюють паралельно. Ця величезна зв'язковість мозку забезпечує його обчислювальну потужність.

У комп'ютерах пам'ять представляє собою окрему частину і є пасивною, тоді як процесор є активним. У людському мозку як обчислювальні одиниці, так і пам'ять розподілені разом; нейрони виконують обробку, а пам'ять знаходиться у синапсах між нейронами [21].

Персептрон з одним шаром ваг може використовуватися лише для наближення лінійних функцій вхідних даних і не може дати рішення задач, де дискримінант, який потрібно оцінити, є нелинійним. З іншого боку, багатошаровий персептрон (БП) може реалізовувати нелинійні дискримінанти,

якщо використовується для класифікації. Вихід БП є лінійною комбінацією значень нелинійної базової функції, заданої прихованими одиницями. Приховані одиниці виконують нелинійне перетворення з простору вхідних даних d-вимірного в простір H-вимірного, який складається з прихованих одиниць, а в цьому просторі другий вихідний шар реалізує лінійну функцію. Оскільки БП не обмежується одним прихованим шаром, можна розмістити більше прихованих шарів з власними вагами після першого шару з сигмовидними прихованими одиницями для обчислення нелінійних функцій першого шару прихованих одиниць, реалізуючи більш складні функції входів. Люди майже ніколи не використовують більше одного прихованого шару на практиці, оскільки аналіз мережі, що включає багато прихованих шарів, є складним. Навчання БП ідентичне навчанню персептрона. На відміну від персептрона, вихід БП є нелінійною функцією входу через нелінійну базову функцію в прихованих одиницях. Розглядаючи приховані одиниці як вхідні дані, другий шар є персептроном, і спосіб оновлення параметрів. Іншими словами, помилка поширюється з виходу назад до входів, і тому цей алгоритм називається алгоритмом зворотного поширення помилки [22].

1.4 Алгоритм сегментації зображень рослин на основі UNet

1.4.1 Архітектура мережі

Мережа UNet була запропонована в 2015 році, і на момент її згадування основним її застосуванням була семантична сегментація медичних зображень. Поява UNet значно скоротила обсяг даних, необхідних для навчання нейронних мереж глибокого навчання, для навчання яких спочатку були потрібні тисячі анотованих даних, а також поклала початок застосуванню нейронних мереж для сегментації зображень. Ця мережа все ще широко використовується, незважаючи на народження багатьох мереж сегментації. Структура традиційної моделі UNet показана на малюнку 3. Основною особливістю цієї структури є те, що вся мережа має U-подібну структуру, звідси й назва UNet. Мережа UNet дуже проста, з першою половиною, що діє як вилучення ознак, а друга половина як

підвищення дискретизації. Цю структуру в деякій літературі також називають структурою кодера-декодера. Частина зменшення дискретизації відноситься до базової структури згорткової нейронної мережі з двома згортковими одиницями, що складаються з згорток 3x3, кожна з яких супроводжується ReLU та максимальним об’єднанням 2x2, при цьому подвоюється кількість каналів функцій для захоплення контексту для вилучення функцій і навчання. У розділі підвищення дискретизації дві згорткові одиниці, що складаються із згорток 3x3, одразу за якими йдуть згортки 2x2, удвічі зменшують кількість каналів функцій, які спочатку були подвоєні, і об’єднують їх із відповідними обрізаними картами функцій у розділі кодування. Відсутні значення на границях після кожного попереднього кроку згортки заповнюються. Нарешті, кожен 64-компонентний вектор ознак відображається на бажану кількість класів за допомогою згорткової одиниці 1x1.[23]

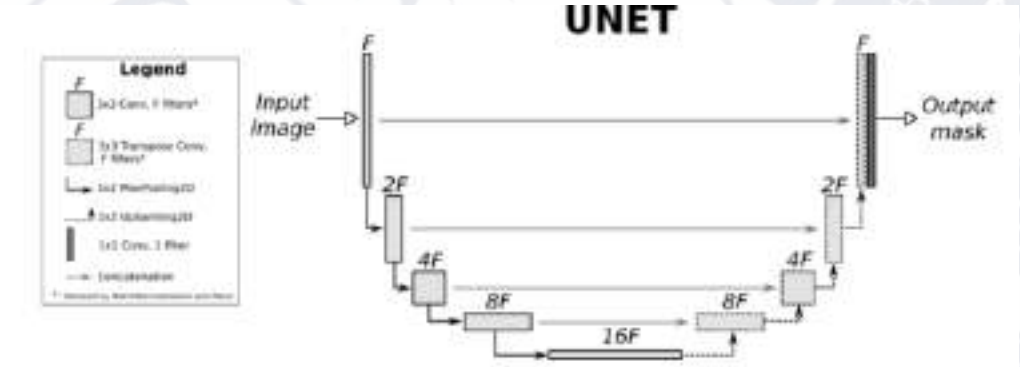


Рисунок 1.7 - Схема архітектури мережі UNet.

Модель UNet забезпечує чудові результати сегментації на різних наборах даних. Сама інформація про особливості рослин є відносно стабільною, і немає ніякої спеціальної чи нової інформації про особливості. Таким чином, як семантичні ознаки високого рівня, так і семантичні ознаки низького рівня є надзвичайно важливими. Як одна з поточних відмінних мереж семантичної сегментації, мережа UNet також має деякі недоліки. По-перше, оскільки кожна піксельна точка потребує патча, це робить ділянки двох сусідніх піксельних точок занадто схожими, що призводить до значної кількості надмірності. Ця надмірність не тільки призводить до поганої ситуації сегментації, але й знижує швидкість навчання мережі. По-друге, важко досягти як точності локалізації, так

і доступу до контекстної інформації одночасно. Чим більший розмір латки, тим більше максимальних шарів об'єднання потрібно, що, у свою чергу, знижує точність локалізації. Крім того, зі збільшенням кількості шарів об'єднання втрачається більше інформації. Тоді пряме введення неглибокої мережевої інформації в декодер призведе до низької швидкості отримання семантичної інформації країв стебла та листя, що призведе до низької точності сегментації. Щоб покращити ефективність навчання та усунути вищезазначені недоліки, у цьому документі представлено наступні вдосконалення традиційної архітектури моделі UNet: (1) Використання VGG16 як основної частини вилучення основних ознак UNet, що значно зменшує кількість обчислень параметрів моделі, зменшує зайнятість пам'яті та покращує швидкість обчислень. (2) Представлення вдосконаленого модуля ASPP у середині кодера та декодера, який розширює сенсорне поле без втрати семантичної інформації та покращує можливості вилучення функцій мережі. (3) Запровадження CCA у частині декодування, щоб зменшити використання пам'яті GPU та покращити точність сегментації моделі. (4) Заміна функції втрат складеними втратами Dice_Boundary щоб усунути дисбаланс розподілу пікселів між категоріями [24].

1.4.2 Оптимізований модуль вилучення функцій

Зовнішнє середовище викликає різні перешкоди в процесі отримання зображення Carpinus betulus. Крім того, сама мережа UNet використовує певну кількість ядер згортки в процесі кодування для вилучення функцій зображення. Ця багатоетапна згортка призводить до надмірної надмірності в карті функцій моделі сегментації, що призводить до поганої семантичної інтерпретації зображень Carpinus betulus та зниження швидкості навчання мережі. Щоб зменшити надлишковість параметрів у мережі UNet, збільшити глибину мережі для покращеної точності класифікації та витягти з зображення більш абстрактні функції вищого порядку, VGG16 використовується як магістральна мережа функцій UNet. Використання попередньо навчених зрілих моделей значно підвищує швидкість навчання мережі UNet, забезпечуючи при цьому точність. У

порівнянні з іншими мережами, VGG16 може використовувати кілька згорткових ядер 3x3 замість великих згорткових ядер, таким чином зменшуючи кількість параметрів під час роботи мережі. У цьому алгоритмі три повністю зв’язані рівні VGG16 опущені через надмірне споживання ними обчислювальних ресурсів. Крім того, VGG16 і VGG19 продемонстрували кращі ефекти сегментації в практичних застосуваннях. Порівняно з VGG19, VGG16 має три згорткових шару менше, що робить його дрібнішою мережею. Враховуючи, що якість мережі забезпечена, у цьому документі вибирається мережа з меншою кількістю параметрів [25].

1.4.3 Об'єднане отримання багатовимірних ознак

Зображення Carpinus betulus демонструють різномасштабні об’єкти, такі як дрібне та велике листя, а також гладкі гілки в порівнянні з гілками з маленькими розвилками. Для інтеграції багатомасштабної контекстної інформації зображення піддаються безперервній згортці і об’єднанню, що призводить до утворення карт функцій з низькою роздільною здатністю. Це ускладнює реконструкцію деталей зображення, оскільки виникає локальна втрата інформації та відсутність кореляції між віддаленими ознаками, що зумовлене ефектом сітки при використанні стандартної згортки.

Щоб подолати ці недоліки, пропонується змінити швидкість розширення вихідної нульової згортки в модулі ASPP. Запроваджується новий тип нульової згортки, що дозволяє збільшити сенсорне поле, зберігаючи чутливість до деталей. Оптимізована порожниста згортка ефективно розширює сприйнятливе поле, включаючи більше контекстуальної інформації без збільшення кількості параметрів та обчислювальних витрат. У алгоритмі семантичної сегментації застосовується двомірна діркова згортка, яка досягається шляхом вставки нулів між пікселями ядра. Для згортки розміром k×k, новий розмір після діркової згортки дорівнює kd×kd, де $kd = k + (k - 1) \times (r - 1)$.

Модуль ASPP використовує кілька паралельних шарів діркової згортки з різними частотами дискретизації. Функції, отримані для кожної частоти, далі

обробляються в окремих гілках і об’єднуються, що дозволяє виділяти багатомасштабні характеристики об’єкта та контекст зображення, підвищуючи його роздільну здатність. Дослідження показало, що неправильні налаштування швидкості розширення діркової згортки можуть призвести до "ефекту сітки". Тому налаштування швидкості розширення повинно уникати втрати важливої інформації та фіксувати контекст у різних масштабах [26].

Згідно з теорією, швидкість розширення для N діркових згорток розміром K×K повинна задовольняти формулі M_2 k, де ri – швидкість розширення і-ої діркової згортки, а M_i – максимальна швидкість розширення і-го шару. Темпи розширення не повинні мати спільних множників, щоб уникнути ефекту сітки. Наприклад, при K = 3 та r = [1, 2, 5] ефекту не буде, але для r = [1, 2, 9] він виникне. Тому в роботі використовуються паралельні шари згортки з різними частотами дискретизації.

Експериментально було визначено набір діркових згорток зі швидкостями розширення 1, 2, 7 і 15 (M_i = 3 = k). Функції з кожної частоти обробляються окремо і об’єднуються в кінцевий результат. Паралельна архітектура додається після четвертого блоку. Ліва частина складається з 1x1 згортки, трьох 3x3 згорток з дірками 2, 7 і 15, а також операцій об’єднання. Права сторона фокусується на об’єктах зображення, де вони згортаються і об’єднуються. Після розпаралелювання та подальшої згортки з 256 ядрами 1x1, результативна карта функцій досягає потрібного розміру.

1.4.4 Оптимізація функцій втрат

Сегментація стебла та листя Carpinus betulus має на меті ідентифікувати три категорії: стебло, лист і фон. Однак, через різну довжину рослин, відокремити листя від стебла складніше, оскільки ці дві категорії мають значно меншу просторову зайнятість у порівнянні з фоном. Це призводить до проблеми незбалансованості, оскільки звичайні методи навчання передбачають, що всі категорії мають однакову важливість. Така ситуація може викликати

нестабільність у процесі навчання та зміщення меж рішень на користь більш численних класів.

Щоб вирішити цю проблему дисбалансу, у статті пропонується комбінувати втрату Dice з функцією втрат на межі, що призводить до створення нової складної функції втрат Dice_Boundary для оптимізації моделі навчання. Вона визначається таким чином:

Функція втрат Dice, яка раніше використовувалася для вимірювання схожості між сегментованими та істинними зображеннями, допомагає зменшити неточності. Вона зазвичай варіюється в діапазоні [0, 1]: чим ближчі результати до очікуваних, тим вищий бал, що вказує на покращення продуктивності. Нижчі значення функції втрат вважаються кращими.

Остання функція втрат на межі фокусується на контурному просторі замість регіональної метрики, що полегшує роботу з дуже незбалансованими даними. Граничні втрати доповнюють регіональну інформацію і часто використовуються в задачах сегментації з високим дисбалансом. Вона визначається як:де «p» - точка на контурі «A», а «q» - відповідна точка на «B», що вказує на перетин з «A», знайденим у «B'. Оператор '||.||' позначає норму.

Крім того, відстань між парними контурами визначається як ΔS, а D - це карта відстані до кордонів, що означає відстань між будь-якою точкою q і найближчою точкою на контурі. Це призводить до наступного рівняння:
де S є двійковою індикаторною функцією області S, що представляє межу [27][28].

1.5.1 Використання Python бібліотек для створення нейромереж

Бібліотека в програмуванні - це готовий набір об'єктів, класів та методів, які використовуються для створення програм. Замість того, щоб кожного разу створювати все з нуля при вирішенні певної задачі, спеціаліст може скористатися готовими бібліотеками, де вже реалізовані необхідні функції та механізми оптимізації обчислень. Використання бібліотек є важливою складовою функціонального підходу до написання програм.

Python - це високорівнева об'єктно-орієнтована мова програмування. Для задач машинного навчання існує багато бібліотек. NumPy, Pandas та Matplotlib.

NumPy - це фундаментальний пакет Python, який дозволяє створювати та виконувати операції з N-вимірними масивами. Він також має механізм векторизації, що підвищує продуктивність та прискорює виконання операцій.

Pandas - це пакет, призначений для маніпуляцій з даними. Крім математичних обчислень, він забезпечує їх агрегацію та візуалізацію.

Matplotlib - це основний інструмент візуалізації даних на Python. Хоча бібліотека є низькорівневою і вимагає багато коду для візуалізації, вона є дуже продуктивною.

SciKit-Learn - це бібліотека, яка призначена для вирішення класичних задач машинного навчання. Вона має широкий набір алгоритмів навчання як з учителем, так і без нього. Однак SciKit-Learn не підтримує паралельні обчислення, тому вона не є оптимальним рішенням для завдань глибокого навчання.

Theano - це бібліотека з відкритим вихідним кодом, розроблена групою машинного навчання в університеті Монреаля. У ній обчислення виражаються за допомогою синтаксису, схожого на NumPy, і компілюються для ефективних паралельних обчислень як на звичайних ЦП, так і на ГПУ. Theano є препроцесором на мові Python для системи обчислень з багатовимірними масивами даних (тензорами), який поєднує в собі математичні пакети Mathematica і MATLAB. Ця бібліотека підтримує створення архітектур згорткових нейронних мереж.

TensorFlow - це бібліотека з відкритим вихідним кодом від Google, яка також оперує тензорами і працює на різних ГПУ та ЦП. Кожний рядок коду проходить через обчислювальний граф, тому два обчислення можуть виконуватися одночасно. Ця бібліотека дозволяє розподіляти обчислення по різних ЦП або ГПУ, що значно зменшує обчислювальні витрати.

