ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ, ПРАВА ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Освітньо-професійна програма Інформаційні управляючі системи та технології
Спеціальність 126 Інформаційні системи та технології
Ступінь вищої освіти Магістр

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри
_____ Юрій УТКІН
«15» грудня 2022 року

КВАЛІФІКАЦІЙНА РОБОТА

на тему: **«Сегментація ділянок лісу за допомогою нейронної мережі»**

виконав здобувач вищої освіти денної форми навчання
Павленко Анатолій Анатолійович

Керівник кваліфікаційної роботи
професор, д. т. н. Вадим СЛЮСАР

Полтава – 2022 року

ЗМІСТ

ВСТУП

Актуальність теми кваліфікаційної роботи підтверджується необхідністю моніторингу площі покриву лісових масивів в умовах глобальних змін клімату. На даний час, для цього використовують знімки, що отримані при дистанційному зондуванні Землі. Однак вони містять однорідні області, для яких середньоквадратичні відхилення оцінок їх характеристик можна порівняти з розкидом між класами. Як наслідок, класичні методи сегментації не гарантують отримання необхідного результату.

Одним з варіантів вирішення цього питання є використання штучного інтелекту на базі нейронних мереж. В свою чергу, реалізація глибокого навчання потребує значних обсягів датасету та обчислювальних потужностей, що є обмежуючим фактором для впровадження систем AI IoT і реалізації інтелектуальних функцій безпілотних та висотних платформ. Як наслідок, доцільно орієнтуватися на зображення з низькою роздільною здатністю. При цьому властивості різних типів згорткових нейронних мереж, що використовуються, впливають на рівень якості одержуваних результатів. Проведений аналіз існуючих робіт свідчить про домінування досліджень архітектури U-Net, тоді як PSPNet потребує більш детального вивчення.

Зв'язок роботи з науковими програмами, темами. Дослідження, що проводились в рамках роботи, відповідають Концепції розвитку штучного інтелекту в Україні (розпорядження Кабінету Міністрів України № 1787-р від 29.12.2021), тематиці досліджень Навчально-дослідної лабораторії інтелектуальних систем, комп'ютерних мереж та інтернет речей Кафедри інформаційних систем та технологій Полтавського державного аграрного університету та дослідженням в рамках науково-дослідної роботи «Управління стратегією інноваційного розвитку підприємств в контексті підвищення їх конкурентоспроможності на аграрному ринку, сталого розвитку

та забезпечення продовольчої безпеки держави» (2021 р.), що фінансувалась господарськими договорами із замовником.

Метою кваліфікаційної роботи є підвищення ефективності моніторингу лісових масивів за рахунок використання глибокого навчання різних архітектур згорткових нейронних мереж сегментації лісу на зображеннях з низьким розрізненням при дистанційному зондуванні Землі.

Завданнями кваліфікаційної роботи є:

– аналіз особливостей реалізації моніторингу лісових масивів на основі дистанційного зондування Землі при використанні семантичної сегментації зображень;

– визначення складових елементів згорткових нейронної мережі при вирішенні завдань семантичної сегментації;

– розроблення моделей глибокого навчання згорткових нейронних мереж сегментації лісу різної архітектури на зображеннях з низьким розрізненням;

– оцінка точності запропонованих варіантів нейронних мереж;

– порівняльний аналіз властивостей синтезованих архітектур згорткових мереж та дослідження їхньої ефективності.

Об'єктом дослідження є процес сегментації зображень.

Предметом дослідження є нейронні мережі різної архітектури, що застосовуютьсчя для сегментації зображень.

Методами дослідження є: аналітичний, методи синтезу та навчання нейронних мереж сегментації ділянок лісу на зображеннях, робота з фреймворком Keras.

Інформаційна база кваліфікаційної роботи сформована з ресурсів, що містять інформацію про методи сегментації зображень, нейронні мережі та їх компоненти, що використовуються для виконання семантичної сегментації зображень, а також інструментарій для роботи з розробки та дослідження нейронних мереж на основі сучасних та перспективних архітектур.

Елементи наукової новизни роботи полягають в розробці моделей глибокого навчання згорткових нейронних мереж на основі удосконалених архітектур типу U-Net і PSPNet для завдань сегментації лісу на зображеннях з низьким розрізненням; порівняльній оцінці точності нейронних мереж на основі удосконалених архітектур U-Net та PSPNet.

Практична значущість роботи полягає в розробці рекомендацій щодо формування тренувальної та перевіркової виборки для моделей глибокого навчання архітектур згорткових нейронних мереж при використанні датасетів з відкритим доступом та обґрунтуванні вибору архітектури нейронної мережі для сегментації ділянок лісу.