Keras - це відкрита нейромережева бібліотека, написана на мові Python, яка працює поверх інших машинних бібліотек з відкритим вихідним кодом, таких як

TensorFlow, Theano. Керас оперує об'єктами, не занурюючись у деталі побудови математичних моделей.

PyTorch - це потужний фреймворк глибокого навчання, який надає дві основні високорівневі моделі: тензорні обчислення (аналогічно до NumPy) з розвиненою підтримкою прискорення на ГПУ та глибокі нейронні мережі на базі системи автоматичного диференціювання.

В поєднанні ці бібліотеки надають широкі можливості для вирішення задач машинного навчання. Однак для створення нейронних мереж необхідно явно визначати архітектуру мережі, а також математичний апарат (функції активації, функції втрат, потоки даних від шару до шару) [29].

1.6 Попередня обробка даних

1.6.1 Сегментація зображення

Сегментація зображення — це важливий етап цифрової обробки зображень та комп'ютерного зору, що передбачає розбивку зображення на кілька сегментів, які також називають регіонами або об'єктами. Кожен сегмент є групою пікселів, що поділяються на основі певних спільних характеристик, таких як колір, текстура чи інтенсивність. Метою сегментації є спрощення та трансформація зображення в таку форму, що дозволяє легше й ефективніше проводити подальший аналіз.

Процес сегментації може бути використаний для виявлення об'єктів на зображенні, визначення меж між різними об'єктами або структурування зображення для подальшого автоматизованого розпізнавання. У результаті сегментації ми отримуємо або набір сегментів, які покривають усе зображення, або контури, що виділяють певні об'єкти чи структури. Ці сегменти можуть відрізнятися за різними властивостями, але пікселі в межах одного сегмента повинні бути схожими за певною характеристикою, що робить цей процес надзвичайно корисним для виявлення структур, об'єктів та важливих деталей.

Однією з розвинутих форм сегментації є семантична сегментація, що полягає в категоризації кожного пікселя на зображенні за класом або об'єктом. У

цьому випадку завданням є створення карти, де кожен піксель належить до конкретного класу, наприклад, "небо", "земля", "автомобіль" або "дерево". Це дозволяє більш глибоко зрозуміти зміст зображення на рівні окремих елементів, що має величезне значення для таких застосувань, як автономні транспортні засоби, медичні зображення, робота з супутниковими знімками та інші. Результатом семантичної сегментації є детальне, попіксельне маркування, яке дає можливість точно виділяти різні об'єкти чи класи на зображеннях, забезпечуючи їх чітке розрізнення й аналіз [38].

1.6.2 Медіанна фільтрація

Зображення підлягають впливу температури, вологості, магнітних полів, втрат під час передачі сигналу, шуму від вібрацій та інших факторів, що можуть погіршити їх якість і викликати спотворення. Ці аспекти негативно позначаються на подальшому аналізі зображень. Наприклад, при отриманні зображень у цьому дослідженні варіювання фізичних джерел світла та природного освітлення в різний час доби створювало додаткові труднощі для обробки зображень.

Щоб зменшити вплив шуму, застосували методи шумозаглушення через фільтрацію зображень. Для зображень Betula luminifera використовується процес просторової фільтрації, що включає як лінійні, так і нелінійні фільтри. У цьому дослідженні перевага надається медіанній фільтрації в рамках нелінійної фільтрації для зменшення шуму. Медіанна фільтрація ефективно усуває імпульсний шум і зберігає крайові деталі. Основний принцип медіанної фільтрації полягає в заміні значення пікселя на медіану значень пікселів у його сусідстві, що дозволяє видаляти шумові точки, які значно відрізняються від навколишніх пікселів.

Для одновимірних послідовностей медіанну фільтрацію можна описати формулою, де p - кількість чисел для обробки, а m = (p-1)/2. У випадку двовимірних зображень вікно фільтра також має двомірну форму, яке може бути представлено різними геометричними фігурами, такими як квадрати чи кола.

Фільтрація середнього значення є звичайним методом лінійної фільтрації, де вихідне значення пікселя замінюється середнім значенням сусідніх пікселів. Це включає цільовий піксель та вісім його сусідів.

Двостороння фільтрація, подібно до медіанної, є нелінійним методом, що комбінує просторову близькість та подібність значень пікселів для зменшення шуму з водночас збереженням країв. У цьому дослідженні результати медіанної фільтрації майже не відрізняються від вихідного зображення, а двосторонній фільтр суттєво не впливає на значення пікселів країв. Проте, через надмірне збереження високочастотної інформації, шум у кольорових зображеннях може не усуватися належним чином, що призводить до втрати текстурних деталей, як видно на зображеннях з недостатньо вираженими жилками листя.

Фільтрація Гауса є лінійним методом, який підходить для усунення гаусового шуму і широко використовується в обробці зображень. Цей метод здійснює зважене усереднення значень пікселів, враховуючи їх сусідів. Хоча між зображеннями, обробленими методами Гауса та медіани, може бути незначна різниця, швидкість обробки при використанні фільтрації Гауса значно вища, ніж при медіанній фільтрації [39].

РОЗДІЛ 2. МАТЕРІАЛИ І МЕТОДИ ДОСЛІДЖЕНЬ

2.1 Створення набору даних

Об’єктом дослідження є здатність нейромережі аналізувати морфологію рослини за жилкуванням листків. В рамках цього дослідження було зібрано 80 зразків листя для створення бази даних, яка містить фотографії листків, знятих у різних умовах — як у місцях з меншою, так і з більшою забрудненістю. Кожен зразок оброблявся спеціальною програмою, за допомогою якої на зображеннях були позначені жилки листків. Це дозволяє створити точну модель для подальшого аналізу за допомогою штучного інтелекту.

Для тренування нейромережі важливо подбати про правильне тестування її ефективності. Для цього 20% зібраної бази даних було відокремлено для тестування моделі, що дозволяє перевірити, чи не сталося перенавчання. Перенавчання (overfitting) — це явище, при якому модель надто точно адаптується до тренувальних даних, втрачаючи здатність до узагальнення на нові, невідомі дані. Іншими словами, нейромережа може "запам'ятати" конкретні деталі тренувальних зразків замість того, щоб навчитися загальним закономірностям. Тестова частина даних (яка не використовувалася під час тренування) допомагає виявити, чи є модель перенавченою.

Програмне забезпечення, яке використовувалося для позначення жилок на зображеннях, називається Supervisely. Це інструмент для мітки даних та тренування моделей комп'ютерного зору, який дозволяє користувачам точно вказувати важливі ознаки на зображеннях, створюючи таким чином тренувальний набір даних. Supervisely надає потужні інструменти для позначення, сегментації та анотації зображень, що є необхідним етапом у підготовці даних для навчання нейромереж.

Таким чином, обране для дослідження програмне забезпечення і методи перевірки на перенавчання забезпечують високий рівень точності та надійності результатів аналізу морфології рослин на основі жилкування листків.

2.1.1 Збір зразків листя

Зразки листя граба були зібрані в кількох географічних локаціях, що дозволило порівняти морфологічні особливості листя в умовах різного рівня забруднення навколишнього середовища:

-Глибина лісопарку, далеко від доріг з інтенсивним рухом транспорту. Це місця з меншим рівнем забруднення повітря та ґрунту, де рослини мають менше стресових факторів і можуть розвиватися в більш природних умовах. Серед таких локацій були райони озера Гуральня та вулиця Генерала Арабея.

-Біля автомобільних доріг з інтенсивним рухом транспорту. Ці локації характеризуються високим рівнем забруднення через викиди транспорту, а також наявність пилу, токсичних сполук і важких металів у повітрі та ґрунті. До таких місць належать Хмельницьке шосе та вулиця Максимовича, які знаходяться в безпосередній близькості до доріг з інтенсивним рухом.

Рис 2.1 – Місця збору більш чистого листя

Рис 2.2 – Місця збору листя що росте ближче до доріг

Забруднення навколишнього середовища, зокрема викиди від транспорту, може суттєво впливати на морфологію рослин, у тому числі на структуру жилкування листя. Під впливом токсичних речовин, таких як діоксини, сажа, оксиди азоту та важкі метали, можуть спостерігатися наступні зміни в жилкуванні листя:

-Зменшення розміру та кількості жилок. У листках, що ростуть в забруднених зонах, може бути спостережено зменшення кількості головних і побічних жилок. Це може бути результатом токсичних впливів на клітини, що відповідають за транспорт води та поживних речовин, що, в свою чергу, впливає на ріст і розвиток листя.

-Деформація жилок. В умовах забруднення жили можуть ставати більш витягнутими або мати спотворену форму, що свідчить про порушення в процесах росту тканин листка під впливом шкідливих речовин. Також можливі сплутування або згущення жилок у певних ділянках листка.

-Зміна напрямку жилкування. В умовах високого рівня забруднення жилкування може змінюватися таким чином, що жили відходять від звичного напрямку і утворюють незвичні патерни, що може бути відповіддю на стреси, спричинені забрудненням.

-Зниження товщини листя та жилок. У більш забруднених умовах товщина листя може зменшуватися, що може бути наслідком порушення фотосинтетичних

процесів через забруднення атмосфери, а також через негативний вплив на водний баланс рослини.

Загалом, морфологічні зміни в жилкуванні листя можуть бути індикаторами стресу рослин, пов’язаного з забрудненням навколишнього середовища, і це може мати важливе значення для подальших досліджень впливу антропогенних факторів на флору.

2.1.2 Сканування

Процес сканування зразків листя проводився у кілька етапів, що забезпечило високу точність отриманих зображень та їх підготовку для подальшої обробки і тренування нейромережі. Кожен з етапів був важливим для забезпечення якості даних, що використовуються для аналізу морфології жилкування листя.

-Першим етапом підготовки зразків листя було ретельне миття, щоб видалити бруд, пил та інші зовнішні забруднення, які могли б спотворити результати сканування. Після миття листя висушувалося природним шляхом, щоб уникнути впливу вологи на процес сканування та забезпечити чіткість зображення.

-Сканування листя виконувалося при роздільній здатності 300-600 dpi (точок на дюйм), що дозволяло отримати детальні зображення з високою роздільною здатністю. Така роздільна здатність забезпечує точність у фіксації деталей жилкування та інших морфологічних особливостей листя.

-Після сканування зображення листя підлягали сегментації за допомогою бібліотеки OpenCV, що є потужним інструментом для обробки зображень. Сегментація полягає в виділенні конкретних частин зображення, таких як жилки листя, щоб відокремити важливі частини для подальшого аналізу. Цей процес дозволяє усунути зайві частини зображення та зосередити увагу на деталях, які є ключовими для навчання нейромережі.

-Після сегментації кожен листок вирізався з зображення за допомогою прямокутної області найменшої площини, що охоплює сам листок. Такий підхід

дозволяє уникнути зайвих артефактів на зображеннях і зберігати лише ту частину зображення, яка необхідна для подальшого аналізу. Це також дозволяє стандартизувати розміри зображень у базі даних.

Рис 2.3 – Готові зразки листя до додавання до набору данних

Цей підхід до обробки зразків забезпечує отримання високоякісних і стандартизованих зображень, що важливо для подальшого аналізу та тренування моделей комп'ютерного зору, а також дозволяє уникнути можливих спотворень через нечіткі чи неповні зображення. Зібрані та оброблені дані використовуються для ефективного навчання нейромережі, що аналізує морфологію жилкування листя.

2.1.3 Розмітка набору даних, поділ на тренувальний та валідаційний

У ході даного дослідження було вручну позначено 80 зображень листя Carpinus betulus (граб звичайний) за допомогою спеціального застосунку для мітки даних. Усі зображення листя були ретельно оброблені, і кожна жилка на них була відмічена як окремий клас, починаючи з основної жилки та продовжуючи всіма побічними жилками, що дозволяє нейромережі вивчати їхню структуру та розташування. Зібрані зображення були розділені на дві групи: 80% даних було використано для тренування моделі (тренувальний набір), а 20% — для перевірки точності її роботи та запобігання перенавчанню (валідаційний набір).

Тренувальний набір використовується для безпосереднього навчання моделі. Це той набір даних, на якому нейронна мережа оптимізує свої параметри (ваги та біаси) через процес навчання, зазвичай за допомогою алгоритмів зворотного поширення помилки та градієнтного спуску. На тренувальному наборі модель вчиться відтворювати вірні маски сегментації, тобто знаходити співвідношення між вхідними зображеннями та їх істинними масками. Зазвичай тренувальний набір складає більшу частину даних, оскільки саме на ньому модель має здобувати більшість знань. Важливо, щоб тренувальний набір був різноманітним та представницьким, адже від його якості залежить здатність моделі до ефективного навчання.

Валідаційний набір є окремим підмножиною даних, яке не використовується для безпосереднього навчання моделі, але служить для її оцінки під час процесу тренування. Він дозволяє відслідковувати, як добре модель навчається на нових даних, а також виявляти потенційне перенавчання. Перенавчання — це ситуація, коли модель надто добре вивчає тренувальні дані, але погано працює на нових, непередбачуваних прикладах, оскільки не здатна узагальнювати знання. Валідаційний набір використовується для періодичної оцінки якості моделі в процесі її тренування. Коли модель проходить через кожну ітерацію або епоху навчання, її продуктивність оцінюється на

валідаційному наборі. Ці оцінки дозволяють налаштовувати гіперпараметри моделі, такі як швидкість навчання або кількість епох, щоб досягти кращих результатів.

Запобігання перенавчанню, оцінка узагальнювальної здатності моделі, коригування гіперпараметрів і забезпечення незалежності тестових даних — це основні переваги поділу на тренувальний та валідаційний набори. Валідаційний набір допомагає виявити можливі проблеми з перенавчанням, дозволяючи своєчасно коригувати параметри моделі, та дає змогу оцінити, як модель працюватиме на нових, непередбачених даних. Він забезпечує більш точну і реалістичну картину ефективності моделі.

Зазвичай для поділу датасету на тренувальний та валідаційний набори використовують один із таких підходів: випадковий поділ, де дані випадковим чином діляться на дві частини, або перехресну перевірку, що передбачає багаторазовий поділ на тренувальні та валідаційні набори з метою кращої оцінки моделі. Це дозволяє отримати більш надійні результати, оскільки модель перевіряється на різних підмножинах даних.

Таким чином, поділ датасету на тренувальний та валідаційний набори є важливою частиною процесу навчання нейронних мереж. Він дозволяє оцінити здатність моделі до узагальнення, запобігти перенавчанню та оптимізувати гіперпараметри, що забезпечує більшу точність та надійність моделі при її застосуванні до нових даних.

Рис 2.4 – Оброблена фотографія листка Carpinus betulus для машинного навчання

Рис 2.5 – Окремо виділена жилка після обробки в Supervisely

Завдяки такій деталізованій мітці кожної жилки, дослідження дозволяє нейромережі розпізнавати не лише загальні патерни жилкування, але й вивчати варіації в їх розташуванні та характеристиках, що може бути особливо корисним при аналізі впливу навколишнього середовища, зокрема забруднення, на морфологічні зміни в рослинах.

2.1.4 Попередня обробка зображень, аугментація

Зображення відсканованих листків проходили такі етапи цифрової трансформації (всі операцій здійснювались з використанням скриптів написаних мовою Python skimage cv2):

-Конвертація кольорового зображення до відтінків сірого: Цей етап дозволяє зменшити кількість інформації, яка обробляється, оскільки кольорові зображення містять три канали (червоний, зелений, синій), а зображення в градаціях сірого — лише один. Це зменшує обсяг даних та полегшує подальшу обробку, особливо для задач комп'ютерного зору, таких як сегментація або розпізнавання.

-Зменшення розміру до 512x512 зі збереженням пропорцій листка: Зменшення розміру зображення до стандартизованого формату допомагає уніфікувати дані та зменшує час обробки. При цьому важливо зберегти пропорції, щоб уникнути

спотворення форми об'єкта. Для цього можуть бути використані методи, які підбирають найближчі розміри, а потім заповнюють зайві області, наприклад, з використанням чорного кольору.

-Обертання на випадковий кут в діапазоні [-45, 45] градусів: Цей етап додатково збільшує різноманітність даних шляхом випадкового обертання зображень. Це допомагає моделі стати більш стійкою до змін в орієнтації об'єктів. Випадкове обертання вказаного діапазону дозволяє моделі навчатися на різних позиціях листка, що підвищує її загальну точність.

-Еластичні деформації: Цей етап полягає в застосуванні еластичних трансформацій до зображень. Метод передбачає деформацію зображення за допомогою малих зсувів у кожній точці, що надає зображенню "еластичний" вигляд. Це може бути корисним для моделювання варіацій в реальних зображеннях, таких як зміни в текстурі або структурі листків. Еластичні деформації дозволяють моделі навчатися розпізнавати об'єкти з різними формами та розмірами, що робить її більш адаптивною до нових даних.

2.1.5 Пробні набори данних

Таблиця 2.1 Набори даних отримані після попередньої обробки та аугментації відсканованих зображень окремих листків

Номер	Розмір, пікселі	Палітра	Товщина ліній	Діапазон Оберт., град	Еластична деформа	Кількість зображень	Додаткові зображення
1	512x512	Gray	3	-	-	80	0
2	512x512	Gray	3	[-45 45]	відсутня	240	2
3	512x512	Gray	3	[-45 45]	відсутня	480	5
4	512x512	Gray	3	[-45 45]	відсутня	640	7
5	512x512	Gray	3	[-135 135]	відсутня	240	2
6	512x512	Gray	3	[-135 135]	відсутня	480	5
7	512x512	Gray	3	[-45 45]	наявна	240	2
8	512x512	Gray	3	[-45 45]	наявна	480	5
9	512x512	RGB	3	-	-	80	0
10	512x512	RGB	3	[-45 45]	відсутня	240	2

Додаткові зображення – зображення, що мають випадковий кут нахилу, а також додаткові трансформації, включаючи випадкові зміни масштабу, повороти, відображення та еластичні деформації. Ці додаткові зображення допомагають розширити варіативність даних для тренування нейромережі, дозволяючи моделі краще адаптуватися до різних умов і знижуючи ризик перенавчання.

Розмір, пікселі - вказує розміри зображення в пікселях по горизонталі та вертикалі. В даному випадку всі зображення мають однаковий розмір — 512x512 пікселів. Цей параметр визначає кількість пікселів, з яких складається кожне зображення, і впливає на точність та швидкість обробки зображень.

Палітра - це тип кольорової палітри, яка використовується для кожного зображення. В таблиці є два варіанти палітри:

-Gray — це відтінки сірого, тобто зображення мають тільки градації сірого кольору (одноканальне зображення).
-RGB — це кольорові зображення з трьома каналами (червоний, зелений, синій), що дозволяє отримати повнокольорові зображення.

Товщина ліній маски, пікселі - вказує на товщину ліній, які використовуються для створення масок на зображеннях. Маски можуть застосовуватися для позначення об’єктів або важливих областей на зображеннях. В таблиці товщина маски становить 3 пікселі для всіх наборів даних, що може означати стандартний розмір для позначення контурів або виділення важливих елементів.

Діапазон обертання, грд - цей параметр вказує на допустимий діапазон обертання зображень. Виглядає, що для більшості наборів даних застосовується діапазон обертання від -45 до +45 градусів. Для деяких наборів цей параметр не застосовується (позначено як "-"), тобто зображення не підлягають обертанню під час аугментації.

Еластична деформація - цей параметр вказує, чи була застосована еластична деформація до зображень. Еластична деформація дозволяє змінювати зображення за допомогою нелінійних трансформацій, що може допомогти моделі краще розпізнавати об'єкти у варіативних умовах:
-Відсутня — еластична деформація не застосовувалась.
-Наявна — еластична деформація була застосована для зображень.

2.2 Проблема незбалансованих класів у семантичній сегментації

Дисбаланс класів — це проблема, що часто виникає при семантичній сегментації зображень. Вона проявляється в тому, що деякі класи (семантичні категорії) мають значно більше пікселів, ніж інші. Це може статися через те, що деякі об'єкти з'являються частіше на зображеннях або в наборі даних, а інші —

рідше. Дисбаланс класів може призвести до кількох проблем під час навчання моделі глибокого навчання, серед яких:

Зміщене навчання: Якщо один клас домінує в наборі даних, модель може виявитися зміщеною на користь цього класу. Це означає, що вона добре працюватиме з більшими класами, але може погано сегментувати рідше представлені класи.

Погіршення узагальнення: Моделі, навчені на незбалансованих даних, можуть мати труднощі з узагальненням на нові дані, що містять слабо представлені класи.

Низький перетин через об'єднання (IoU): Якщо клас недостатньо представлений, його сегментація може мати низький показник IoU, що свідчить про погану якість результату.

Для боротьби з цією проблемою використовуються різні методи, такі як:

-Аугментація даних: Збільшення кількості даних за допомогою трансформацій, таких як обертання, масштабування та перевертання.

-Зважена втрата: Присвоєння різних ваг класам під час обчислення втрат.

-Стратегії вибірки: Надлишкова вибірка малих класів або недостатня вибірка великих класів.

-Спеціальні функції втрат: Наприклад, фокальна втрата (focal loss), яка зменшує вплив простих для класифікації прикладів.

-Генерація синтетичних даних: Використання методів, таких як генеративні змагальні мережі (GAN), для створення додаткових даних для слабо представлених класів.

2.3 Функції втрат у семантичній сегментації

Функції втрат для семантичної сегментації можна розподілити на три категорії: піксельні функції втрат, регіональні функції втрат та функції втрат для меж. Кожна з цих категорій має свій підхід до оцінки та покращення якості сегментації:

Піксельні функції втрат: Оцінюють точність класифікації кожного пікселя в межах сегментованих регіонів, зокрема через крос-ентропію або фокальну втрату. Вони орієнтовані на деталі та точність класифікації окремих пікселів.

Регіональні функції втрат: Зосереджуються на загальній точності сегментації класу, прагнучи максимально точно відобразити маску об'єкта. Сюди входять такі функції, як Dice Loss та IoU Loss, які враховують перетин і об'єднання між передсказаними та істинними масками.

Функції втрат для меж: Оцінюють точність розмежування об'єктів, мінімізуючи відстань або різницю між передбаченою та істинною межею об'єкта.

Pixel-level Loss Functions (Функції втрат на рівні пікселів) Pixel-level loss functions спрямовані на точне класифікування кожного пікселя на зображенні, оцінюючи розбіжність між передбаченими значеннями пікселів і їх фактичними мітками (ground truth). Ці функції втрат часто використовуються для задач, де важлива точність на рівні окремих пікселів.

Крос-ентропія (Cross-Entropy Loss) Крос-ентропія — це стандартна функція втрат, яка використовується для вимірювання відмінності між ймовірнісними розподілами передбачених класів і істинними класами для кожного пікселя. Вона є найпоширенішою при роботі з задачами семантичної сегментації. Формула для крос-ентропії виглядає так:

$$L_{CE}(y,t) = -\sum_{n=1}^{N} \log\left(t_n \cdot y_n\right) \tag{2.1}$$

де yn — це передбачена ймовірність для класу пікселя n, а tn — це мітка цього пікселя (вектор one-hot encoding).

У випадку, коли дані сильно незбалансовані, можна використовувати зважену крос-ентропію (Weighted Cross-Entropy), де для кожного класу призначаються різні ваги, залежно від того, наскільки часто цей клас зустрічається у даних. Вага для кожного класу зазвичай визначається через обернену частоту класу в наборі даних:

$$L_{WCE}(y,t,w) = -\sum_{n=1}^{N} t_n \cdot w\log(t_n \cdot y_n) \tag{2.2}$$

де w — це вага для кожного класу.

Фокальна втрата (Focal Loss) Фокальна втрата є модифікацією крос-ентропії, яка була запропонована для боротьби з класовим дисбалансом. Її мета — зменшити вплив простих для класифікації прикладів (які модель правильно класифікує з високою ймовірністю) і зосередити увагу на складних (погано класифікованих) прикладах. Формула фокальної втрати виглядає так:

$$L_{focal}(y,t,\gamma) = -\sum_{n=1}^{N} (1 - t_n \cdot y_n)^{\gamma} \log(t_n \cdot y_n) \tag{2.3}$$

де γ — це гіперпараметр, який визначає, наскільки сильно зменшується вплив простих прикладів. Якщо γ = 0, функція втрат перетворюється на стандартну крос-ентропію.

Фокальна втрата дозволяє зменшити внесок простих класів (які моделі добре класифікують) і збільшити важливість тих, що важко класифікуються. Це дозволяє моделі краще справлятися з малопредставленими класами.

Region-level Loss Functions (Функції втрат на рівні регіонів). Region-level loss functions орієнтовані на загальну точність сегментації об'єктів, а не на класифікацію кожного пікселя. Вони намагаються забезпечити точну відповідність між маскою передбаченої сегментації та реальною маскою, орієнтуючись на об'єкти, а не на окремі пікселі.

Функція втрат Дайса (Dice Loss). Функція втрат Дайса виникає з коефіцієнта подібності Дайса (Dice Coefficient), який вимірює схожість між двома множинами (в даному випадку, передбаченою та реальною сегментацією). Ця функція корисна при роботі з незбалансованими наборами даних, оскільки надає великий вплив на правильне розпізнавання невеликих об'єктів або класів, які рідко зустрічаються.

Формула для функції втрат Дайса:

$$L_{dice} = 1 - \frac{1}{C}\sum_{c=0}^{C-1}\frac{2\sum_{n=1}^{N} t_n^c y_n^c}{\sum_{n=1}^{N}(t_n^c + y_n^c)} \quad (2.4)$$

де tn — це мітка пікселя, а yn — це передбачення для кожного пікселя. Ця функція зазвичай використовується для вимірювання якості сегментації, орієнтуючись на загальну схожість між передбаченою та реальною масками.

Лог-косинусова функція втрат Дайса (Log-Cosh Dice Loss). Лог-косинусова функція втрат Дайса (Log-Cosh Dice Loss) є модифікацією стандартної функції втрат Дайса, яка використовує гіперболічну косинусову функцію для зменшення чутливості до аномальних чи неточних даних (шумових точок). Це забезпечує більшу стабільність в оптимізації та зменшує вплив викидів або неправильної аннотації в даних.

Формула для Log-Cosh Dice Loss:

$$L_{lc-dice} = \log\big(\cosh(DiceLoss)\big) \quad (2.5)$$

Де $\cosh(x) = (e^{x} - e^{-x})/2$.Ця функція допомагає створити більш стійке навчання за рахунок зменшення впливу шуму та нерівномірних даних.

Boundary-level Loss Functions (Функції втрат на межовому рівні). Boundary-level loss functions орієнтовані на точність сегментації меж об'єктів. Вони фокусуються на мінімізації відстані між передбаченими межами об'єкта та істинними межами, що дозволяє покращити точність сегментації в складних випадках, коли об'єкти перекриваються або мають неправильні межі.

IoU (Intersection over Union) Loss

Функція втрат IoU, або Жаккара (Jaccard), використовується для вимірювання схожості між передбаченими та реальними масками на основі метрики "перетину через об'єднання" (IoU). Вона визначається як розмір перетину передбаченої та істинної маски, поділений на їх об'єднання. Чим більший цей показник, тим краща сегментація.

Формула для IoU loss:

$$L_{IoU} = 1 - \frac{1}{C}\sum_{c=0}^{C-1}\frac{\sum_{n=1}^{N} t_n^c y_n^c}{\sum_{n=1}^{N}(t_n^c + y_n^c - t_n^c y_n^c)} \quad (2.6)$$

де tn — мітка пікселя, а yn — передбачення пікселя. Ця функція дуже ефективна для оцінки якості сегментації, особливо коли об'єкти частково перекриваються або мають нечіткі межі.

Втрата Тверського (Tversky Loss). Втрата Тверського походить від індексу Тверського (Tversky index) [28], який є асиметричною мірою схожості між двома наборами даних. Це узагальнення коефіцієнта Дайса (Dice coefficient) та показника IoU (Intersection over Union), яке дозволяє вагувати хибні позитиви (false positives) і хибні негативи (false negatives) незалежно. Він визначається наступним чином:

$$\text{Tversky index} = \frac{|Y \cap T|}{|Y \cap T| + \alpha|Y \setminus T| + \beta|Y \setminus T|} \tag{2.7}$$

де α\alphaα та β\betaβ — це ваги для хибних негативів і хибних позитивів відповідно. Якщо α=β=0.5\alpha = \beta = 0.5α=β=0.5, то індекс Тверського зводиться до коефіцієнта Дайса. Якщо α=β=1\alpha = \beta = 1α=β=1, то він зводиться до показника IoU.

Натхненний індексом Тверського, було запропоновано функцію втрат Тверського (Tversky Loss). Функція втрат Тверського дозволяє ефективно налаштовувати модель для задач з сильним класовим дисбалансом, оскільки дає змогу змінювати ваги для помилок різних типів (хибних позитивів і хибних негативів). Це особливо корисно при роботі з малопредставленими класами, коли одна з типових помилок (наприклад, хибні негативи) має бути знижена в порівнянні з іншими.

РОЗДІЛ 3. ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

3.1 Створення набору даних

Для створення набору даних для семантичної сегментації жилкування листя було зібрано кілька сотень зображень листя з різних куточків і типів рослин. Зображення були зібрані в різних умовах освітлення та при різних ракурсах, щоб забезпечити різноманітність зразків для навчання нейромережі.

Кожне зображення було вручну розмічене для виявлення жилкування на листі. Для кожного зображення створювалася маска, що позначала пікселі, які належать жилкуванню. Це було важливо для подальшого навчання нейромережі, оскільки модель повинна була навчитися визначати, які пікселі на зображенні належать до жилкування листя.

3.2 Підготовка даних

Після збору та розмітки даних, вони були підготовлені для навчання. Це включало нормалізацію пікселів зображень до діапазону [0, 1], що є стандартною практикою для згорткових нейромереж. Також була застосована аугментація даних, яка включала:

-Обертання зображень на різні кути (напр. 90°, 135°)

-Зміна яскравості та контрастності

-Збільшення та зменшення зображень

-Горизонтальне та вертикальне віддзеркалення

Ця методика дозволила створити більший та більш різноманітний набір для навчання моделі.

3.3 Архітектура згорткової нейромережі

Для сегментації малюнку жилкування листя була обрана класична архітектура нейромережі UNet. В оригінальній архітектурі UNet є дві основні частини: енкодер, що складається з декількох блоків згорток і пулингу, та декодер, який за допомогою зворотних згорток відновлює зображення до вихідної роздільної здатності.

У нашій роботі ми модифікували стандартну архітектуру UNet, змінивши кількість етапів у енкодері та кількість фільтрів на кожному етапі, щоб адаптувати мережу до специфічних особливостей задачі сегментації жилкування листя. Ми також застосували спеціалізовані функції втрат для покращення результатів на незбалансованих класах (наприклад, wDice, wIoU), що зменшують вплив фону на результати сегментації.

3.4 Критерії якості моделі

Основними функціями втрат для цієї задачі були:

-wDice – модифікована функція втрат для покращення якості сегментації у разі незбалансованих класів.

-wiou – функція втрат, яка оцінює співвідношення інтерсекции та об'єднання (IoU) для кожного класу.

-wdice2 – ще одна модифікація функції втрат, яка поєднує ваги для кожного класу в обчисленнях Dice.

Ці функції були вибрані через їх здатність добре працювати на датасетах з великим дисбалансом класів. Також тестувались різні метрики оцінки:

-Dice: коефіцієнт подібності між розпізнаними об'єктами та реальними масками.

-IoU (Intersection over Union): вимірює точність сегментації, оцінюючи співвідношення між перетином і об'єднанням двох областей (прогнозованої та істинної маски).

-Accuracy: загальна точність сегментації по всьому зображенню.

Проблема незбалансованих класів. Однією з основних проблем у задачі сегментації жилкування листя є значний дисбаланс класів: жилкування займає набагато меншу частину зображення порівняно з фоном. Це призводить до того, що модель часто неправильно класифікує пікселі як фон. Для вирішення цієї проблеми були використані функції втрат, які враховують ваги класів, що дозволяє моделі приділяти більше уваги менш представленим класам (жилки).

3.5 Функція втрат

Таблиця 3.1. Оцінка якості нейромережі UNet з різними варіантами функції втрат та розміру пакетів (batch size).

Loss function	Optimizer	Learning rate	Epoch	Batch size	train dataset				validation dataset			
					Loss	Dice	IoU	Accuracy	loss	dice	IOU	accuracy
wdice	SGD	0.01	440	8	0.2018	0.9841	0.9688	0.9842	0.5121	0.9596	0.9225	0.9597
wdice	SGD	0.01	400	2	0.1664	0.9904	0.981	0.9904	0.4749	0.9656	0.9336	0.9656
wdice2	SGD	0.01	400	2	1.	0.0516	0.0265	0.0516	0.9831	0.0346	0.0176	0.0346
wdice2	SGD	0.01	400	8	1.	0.7077	0.549	0.9304	0.9994	0.7125	0.5554	0.931
wtversky	SGD	0.01	270	8	0.9514	0.631	0.4628	0.6328	0.9514	0.6368	0.4698	0.6385
wiou	SGD	0.01	600	8	0.3566	0.9785	0.9579	0.9786	0.6677	0.9586	0.9207	0.9587
wiou2	SGD	0.01	560	8	1.	0.7027	0.543	0.9305	0.9997	0.7074	0.5491	0.931
wcoshdice2	SGD	0.02	600	8	0.0558	0.9744	0.9502	0.9747	0.1588	0.956	0.9158	0.9562

При навчанні використовувався оптимізатор SGD із швидкістю навчання 0.01 і розміром пакету 2. Як видно з таблиці, функція втрат wdice дала найкращі результати для тренувального та валідаційного наборів, з високими значеннями метрик Dice та IoU.

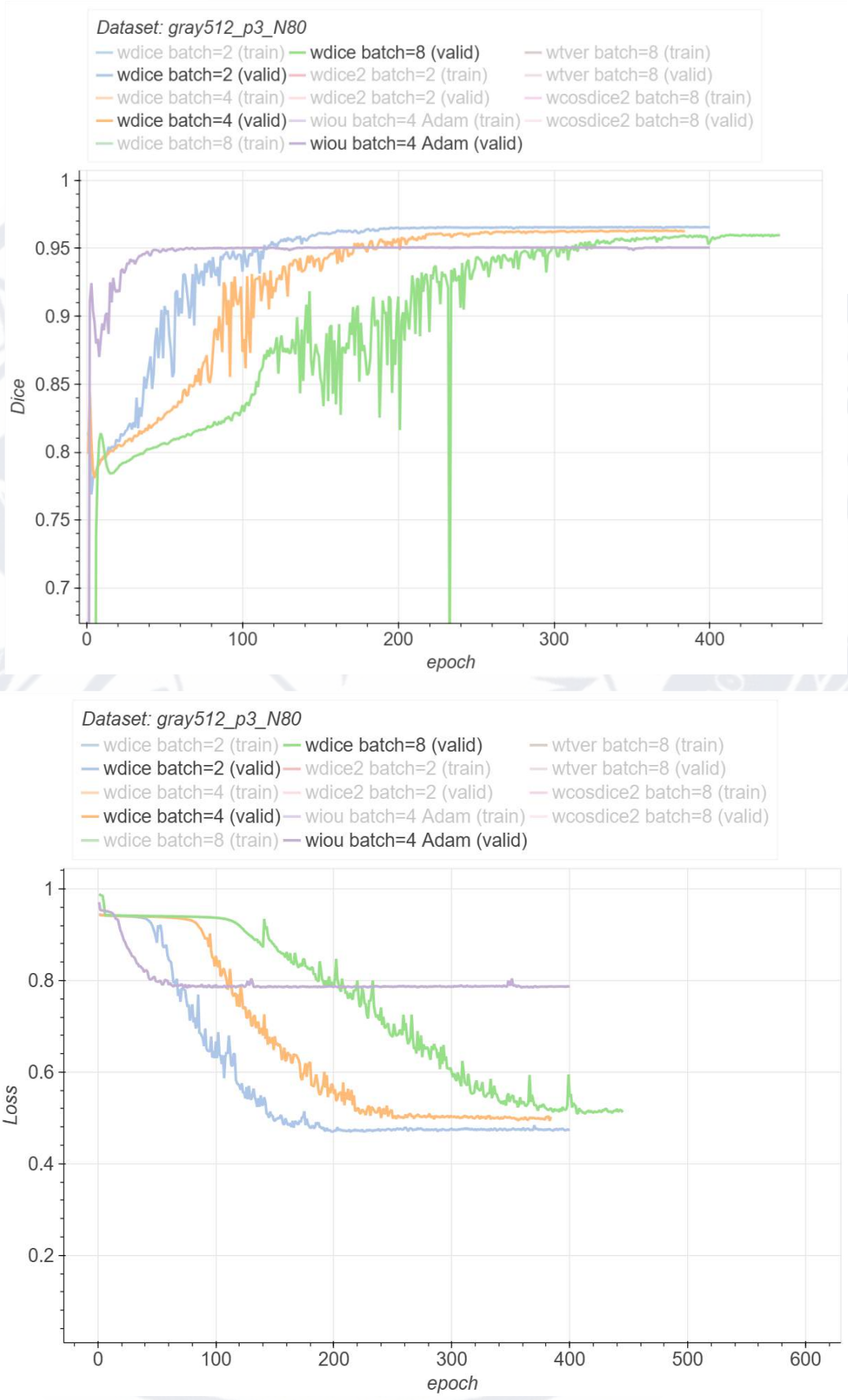


Рис. 3.1 Динаміка процесу навчання нейромережі UNet для різних наборів даних.

На графіках показана динаміка функції втрат та метрик якості навчання. Для тренувального набору даних, при збільшенні номера ітерації (епохи), функція втрат зазвичай зменшується, а метрики якості навчання, в свою чергу, збільшуються. Мінімальною величиною для функції втрат є нуль, а для метрик навчання — одиниця. Зменшення величини функції втрат є результатом оптимізації вагових коефіцієнтів нейромережі за допомогою алгоритму градієнтного спуску, який намагається знайти мінімум функції втрат.

З часом обидві величини виходять на плато, тобто значення функції втрат та метрик якості не змінюються суттєво. Функція втрат та метрики якості зазвичай поводять себе схожим чином, але є одна важлива відмінність: плато метрики навчання має меншу величину, тоді як функція втрат — більшу. Така ситуація часто є ознакою перенавчання (overfitting), що свідчить про те, що нейромережа надмірно адаптувалася до тренувального набору даних. Це може бути результатом того, що нейромережа "запам'ятала" специфіку тренувального набору, але не змогла узагальнити закономірності, що можуть бути застосовані до нових, невідомих даних.

Перенавчання — це явище, коли модель показує відмінні результати на тренувальних даних, але погано працює на нових, не бачених раніше даних. Для того, щоб уникнути перенавчання, можна застосувати кілька стратегій: передчасну зупинку навчання моделі, коли функція втрат перестає значно зменшуватися на валідаційному наборі даних; зменшення складності нейромережі, наприклад, за допомогою зменшення кількості параметрів або шарів, що допомагає уникнути перенавчання; а також збільшення розміру набору даних, що ускладнює "запам'ятовування" специфічних деталей і дозволяє моделі узагальнити закономірності.

Найкращі результати були досягнуті за допомогою функцій втрат **wdice** та **wiou**. Ці функції показали найкращі результати, оскільки вони є менш чутливими до дисбалансу класів у даних, що дуже важливо при роботі з даними, де один клас (наприклад, вени) значно менший за інші. Функції втрат, які показали гірші результати, це **wtver**, **wcosdice2** та **wdice2**. Ці функції не змогли забезпечити таку

стабільну і точну оптимізацію як перші дві, що може бути пов'язано з їхньою більшою чутливістю до дисбалансу класів або іншими особливостями їхньої структури.

У більшості випадків вени були класифіковані неправильно, або як частина листка, або як фон. Для зменшених зображень (UNet-MobilNetV 2) спостерігався зворотний ефект, коли пікселі листка були класифіковані як вени, але загалом розмір зображення не мав суттєвого впливу на результати. Для полегшення цієї проблеми були введені ваги класів, обернено пропорційні розміру класу.

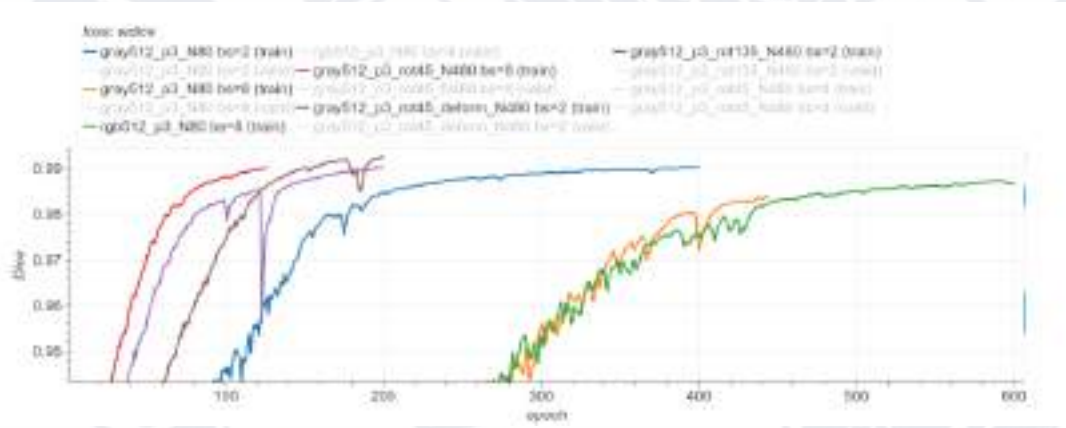


Рис.3.2 Приклади сегментації нейромережі з функцією втрат wdice, зображень з тренувального набору

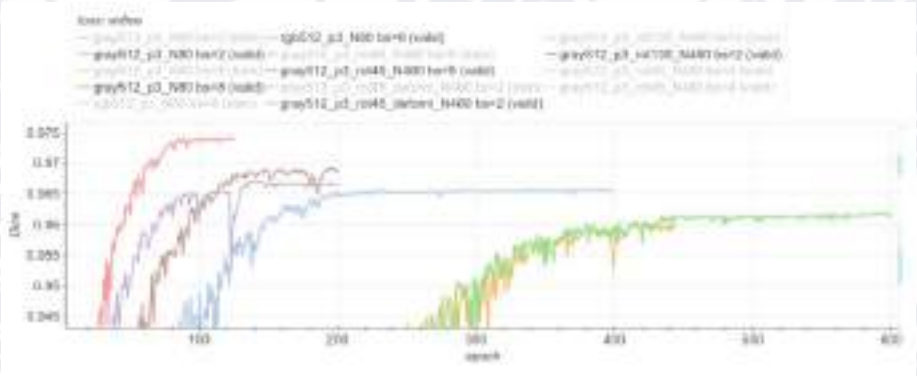


Рис.3.3 Приклади сегментації нейромережі з функцією втрат wdice, зображень з валідаційного набору

Як показує таблиця 3.1, зменшення розміру пакету до 2 дозволило досягти кращих результатів для валідаційного набору при однакових параметрах навчання.

3.6 Набір даних

Таблиця 4.2. Оцінка якості нейромережі UNet для різних наборів даних.

					train dataset				validation dataset			
Dataset name	Optimizer	earning rat	Epoch	Batch size	Loss	Dice	IoU	Accuracy	loss	dice	IOU	accuracy
gray512_p3_N80	SGD	0.01	440	8	0.2018	0.9841	0.9688	0.9842	0.5121	0.9596	0.9225	0.9597
rgb512_p3_N80	SGD	0.01	440	8	0.2338	0.9798	0.9604	0.98	0.4936	0.9607	0.9244	0.9608
gray512_p3_rot135_N480	SGD	0.01	200	2	0.041	0.9929	0.9859	0.9929	0.3198	0.9691	0.9403	0.9691

3.7 Оптимізація архітектури моделі

Таблиця 4.3. Оцінка якості нейромережі UNet при модифікаціях її архітектури.

Number of stages	Number of filters	Layers	Optimizer	Learning rate	Epoch	Batch size	train dataset				validation dataset			
							Loss	Dice	IoU	Accuracy	Loss	Dice	IOU	Accuracy
3	64	2	SGD	0.01	440	8	0.2018	0.9841	0.9688	0.9842	0.5121	0.9596	0.9225	0.9597
3	64	2	SGD	0.01	200	2	0.9244	0.8836	0.7923	0.8962	0.9235	0.888	0.7991	0.8993
4	16	2	SGD	0.01	200	2	0.2502	0.9733	0.948	0.9733	0.3998	0.9615	0.926	0.9615
4	64	1	SGD	0.01	200	2	0.116	0.9871	0.9745	0.9871	0.3664	0.9643	0.9314	0.9644
3	64	1	SGD	0.01	200	2	0.111	0.9854	0.9714	0.9855	0.4215	0.9596	0.9228	0.9596
4	8	2	SGD	0.01	600	2	0.2582	0.9808	0.9624	0.9808	0.4174	0.9659	0.9343	0.9659
4	8	2	SGD	0.01	600	8	0.1811	0.9787	0.9584	0.9787	0.3985	0.9599	0.9233	0.9599
4	16	2	SGD	0.01	600	2	0.2746	0.9815	0.9637	0.9815	0.4947	0.96	0.9234	0.96
4	16	2	SGD	0.01	600	8	0.2777	0.9794	0.9597	0.9794	0.4693	0.9612	0.9256	0.9612
3	8	2	SGD	0.01	1000	8	0.3406	0.9677	0.9377	0.9677	0.5675	0.9503	0.9057	0.9503
5	8	2	SGD	0.01	1000	2	0.2661	0.9749	0.951	0.9749	0.3889	0.9639	0.9306	0.9639

Модифікація архітектури нейромережі, включаючи зміну кількості етапів енкодера та кількості фільтрів, показала значні зміни в якості сегментації.

Нейромережа UNet складається з двох основних гілок — енкодера та декодера. Вхідне зображення спочатку потрапляє на вхід енкодера, де воно проходить через кілька етапів перетворення. Кожен етап перетворення реалізується стеком, що складається з декількох згорткових шарів та шару пулінга (Max Pooling). Після кожного етапу просторові розміри зображення зменшуються вдвічі, однак кількість каналів збільшується вдвічі, що дає змогу зберігати більше інформації на кожному етапі. Такі трансформовані зображення на виході кожного етапу називають картами активації. Таким чином, відбувається зменшення просторової розмірності зображення, але збільшення глибини. У гілці декодера, в свою чергу, відбувається зворотний процес, де розміри зображення знову збільшуються.

В класичному варіанті архітектури UNet є чотири етапи в кожній з гілок — енкодері та декодері. На кожному етапі міститься два згорткові шари. На першому етапі енкодера згорткові шари мають 64 фільтри, що дозволяє створювати 64 карти активації на виході. На наступних етапах кількість фільтрів подвоюється, що дозволяє моделі зберігати більшу кількість характеристик зображення на більш глибоких етапах.

У нашій роботі ми намагались змінити архітектуру нейромережі і дослідити, як ці зміни впливають на якість навчання. Мотивація до таких змін була двоякою: з одного боку, ми прагнули спростити архітектуру для зменшення обчислювальних ресурсів, зниження вимог до пам'яті та прискорення процесу обчислень, з іншого — ускладнення архітектури, якщо це необхідно для підвищення точності моделі в складних задачах.

Для спрощення архітектури нейромережі ми передбачали, що зменшення кількості параметрів та обчислювальних операцій може знизити ризик перенавчання та прискорити навчання. У той же час, у разі недостатньої складності архітектури, нейромережа може не мати необхідного потенціалу для успішного розв'язання задачі. Тому ми також експериментували з ускладненням архітектури для покращення якості навчання.

Наші спроби змінити архітектуру UNet включали кілька варіантів конфігурацій. Ми змінювали кількість стадій енкодера та декодера, тестуючи варіанти з трьома, чотирма та п’ятьма стадіями. Крім того, на кожній стадії енкодера та декодера варіювалася кількість згорткових шарів — один чи два. Також ми випробували різну кількість фільтрів на першій стадії енкодера, тестуючи значення 8, 16 та 64 фільтри.

Отримані результати дозволяють класифікувати варіанти архітектури від найкращих до найгірших. Найкращі результати були отримані при використанні п’яти стадій в енкодері та декодері, двох згорткових шарів на кожному етапі та 64 фільтрів на першій стадії. Ця конфігурація забезпечила високі показники точності та стабільності моделі. Варіанти з меншою кількістю стадій (три або чотири), меншими кількостями фільтрів та згорткових шарів показали значно гірші результати, що підтверджує необхідність більш складних архітектур для досягнення високої якості навчання в задачах сегментації зображень. Всі отримані результати можна ранжувати від найкращих до найгірших наступним чином:

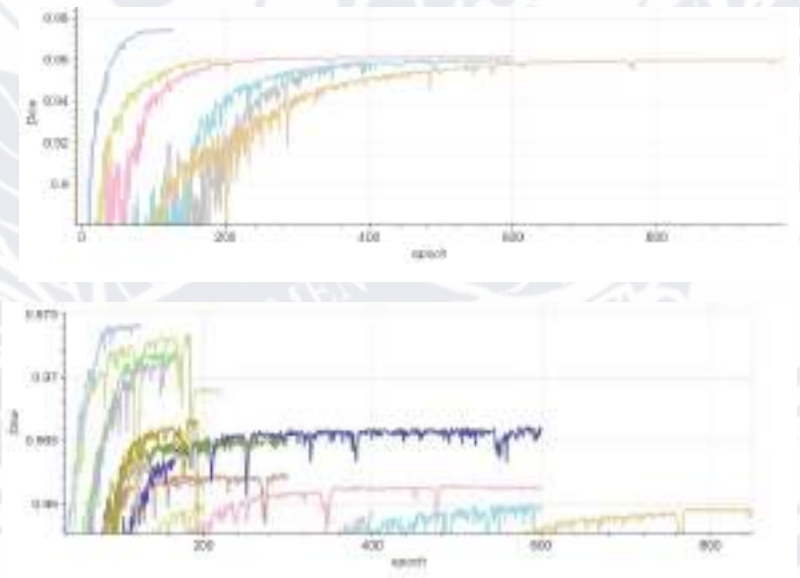


Рис.3.4 Динаміка процесу навчання нейромережі UNet для різних варіантів архітектури. Функція втрат та метрики оцінки якості навчання залежно від епохи.

Найкращими тестованими моделями були:

- dice 0.97-0.975:
  - 'stg=[4] filt=[64] cnv=[2] bs=8'
  - 'stg=5 filt=16 cnv=[2] bs=2'
  - 'stg=[4] filt=16 cnv=[2] bs=2'
  - 'stg=[4] filt=16 cnv=[2] bs=2 rep c'
- dice 0.965:
  - 'stg=5 filt=16 cnv=[2] bs=4'
  - 'stg=5 filt=8 cnv=[2] bs=2'
- dice 0.962:
  - 'stg=[4] filt=16 cnv=[2] bs=8'
  - 'stg=3 filt=[64] cnv=1 bs=2'
  - 'stg=[4] filt=8 cnv=[2] bs=2'
  - 'stg=[4] filt=16 cnv=[2] bs=2 sgd.001'
  - 'stg=[4] filt=16 cnv=[2] bs=16'
- dice 0.95:
  - 'stg=3 filt=8 cnv=[2] bs=8'
- dice 0.92 (steady state):
  - 'stg=3 filt=[64] cnv=[2] bs=2'

Як зміни параметрів архітектури впливають на якість навчання:

1. Кількість стадій енкодера/декодера:
   - Зменшення кількості стадій значно погіршувало якість навчання (метрика dice знижувалася). Це свідчить про те, що менш глибока мережа не здатна адекватно витягнути всі необхідні ознаки з даних.
   - Збільшення кількості стадій мало незначний вплив на якість, або навіть дещо знижувало точність. Тобто, кількість стадій більше ніж 4 або 5 не завжди приносила покращення.

   Параметри, які тестувались:
   - dice 0.97-0.975:
     - 'stg=[4] filt=[64] cnv=[2] bs=8'

- dice 0.95:
  - 'stg=3 filt=8 cnv=[2] bs=8'
- dice 0.92 (steady state):
  - 'stg=3 filt=[64] cnv=[2] bs=2'
- dice 0.965:
  - 'stg=5 filt=16 cnv=[2] bs=4'

2. Кількість згорткових шарів на кожній стадії енкодера/декодера:
   - Зменшення кількості згорткових шарів на кожній стадії погіршувало якість навчання. Використання більшої кількості згорткових шарів допомагає мережі краще витягувати ознаки з зображень, що сприяє покращенню результатів.

   Параметри, які тестувались:
   - dice 0.962:
     - 'stg=3 filt=[64] cnv=1 bs=2' (1 згортковий шар на стадію)
   - dice 0.97-0.975:
     - 'stg=[4] filt=[64] cnv=[2] bs=8'
     - 'stg=[4] filt=[16] cnv=[2] bs=2'
3. Кількість фільтрів на першій стадії енкодера:
   - Зменшення кількості фільтрів на першій стадії до 16 не вплинуло суттєво на якість навчання. Однак подальше зменшення до 8 фільтрів вже знижувало точність, оскільки мережа починала недоцільно зменшувати кількість витягнутих ознак.

   Параметри, які тестувались:
   - dice 0.97-0.975:
     - 'stg=[4] filt=16 cnv=[2] bs=2'
     - 'stg=[4] filt=16 cnv=[2] bs=2 rep c'
   - dice 0.962:
     - 'stg=[4] filt=8 cnv=[2] bs=2'

3.8 Регуляризація

У процесі навчання нейромережі для сегментації жилкування листя важливо уникати перенавчання (overfitting), коли модель "запам'ятовує" тренувальний набір і не може адекватно працювати з новими даними. Для цього використовувалися різні методи регуляризації, зокрема:

- Dropout: Цей метод допомагає уникнути перенавчання, випадковим чином вимикаючи деякі нейрони під час кожної ітерації навчання. Це дозволяє мережі навчатися більш узагальнено і не прив'язуватися до окремих деталей тренувальних даних.
- L2 регуляризація (Weight Decay): Використовувалися штрафи на великі значення ваг, що також допомагає уникати перенавчання. В результаті нейромережа більш рівномірно розподіляє важливість між різними характеристиками зображення, замість того щоб переоцінювати важливість окремих вхідних ознак.
- Data Augmentation: Як зазначалося раніше, аугментація даних є важливим методом регуляризації. Вона дозволяє моделі побачити різноманітні варіації зображень, що підвищує її здатність до узагальнення на нових, раніше не бачених зображеннях.

Експерименти з різними методами регуляризації показали, що поєднання Dropout та L2 регуляризації забезпечує найкращі результати в плані зниження перенавчання. Також варто зазначити, що збільшення набору даних завдяки аугментації значно покращило здатність моделі до узагальнення на нові зображення.

3.9 Кількість класів, об'єднання класів

В процесі сегментації було використано два основних типи масок для сегментації зображень: бінарна сегментація та мультикласова сегментація.

У процесі навчання нейронної мережі для сегментації зображень ми використовували оптимальні параметри, отримані раніше в експериментах. Далі ми порівняли результати сегментації, які були отримані моделлю, з істинними масками (ground-truth mask) для різних наборів зображень, зокрема з тестового та валідаційного наборів. Зокрема, ми здійснили порівняння прогнозованих масок (prediction mask) з цільовими істинними масками, що є зразками правильних позначок для кожного зображення.

У контексті сегментації зображень можна виділити кілька важливих типів задач:

1. Multi-class segmentation (сегментація з кількома класами) – це задача, коли зображення містить кілька різних об'єктів, і кожен піксель може належати одному з кількох класів. Важливим є правильне визначення меж кожного класу, оскільки це дозволяє точно виділяти різні об'єкти на зображенні. Зазвичай, для цього використовуються техніки, які дозволяють відрізняти пікселі, що належать різним категоріям, і класифікувати їх у відповідні класи.
2. Binary Image Segmentation (бінарна сегментація зображень) – це простіший варіант сегментації, де кожен піксель належить до одного з двох класів, наприклад, "об'єкт" чи "фон". Бінарна сегментація зазвичай використовується для задач, де необхідно виділити один конкретний об'єкт, наприклад, в медицині для виявлення пухлин або в агрономії для виділення рослин.
3. Multi-Class Image Segmentation (сегментація з багатьма класами на зображенні) – у цьому випадку кожен піксель на зображенні може бути віднесений до одного з декількох класів, наприклад, "дорога", "будівля", "дерево" тощо. Такий тип сегментації вимагає більш складних моделей і підходів для коректного виділення та класифікації різних об'єктів на зображенні.

Порівняння прогнозованих масок з істинними масками дозволяє оцінити якість роботи моделі в різних умовах і типах сегментації. На Рис. 4.5 та Рис. 4.6

наведені приклади прогнозованих масок разом з відповідними істинними масками, що дає змогу візуально оцінити точність моделі.

Таким чином, аналіз результатів сегментації на основі порівняння масок допомагає не лише виявити точність прогнозів, а й визначити області, де модель може потребувати подальшої оптимізації для покращення результатів.

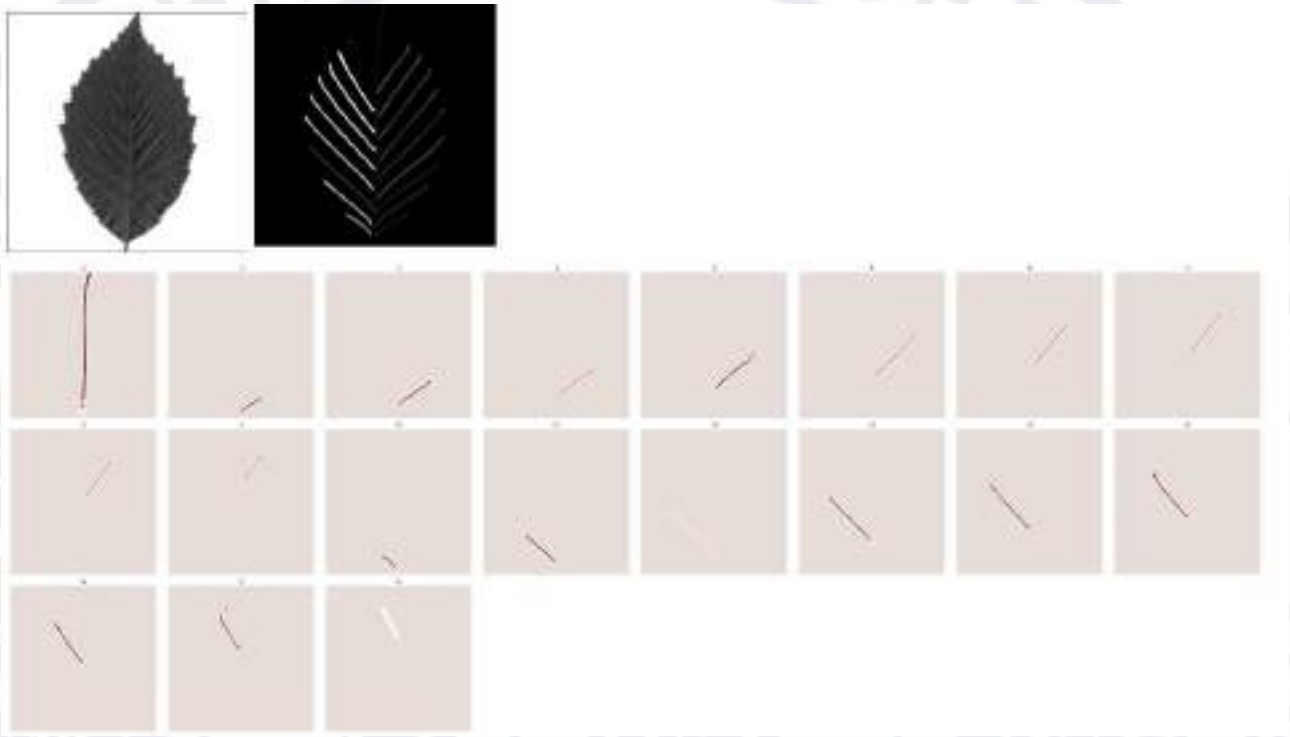


Рис. 3.5 Приклади мультикласової сегментації зображень з тренувального набору.

Використовувалась нейромережа U-Net з оптимальними параметрами яка навчалась класифікувати пікселі на 20 класів. Показані: вихідне зображення; прогнозовані маски накладені одна на одну (пікселі різних класів показані відтінками сірого); окремі прогнозовані маски на які накладені напівпрозорі істини маски.

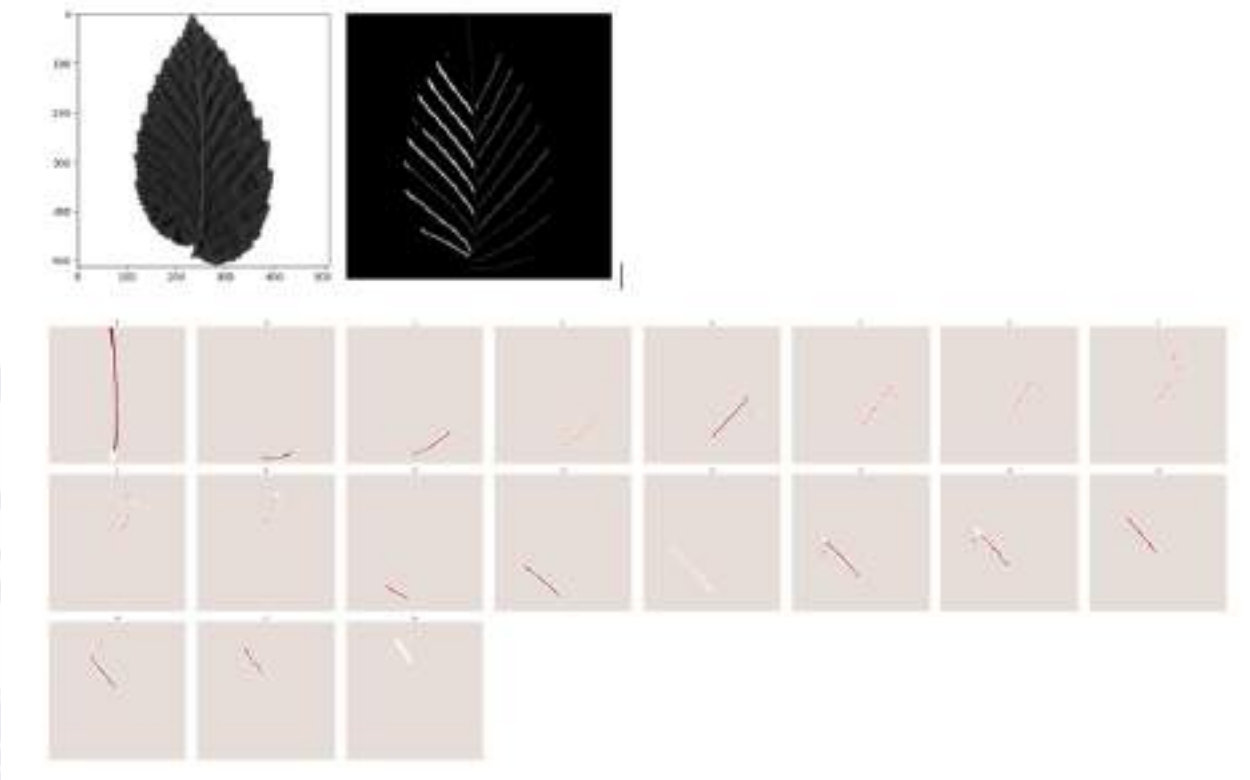


Рис. 3.6 Приклади мультикласової сегментації зображень з валідаційного набору.

Використовувалась нейромережа U-Net з оптимальними параметрами яка навчалась класифікувати пікселі на 20 класів. Показані: вихідне зображення; прогнозовані маски накладені одна на одну (пікселі різних класів показані відтінками сірого); окремі прогнозовані маски на які накладені напівпрозорі істини маски.

Як можна побачити, нейронна мережа досить ефективно розподіляє пікселі між фоном і класами, що належать жилкам. У результаті, структура жилок на зображеннях відтворюється досить точно. Однак, існують істотні помилки при класифікації пікселів, що належать жилкам. Часто нейромережа неправильно класифікує піксель жилки, хибно відносячи його до класу іншої жилки. Це може призводити до ситуацій, коли частина жилки або вся жилка сегментуються геометрично правильно, але її клас визначається невірно.

Такі помилки виникають через те, що нейронна мережа не завжди ефективно враховує контекст. Для коректного визначення класу пікселя жилки необхідно, щоб мережа враховувала його ближнє та дальнє оточення, оскільки

класи жилок можуть бути схожими між собою. Відсутність достатньої уваги до контексту призводить до помилок в класифікації.

Для того щоб подолати ці проблеми, ми вирішили спростити задачу, зменшивши кількість класів, що класифікуються нейронною мережею. Для цього було реалізовано кілька стратегій:

- Обмеження кількості жилок для сегментації: замість сегментування всіх 19 жилок, ми обрали меншу кількість жилок для навчання. Нейронна мережа мала сегментувати лише одну або кілька жилок, що дозволило спростити задачу і зменшити кількість можливих помилок класифікації.

- Об'єднання класів жилок: для спрощення задачі було об'єднано певні класи жилок в один клас. Наприклад, ми об'єднали всі бічні ліві жилки в один клас, що дозволило мережі зосередитися на меншій кількості класів та покращити загальну точність сегментації.

В рамках цих підходів ми дослідили можливості нейронної мережі U-Net для мультикласової сегментації на шести варіантах. Ось вони:

1. Шість класів: центральна жилка (v0), чотири бічні жилки (v5R, v5L, v6L, v6R) та фон. Це дозволяє детально сегментувати кілька жилок та фон.

Рис. 3.7 Приклади мультикласової сегментації зображень з тренувального набору.

Використовувалась нейромережа U-Net з оптимальними параметрами яка навчалась класифікувати пікселі на шість класів:

центральна жилка (v0), чотири бічні (v5R, v5L, v6L, v6R) та фон. Показані: вихідне зображення; прогнозовані маски накладені одна на одну (пікселі різних класів показані відтінками сірого); окремі прогнозовані маски на які накладені напівпрозорі істини маски.

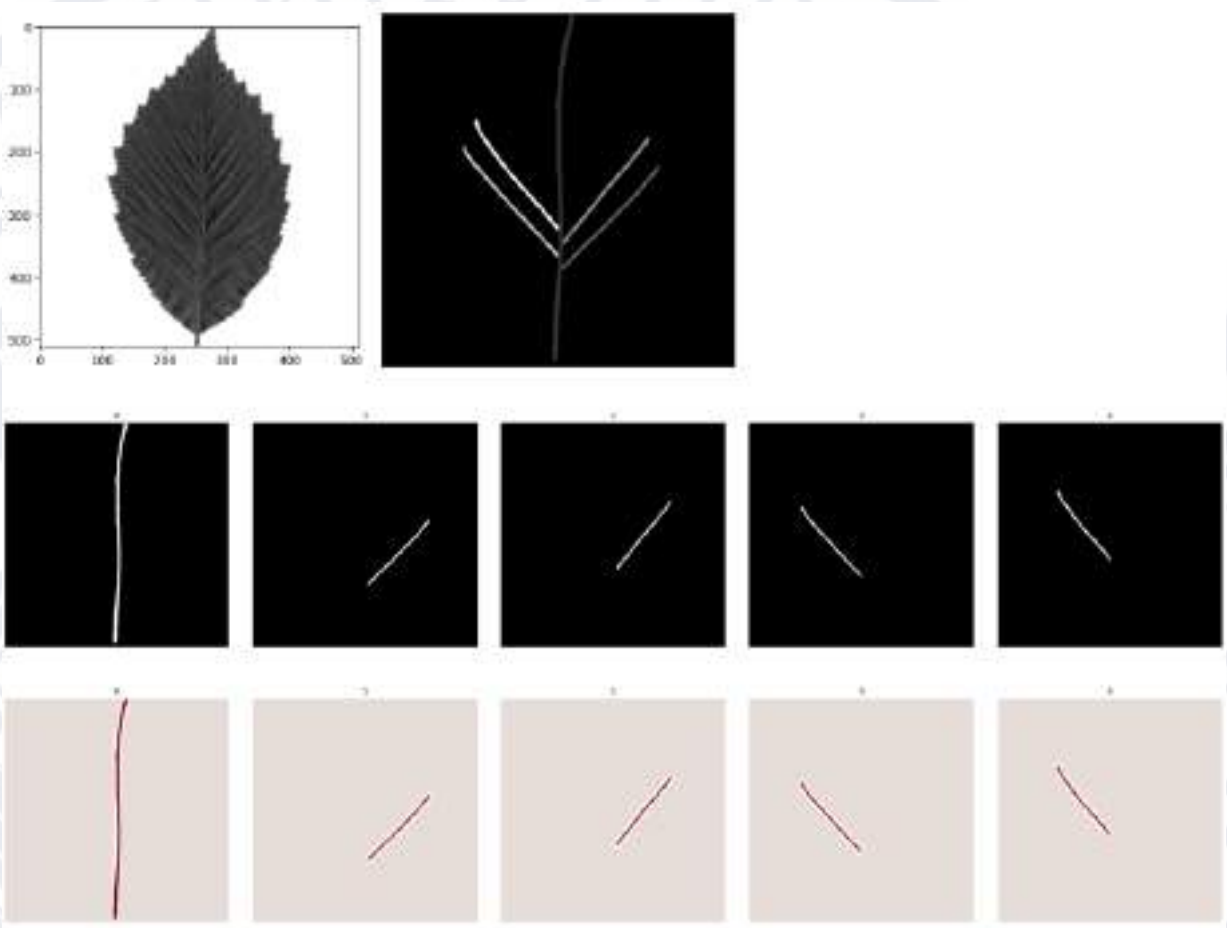


Рис. 3.8 Приклади мультикласової сегментації зображень з валідаційного набору.

Використовувалась нейромережа U-Net з оптимальними параметрами яка навчалась класифікувати пікселі на шість класів: центральна жилка (v0), чотири бічні (v5R, v5L, v6L, v6R) та фон. Показані: вихідне зображення; прогнозовані маски накладені одна на одну (пікселі різних класів показані відтінками сірого); окремі прогнозовані маски на які накладені напівпрозорі істини маски.

Як можна бачити з прикладу наведеного на рис 3.7 для зображень з тренувального набору нейромережа успішно виконує сегментацію, прогнозовані маски збігаються з істинними. Однак процес тренування мав

значне перенавчання і валідаційний набір, зображення якого нейромережа раніше не бачила, сегментується набагато гірше. Як можна бачити на рис 3.8 центральна жилка сегментується вірно, але бічні жилки 5 та 6 сегментовані не повністю, а також на додачу до них нейромережа сегментує і інші жилки. Це відбувається завдяки тому, що нейромережа не вивчила глобальної інформації необхідної для правильного визначення класу пікселів даної бічної жилки. Пікселі центральної жилки, що знаходиться по середині листка, класифікувати легше оскільки для цього не потрібно враховувати взаємне розташування інших жилок, нейромережа використовує для цього якісь інші ознаки.

2. Чотири класи: центральна жилка (v0), дві бічні жилки (v5R, v5L) та фон. Цей варіант спрощує задачу, зменшуючи кількість класів жилок для класифікації.

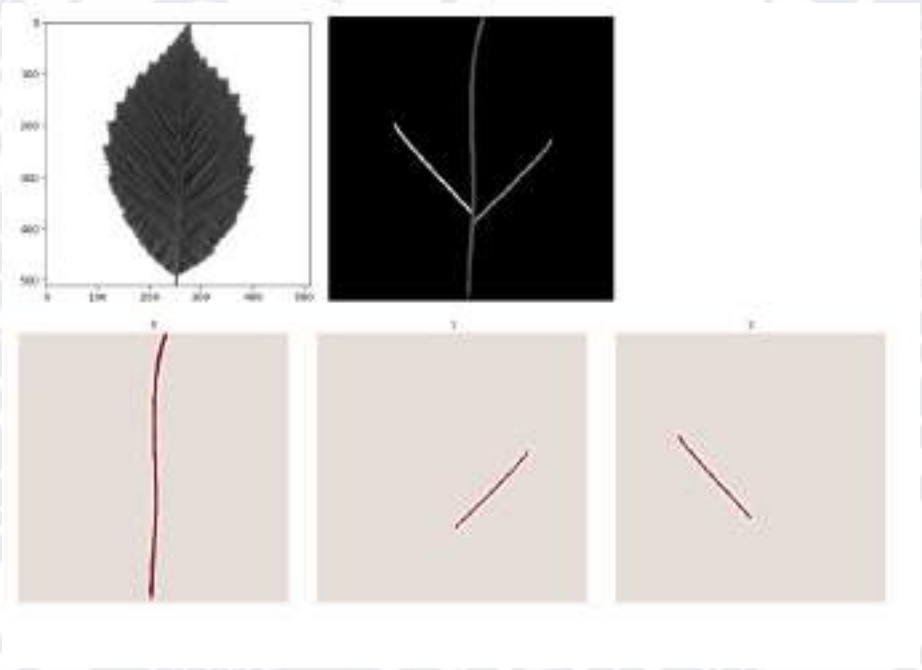


Рис. 3.9 Приклади мультикласової сегментації зображень з тренувального набору.

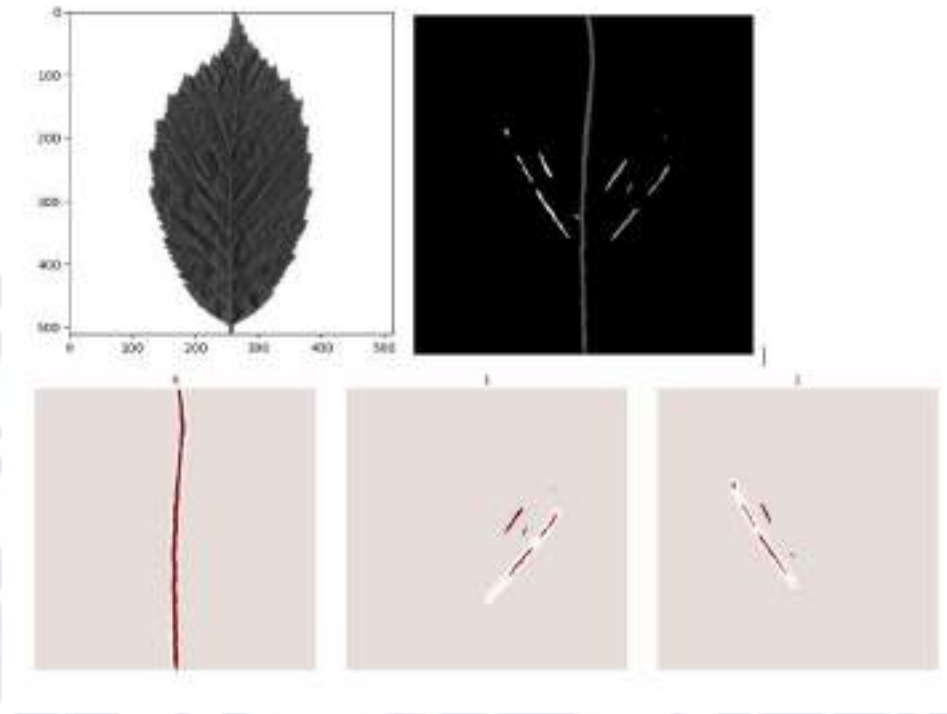


Рис. 3.10 Приклади мультикласової сегментації зображень з валідаційного набору.

Не дивлячись на те, що кількість класів було зменшено, ситуація аналогічна попередній.

3. Два класи: центральна жилка (v0) та фон. Це ще більш спрощена задача, де лише одна жилка і фон підлягають сегментації.

Рис. 3.11 Приклади мультикласової сегментації зображень з валідаційного набору.

На цей раз і тренувальний і валідаційний набір показав себе однаково. Нейромережа має певні помилки з класифікації жилок. На Рис. 3.11 видно невеличку помилку у визначенні центральної жилки з правого

боку. Хоча усю область де має бути розташована центральна жилка визначенно правильно.

4. Два класи: одна бічна жилка (v5L) та фон. Цей варіант ще більше обмежує кількість сегментованих класів, що дозволяє мережі фокусуватися на окремих бічних жилках.

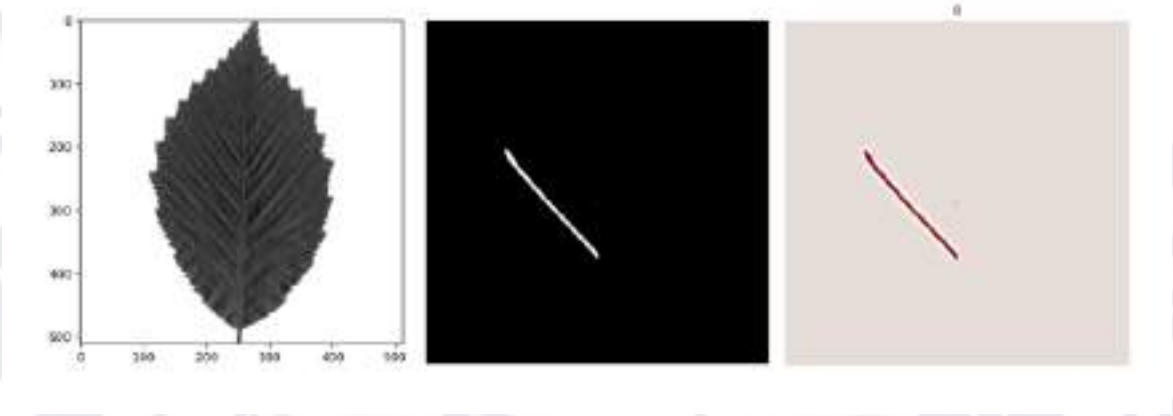


Рис. 3.12 Приклади мультикласової сегментації зображень з тренувального набору.

Рис. 3.13 Приклади мультикласової сегментації зображень з валідаційного набору.

Як ми бачимо по найтиповішим представиникам данного експерименту, є певні ознаки перенавчання хоча і менші ніж в попередній ситуації. Але нейромережа справляється з поставленою задачею та добре класифікує жилки.

5. Три класи: всі бічні жилки (v1L+...+v9L+v1R+...+v9R), центральна жилка (v0) та фон. Тут об'єднуються всі бічні жилки в один клас, що дозволяє зменшити складність класифікації і покращити якість сегментації.

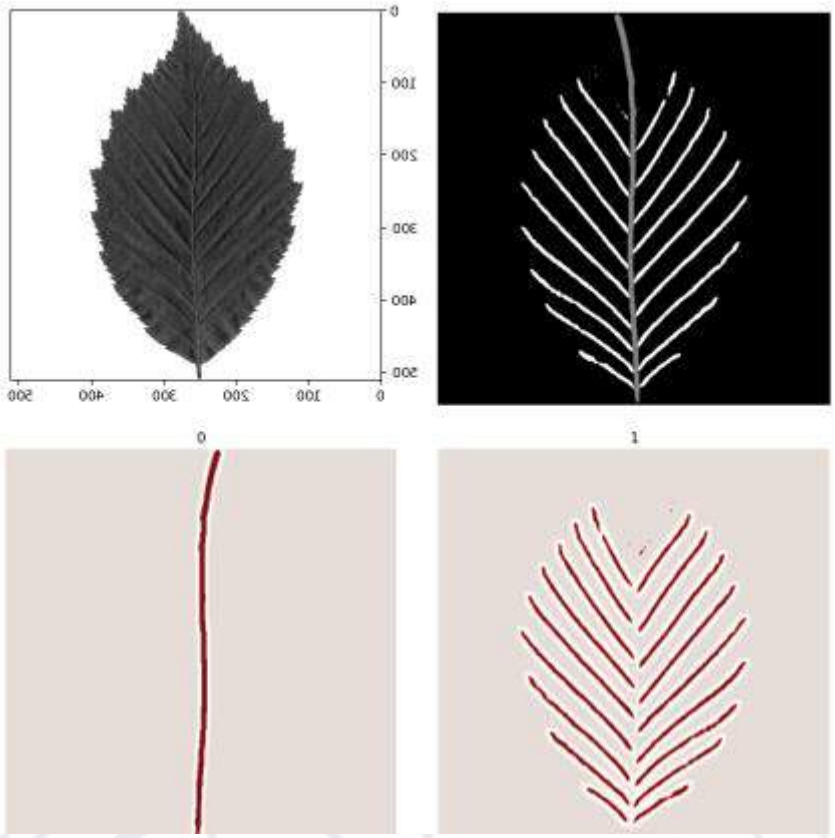


Рис. 3.14 Приклади мультикласової сегментації зображень з тренувального набору.

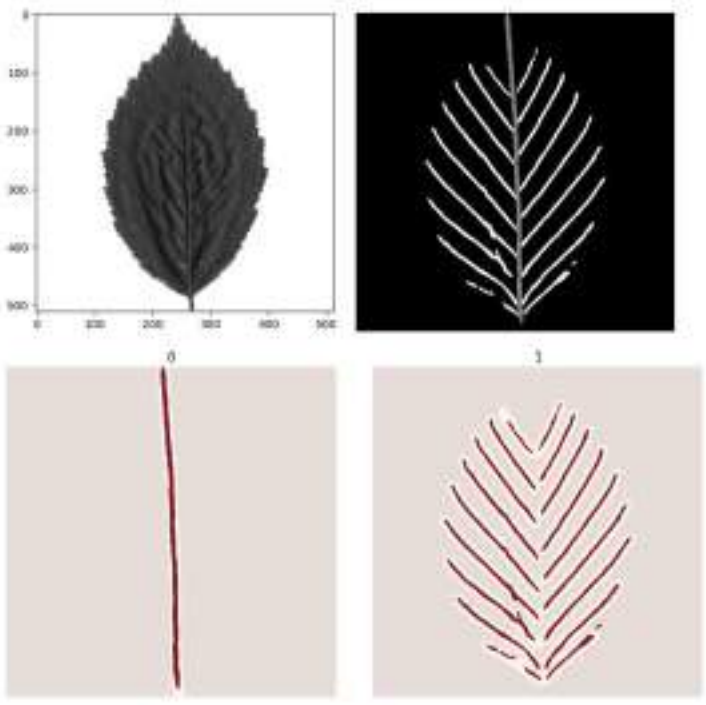


Рис. 3.15 Приклади мультикласової сегментації зображень з валідаційного набору.

Загалом визначення бічних та центральної жилки в обох наборах сильно не відрізняється.

6. Два класи: всі жилки (v1L+...+v9L+v1R+...+v9R+v0) та фон. В цьому варіанті всі жилки об'єднуються в один клас, що дозволяє мережі виконувати сегментацію в значно більшій мірі як бінарну задачу: всі жилки проти фону.

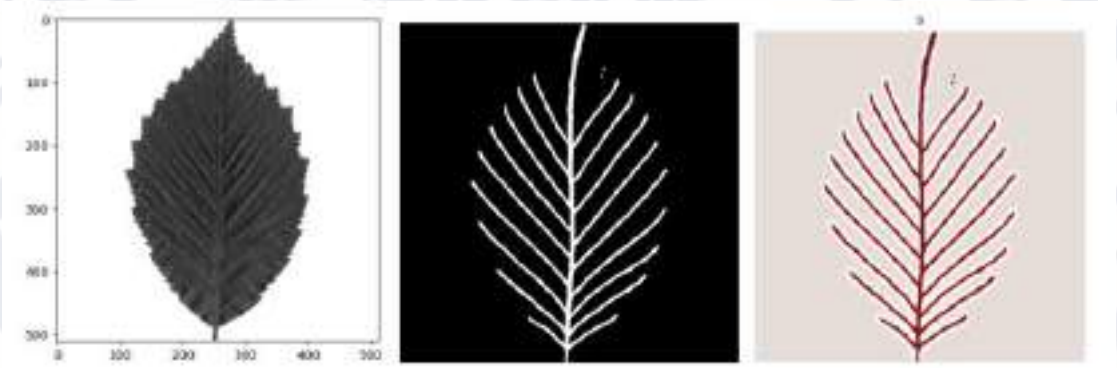


Рис. 3.16 Приклади мультикласової сегментації зображень з тренувального набору.

Рис. 3.17 Приклади мультикласової сегментації зображень з валідаційного набору.

Порівнняно з попереднім експериментом було виявленно менше помилок класифікації в тренувальному та валідаційному наборах. Ці варіанти мультикласової сегментації були обрані для того, щоб перевірити, який рівень класифікації та спрощення задачі найбільше сприятиме поліпшенню точності сегментації жилок, зменшуючи кількість помилок

класифікації пікселів. Тепер, маючи менше класів або об'єднаних класів, нейронна мережа зможе краще вивчити контекст і відтворити більш точну структуру жилок.

Binary Image Segmentation. У бінарній сегментації задається два класи: фон та жилкування листя. Це найпростіший варіант сегментації, де модель повинна визначити, чи є піксель частиною жилкування листя, чи належить до фону. Для цієї задачі застосовувалися класичні методи, такі як IoU (Intersection over Union) та Dice Similarity Coefficient для оцінки точності класифікації кожного пікселя.

Multi-Class Image Segmentation. У мультикласовій сегментації модель повинна класифікувати пікселі не лише як жилкування чи фон, а й відрізняти різні типи жилкування або інші елементи на листі (наприклад, листяна поверхня, межі листа). Мультикласова сегментація є більш складною задачею, оскільки вона вимагає більш детальної аналітики для кожного пікселя.

Для мультикласової сегментації також використовувалися метрики, як mIoU (Mean Intersection over Union) та Accuracy, для кожного класу окремо. Виявилося, що в умовах сильної різноманітності структур і текстур на листі мультикласова сегментація демонструє вищі результати у порівнянні з бінарною, особливо в контексті складних зображень з багатьма елементами. Однак мультикласова сегментація потребує більших обчислювальних ресурсів та більш складної обробки, оскільки кількість класів зростає.

У результатах порівняння обох підходів (бінарна та мультикласова сегментація) для набору даних показано, що мультикласова сегментація забезпечує вищу точність для складних зображень, де кілька різних класів можуть перекриватися або бути важко розрізняними, особливо у випадках, коли жилкування на листі має подібні текстури до фону або інших частин листа.

Підсумки по сегментаційним маскам:

- Бінарна сегментація ефективна для простих випадків, коли потрібно лише виділити жилкування від фону.

- Мультикласова сегментація є необхідною, коли зображення містить кілька об'єктів, що потребують окремої ідентифікації, як це відбувається у складних структурах рослинного листя.

3.10 Обговорення результатів

Використання U-Net для сегментації окремих бічних жилок всіх або однієї окремої, не є доцільним, оскільки бічні жилки часто мають схожі характеристики за структурою та текстурою, що ускладнює їх точне розпізнавання як окремих об'єктів. Ці жилки можуть бути надто тонкими, звивистими або перекриватися з іншими елементами зображення, що ускладнює їх сегментацію окремо. Тому, моделі для сегментації, як правило, об'єднують ці жилки в один клас, щоб спростити задачу та уникнути неточностей, які можуть виникнути через їх подібність.

Нейромережа може успішно сегментувати центральну жилку, бічні жилки об'єднані в один клас, оскільки центральна жилка має чітко виражену, довгу та рівну форму, що дозволяє легко її відокремити від інших елементів зображення. Вона виступає як ключова ознака для розрізнення між центральною жилкою та іншими частинами зображення. Бічні жилки, в свою чергу, можуть бути значно тоншими і менш помітними, тому їх об'єднання в один клас допомагає знизити складність сегментації і зберегти адекватність результатів на всіх етапах обробки. Для практичного використання можна запропонувати два сценарії використання нейромережі U-Net для сегментації жилок листя граба:

1. Для сегментації усіх жилок може бути використані дві паралельні нейромережі. Одна має бути навчана сегментувати тільки центральну жилку, друга всі жилки разом як один клас (центральну + всі бічні). Для отримання маски бічних жилок потрібно знайти різницю між прогнозними масками другої нейромережі та першої нейромережі. В результаті ми будемо мати маски з усіма об'єднаними в один клас бічними жилками. Для

сегментації бічних жилок на окремі класи може бути застосована інше нейромережа або методи сегментації засновані на обробці зображень.

2. Для сегментації усіх жилок може бути створена одна нейромережа що класифікує пікселі вхідного зображення на три або чотири класи: центральна жилка, всі бічні жилки об'єднані в один клас або бічні жилки об'єднані в два класи (один клас бічні жилки розташовані ліворуч, а другий клас бічні жилки розташовані праворуч). При цьому сценарії, прогнозовані маски бічних жилок також потребують додаткової обробки для виокремлення кожної бічної жилки та визначення її класу.

Однією з ключових тем експерименту було оптимізувати архітектуру нейромережі для точнішої сегментації жилкування листя. Стандартна архітектура UNet є популярним вибором для задач семантичної сегментації завдяки своїй здатності зберігати просторову інформацію завдяки симетричному поєднанню етапів енкодера та декодера. У нашому дослідженні було перевірено кілька варіантів модифікацій, зокрема зміни кількості фільтрів і етапів в енкодері.

У порівнянні з оригінальною версією UNet, модифікація архітектури, що включала більше фільтрів на кожному етапі енкодера, дала значне покращення в результатах сегментації. Це було особливо помітно при використанні глибших версій мережі, які змогли краще ідентифікувати тонкі деталі жилкування, що виявилися важливими для точної сегментації. Розширення кількості фільтрів дозволило моделі ефективніше захоплювати різні рівні абстракції, що допомогло покращити результати в сегментації складних та дрібних структур.

Вплив функцій втрат на точність сегментації. Під час експериментів з різними функціями втрат було виявлено, що wDice є найбільш підходящою функцією для цієї задачі, оскільки вона дозволяє знизити вагу фону і збільшити значення для менших класів, таких як жилкування, що забезпечує кращу точність на малих об'єктах. Інші функції втрат, такі як IoU та Dice, хоч і також показали хороші результати, але вони мали тенденцію до заниження точності на фонових пікселях, що призводило до меншої чутливості до тонких деталей.

Загалом, для досягнення найкращих результатів у задачах з сильно незбалансованими класами, доцільно застосовувати вагові функції втрат, які можуть враховувати важливість менших класів, таких як жилкування на листках.

Роль аугментації даних. Аугментація даних зіграла важливу роль у покращенні якості моделі, зокрема у випадках, коли наявний набір даних не був достатньо різноманітним. Використання таких методів, як обертання, віддзеркалення та зміна контрасту зображень, значно підвищило здатність моделі до узагальнення та її стабільність на валідаційних даних. Це дозволило зменшити перенавчання і підвищити точність сегментації на нових зображеннях, які мали відмінні характеристики від тренувальних.

Проблеми з дисбалансом класів. Однією з основних проблем, яку було виявлено в процесі експериментів, є сильний дисбаланс між фоном і об'єктами, що сегментуються. Величезний розрив у кількості пікселів фонового класу та класу жилкування листя призводив до того, що базові метрики точності не могли повністю відобразити ефективність моделі. У таких випадках модифікація функцій втрат і регуляризація дозволили значно покращити результати.

Порівняння з іншими підходами. Деякі альтернативні методи сегментації, такі як DeepLabV3+ або FCN (Fully Convolutional Networks), можуть бути застосовані до цієї задачі, однак вони потребують значно більших обчислювальних ресурсів, що не завжди є доступним для практичних застосувань. Зокрема, UNet виявився більш економічним у плані часу навчання та ресурсів, зберігаючи високу ефективність при помірному налаштуванні параметрів. Різниця між UNet і альтернативними методами була незначною, проте архітектура UNet залишалася більш зручним варіантом для обробки складних дрібних деталей, як у випадку з жилкуванням листя.

Тестування на реальних даних. Тестування моделі на реальних зображеннях, які не входили до тренувального набору, показало гарну здатність моделі до узагальнення. Модель змогла точно сегментувати жилкування на різних типах листя, що доводить її ефективність для практичних застосувань, таких як автоматичне дослідження листя в ботаніці або сільському господарстві.

Перспективи подальших досліджень:

1. Використання трансформерних моделей: Подальші дослідження можуть зосередитися на використанні трансформерних моделей, таких як Swin UNet, для підвищення точності та ефективності моделі.
2. Розширення набору даних: Збільшення розміру набору даних, а також впровадження нових методів аугментації може ще більше підвищити якість сегментації.
3. Моделі з використанням глибоких навчальних архітектур: Подальші моделі можуть враховувати взаємозв'язок між жилками та іншими елементами рослинного аркуша для поліпшення загальної точності.

ВИСНОВКИ

В ході дослідження було розроблено та навчено модель для сегментації жилкування листя за допомогою згорткової нейромережі UNet. Ключовими результатами є:

1. Було створено розмічений набір даних листя Carpinus betulus для тренування нейромережі. Зібрані та відскановані зразки листя в декількох місцях, що відрізнялись рівнем забруднення. Набір включав 200 зображень, на кожному розмічено до 30 класів жилок.
2. Мовою Python з використанням платформи GoogleColab була реалізована нейромережа U-Net, було здійснено її навчання та оцінка якості.
3. Ми здійснили експериментальне налаштування та навчання низки моделей. Оптимальною функцією втрат виявилися функція втрат Дайса та Жакара які дозволяють врахувати наявність незбалансованих класів. Було з'ясовано, що для навчання, без погіршення його якості, можуть бути використані зображення у відтінках сірого замість кольорових. Якість навчання можна покращити доповнюючі набір даних з використанням методів аугментації, в нашому випадку оптимальним було випадкове обертання в діапазоні +-45°. Без значного зменшення якості можна спростити нейромережу зменшуючи кількість фільтрів на першій стадії енкодера з 64 до 32.
4. При навчанні U-Net з цільовими масками, що містили різну кількість класів чи різні типи об'єднання класів, було показано, що нейромережа U-Net здатна гарно локалізувати жилки, але разом з тим значно гірше розв'язує задачу класифікації бічних жилок за їх номером;
5. Запропоновані сценарії практичного застосування нейромережі U-Net для сегментації жилок Carpinus betulus. Нейромережа може бути навчена сегментувати центральну жилку та бічні жилки як один клас, подальше відокремлення та ідентифікація окремих бічних жилок можуть бути здійснені методами обробки зображень або за допомогою іншої нейромережі.

СПИСОК ЛІТЕРАТУРИ

1. Abdulhamit S. (2019) Practical Guide for Biomedical Signals Analysis Using Machine Learning Techniques. *ScienceDirect*, 27-434 p.
2. Xu H., Blonder B., Jodra M., Malhi Y., Fricker M. (2020, September) Automated and accurate segmentation of leaf venation networks via deep learning. *New Phytologist*, 8-21 p.
3. Hiep X. H., Kiet T. N., Dinh Q. T. (2020) Plant Identification Using New Architecture Convolutional Neural Networks Combine with Replacing the Red of Color Channel Image by Vein Morphology Leaf. *Vietnam Journal of Computer Science*, 197-208 p.
4. Hoshang K., Bong M. F, Tanzila S., Amjad R. (2018, August) A New Leaf Venation Detection Technique for Plant Species Classification. *Computer Engineering and Computer Science*, 3315–3327 p.
5. Гонтова Т. М., Руденко В. П., Машталер В. В., Мінаєва А. О. (2016) Анатомія рослин. *Електронний архів Національного фармацевтичного університету*, 16 p.
6. Kolozsvari A. G. (2020) Morphometric and Physiological Analysis of Fagus Sylvatica and Carpinus Betulus Leaves. *Research Journal of Agricultura*, 66 -71 p.
7. Costa L., Ampatzidis Y., Casteluci L., Archer L. (2020, November) Determining leaf stomatal properties in citrus trees utilizing machine vision and artificial intelligence. *Research Gate*, 1107–1119 p.
8. Кругляк Ю. (2018) Морфометричні характеристики продихів рослин роду Philadelphus L. у зв'язку з їх посухостійкістю. *Науковий вісник Чернівецького університету. Біологія (Біологічні системи)*, 193-197 p.
9. Dittberner H., Korte A., Mettler-Altmann T., Weber A., Monroe G. (2018) Natural variation in stomata size contributes to the local adaptation of water-use efficiency in Arabidopsis thaliana. *PubMed*, 4052-4065 p.
10. Laura L., Harlan T. (2017) Comprehensive Methods for Leaf Geometric Morphometric Analyses. *PubMed*, 1480 p.

11. Migicovsky Z. (2022) Increases in vein length compensate for leaf area lost to lobing in grapevine. *American Journal of Botany*, 1063-1073 p.

12. Chitwood D., Mullins J., Migicovsky Z., Frank M., VanBuren R. (2021) Vein-to-blade ratio is an allometric indicator of leaf size and plasticity. *PubMed*, 571-579 p.

13. Sun Z., Cui T., Zhu Y., Zhang W., Shi S., Tang S., Du Z., Liu C., Cui R. (2018, September) The mechanical principles behind the golden ratio distribution of veins in plant leaves. *Scientific Reports*, 8 p.

14. Shi P., Miao Q., Niinemets U., Liu M., Li Y., Yu K., Niklas K. (2022) Scaling relationships of leaf vein and areole traits versus leaf size for nine Magnoliaceae species differing in venation density. *American Juornal of America*, 899-909 p.

15. He K., Niklas K., Niinemets U., Wang J., Jiao Y., Shi P. (2024) Significant correlation between leaf vein length per unit area and stomatal density: evidence from Red Tip and Chinese photinias. *PubMed*, 136 p.

16. Zhao W., Siddiq Z., Fu P., Zhang J., Cao K. (2017) Stable stomatal number per minor vein length indicates the coordination between leaf water supply and demand in three leguminous species. *PubMed*, 2211 p.

17. Жила А. І. (2017) Макро- та мікроморфологічна будова листків деяких видів роду Phaedranassa Ravenna (Amaryllidaceae), пов'язана з висотою зростання. *Інтродукція рослин*, 38-49р.

18. Iqbal H. S. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, Volume 2, 420 p.

19. Georgios B. G., Francis B., Raphael C., Michael M., Jennifer N. (2014). Signal Processing for Big Data. *IEEE Signal Processing Magazine*, Volume 31, 15-16 p.

20. Zhang Y., Dong Z., Liu A., Wang S., Ji G., Zhang Z., & Yang J. (2015). Magnetic resonance brain image classification via stationary wavelet transform and generalized eigenvalue proximal support vector machine. *Journal of Medical Imaging and Health Informatics*, Volume 5, 1395–1403 p.

21. Zhang C., Li B., Chen B., Cao H., Zi Y., & He Z. (2015). Weak fault signature extraction of rotating machinery using flexible analytic wavelet transform. *Mechanical Systems and Signal Processing*,Volume 64, 162–187 p.

22. Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q.(2021). Swin-unet: unet-like pure transformer for medical image segmentation. Lect. *Notes Comput. Sci.* 13803, 205–218. doi: 10.1007/978-3-031-25066-8_9

23. Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L. (2018). DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 834–848. doi: 10.1109/TPAMI.2017.2699184

24. Chen, L.-C., Papandreou, G., Schroff, F., Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. Available at: http://arxiv.org/abs/1706.05587.

25. Fu, L., Tola, E., Al-Mallahi, A., Li, R., Cui, Y. (2019). A novel image processing algorithm to separate linearly clustered kiwifruits. Biosyst. Eng. 183, 184–195. doi: 10.1016/j.biosystemseng.2019.04.024

26. Genze, N., Wirth, M., Schreiner, C., Ajekwe, R., Grieb, M., Grimm, D. G. (2023). Improved weed segmentation in UAV imagery of sorghum fields with a combined deblurring segmentation model. *Plant Methods* 19, 1–12. doi: 10.1186/s13007-023-01060-8

27. Hong, D., Gao, L., Yao, J., Zhang, B., Plaza, A., Chanussot, J. (2021). Graph convolutional networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 59, 5966–5978. doi: 10.1109/TGRS.2020.3015157

28. Hong, D., Zhang, B., Li, H., Li, Y., Yao, J., Li, C., et al. (2023). Cross-City Matters: A Multimodal Remote Sensing Benchmark Dataset for Cross-City Semantic Segmentation using High-Resolution Domain Adaptation Networks. Available at: http://arxiv.org/abs/2309.16499.

29. Tan J., Ung M., Cheng H., & Irvine J. (2020). A deep learning approach to heart sound classification. *Journal of Healthcare Engineering*, 20 p.

30. He K., Zhang X., Ren S., & Sun J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778 p.

31. Zhang Y., Yang Q., & Li Y. (2017). A survey on deep learning for big data. *Information Fusion*, Volume 42, 146-157 p.

32. Wang Z., Oates T., & Foskey M. (2017). Time series classification from scratch with deep neural networks: A strong baseline. *International Joint Conference on Neural Networks (IJCNN)*, 1578-1585 p.

33. Li R., Yang B., Jiang M., Xu W., & Li R. (2017). A survey of deep learning methods for protein–protein interaction prediction. *Computational Biology and Chemistry*, Volume 70, 10-17 p.

34. Mamoshina P., Vieira A., Zhavoronkov A., & Osterman A. (2016). Applications of deep learning in biomedicine. *Molecular Pharmaceutics*, Volume 13, 1445-1454 p.

35. Oluwatamilore O., Jonathan T., McGinnity T. M., Mufti M. (2023). The Multi-Recurrent Neural Network for State-Of-The-Art Time-Series Processing. *Science Direct*, Volume 222, 488-489 p.

36. Nguyen T. H., & Grishin D. (2017). Protein-protein interaction networks derived from deep learning models. *Bioinformatics*, Volume 33, 201-206 p.

37. Schmidhuber J. (2014). Deep learning in neural networks: An overview. *Neural Networks*, Volume 61, 85-117 p.

38. O'Shea T., & Nash R. (2015). An introduction to convolutional neural networks. *Cornell University*, 1-2 p.

39. Wang Z., Yan W., & Oates T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. *International joint conference on neural networks (IJCNN)*, 1578-1585 p.

40. Attia Z. I., Friedman P. A., Noseworthy P. A., Lopez-Jimenez F., Ladewig D. J., Satam G., & Asirvatham S. J. (2019). Age and sex estimation using artificial intelligence from standard 12-lead ECGs. *Arrhythmia and Electrophysiology*, Volume 12, 834 p.

41. Głowacki J., Kaliciak L., Sankowski W., & Wojciechowski K. (2017). Data augmentation in ECG classification using mobile neural networks. *Journal of Biomedical Informatics*, Volume 66, 156-165 p.

42. Sharma L. N., & Subramanian R. K. (2020). Brain–computer interface and its applications. *SN Computer Science*, Volume 1, 1-18 p.

43. Atzori M., Gijsberts A., Müller H., Caputo B., & Hager G. D. (2014). Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Frontiers in Neurorobotics*, Volume 8, 9 p.

44. Sutskever I., Vinyals O., & Le Q. V. (2014). Sequence to sequence learning with neural networks. *In Advances in neural information processing systems*, 3104-3112 p.

45. James A. N., Hsien W. C., and Matthew A. B. (2018 September 4) Machine learning: applications of artificial intelligence to imaging and diagnosis. *PubMed*, 111-118 p.

46. Abhishek J., Rohit R., Sumit S., Prakash C., Jayesh G & Manoj R. (2023, November 5). Analysis of EEG signals and data acquisition methods: a review. *Department of Computer Science and Engineering, Manipal Institute of Technology, ManipalManipal Institute of Technology, Manipal*, 11 p.

47. Hojjat S., Sharan S., Joseph B., Errol C., Shahrokh V.(2018). Recent Advances in Recurrent Neural Networks. *Computer Science, Neural and Evolutionary Computing*, Volume 12, 1-4 p.

48. Goodfellow I., Bengio Y., Courville A., & Bengio Y. (2016). Deep learning. *MIT press Cambridge*, Volume 1, 18 p.

49. Anthony L. C., & Dong E. C.(2018, March 23). Deep Neural Networks in a Mathematical Framework. *SpringerBriefs in Computer Science*, 59–79 p.

50. Bahdanau D., Cho K., & Bengio Y. (2014). Neural machine translation by jointly learning to align and translate. *Cornell University*, 1-9 p.

51. Cho K., Van M. B., Bahdanau D., & Bengio Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. *ACL Anthology*, 1-8 p.

52. Rita Q. F., Humberto P., María A. F. (2022). Biosignals analysis (heart, phonatory system, and muscles). *ReserchGate*, 7-26 p.

53. Lipton, Z. C., Kale, D. C., & Elkan, C. (2015). Learning to diagnose with LSTM recurrent neural networks. *Cornell University*, 1-18 p.

54. Alom M. Z., Taha T. M., Yakopcic C., Westberg S., Sidike P., Nasrin M. S., & Asari V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics*, Volume 8, 292 p.

55. Ozturk, A., & Gulten, A. (2020). Deep learning for plant identification in the wild. *Neural Computing and Applications*, Volume 32, 6527-6538 p.

56. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, Volume 542, 115-118 p.

57. Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 1-5 p.