Апробація результатів відбувалася в рамках IV Міжнародної науково-практичної конференції, присвяченої 50-й річниці кафедрі Інформаційних систем та технологій «Інтеграція інформаційних систем та інтелектуальних технологій в умовах трансформації інформаційного суспільства» (жовтень 2021 р., м. Полтава), Щорічної студентської наукової конференції Полтавського державного аграрного університету (листопад 2022 р., м. Полтава) та MRRS в рамках 2022 IEEE 2nd Ukrainian Microwave Week (листопад 2022 р., м. Київ).

За результатами досліджень здійснено 3 публікації тез доповідей.

Структура кваліфікаційної роботи логічно пов'язана з завданнями досліджень і містить вступ, три розділи основної частини, висновки, список використаних джерел, додатки. Загальний обсяг пояснювальної записки кваліфікаційної роботи складає 76 сторінок формату А4. Вона містить 37 рисунків і 7 таблиць.

РОЗДІЛ 1
АНАЛІЗ ІНСТРУМЕНТАРІЮ ДЛЯ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ

1.1 Аналіз чинників, що впливають на стан лісової екосистеми

Глобальний моніторинг навколишнього середовища – завдання, яке потребує уваги у зв'язку зі зміною клімату. Це включає в себе моніторинг темпів скорочення лісового покрову та територій, що постраждали від повені. У зв'язку з тим, що останнім часом загроза зміни клімату, що насувається, стає все більш серйозною, погодні явища, такі як періоди сильної спеки і поступове підвищення температури, стали більш помітними. Ці ефекти особливо помітні в посушливих ділянках місцевості, які останнім часом стали гарячими точками лісових пожеж величезної сили. Лісові пожежі в 21 столітті не тільки наразили на небезпеку лісу, а й забруднили повітря, змусивши тисячі людей залишити свої будинки і навіть забравши людські життя. При управлінні лісами слід вживати відповідних запобіжних заходів проти лісових пожеж, оскільки вони завдають тяжких втрат людям і навколишньому середовищу. Також існує шкода, що завдається лісам короїдом та вітровалом.

У світі виникає ще одна проблема з лісозаготівельною галуззю, яка стала звичайною практикою на тлі збільшення попиту на деревину. Найпростіший спосіб задовольнити цей новий попит – просто зрубати стільки дерев, скільки потрібно. В результаті це безповоротно руйнує крихкі лісові екосистеми та збільшує забруднення, аналогічно наслідкам лісових пожеж, описаних вище.

Щоб боротися зі зникненням лісових масивів, що утворилися природним шляхом, різні організації почали вирощувати ліси, що самопідтримуються. Даний підхід можливо розглядати як метод боротьби з руйнівними наслідками заготівки лісу, лісових пожеж. Зрештою, ліси на Землі є притулком для широкого кола тварин і поглинають CO_2 з атмосфери, використовуючи процес фотосинтезу, щоб перетворити його на дорогоцінний кисень. Це представляє проблему відстеження здоров'я та безпеки лісів, щоб звести до мінімуму

людські жертви та інциденти, а також побічні збитки нашому довкіллю. У зв'язку з частими коливаннями розмірів лісів метод простого відстеження та моніторингу граничних розмірів лісів має вирішальне значення у цю епоху.

Як відомо, на стан лісової екосистеми може впливати безліч факторів, основними з яких слід вважати: зміна клімату, інтенсивність пожеж, ступінь відновлення лісу після пожежі, склад фауни в даному ареалі, незаконна вирубка або інші варіанти впливу людини. Крім того, досить складним завданням є моніторинг пожеж, у тому числі природного походження, що виникають у лісових масивах з неоднорідним рельєфом, які територіально віддалені від населених пунктів.

1.2 Моніторинг лісових масивів на основі дистанційного зондування Землі

При використанні супутникових [1] та висотних платформ [2] дистанційного зондування Землі одним із важливих завдань є моніторинг лісових масивів. Щоб запобігти або пом'якшити завдані збитки, дослідники часто використовують аерофотознімання, супутникові знімки.

Такі зображення часто містять різноманітні однорідні області, для яких внутрішньокласові середньоквадратичні відхилення їх характеристик часто можна порівняти з розкидом між класами. Крім того, вони можуть мати низьку роздільну здатність. Ця технологія має значну потенційну цінність.

Тепер за допомогою нових інновацій у комп'ютерному зорі можна на широко поширених супутникових зображеннях визначати межі лісу та класифікацію рослинності. Для цього використовується процедура сегментації (рис. 1.1).

Рис. 1.1 – Приклад процедури сегментації ділянки ліса

Сегментація зображення – завдання пошуку груп пікселів, кожна з яких характеризує один значеннєвий об'єкт. У статистиці ця проблема відома як кластерний аналіз і є широко вивченою областю із сотнями різних алгоритмів. У комп'ютерному зорі, сегментація – це процес поділу цифрового зображення на кілька сегментів (множина пікселів, також званих суперпікселями). У комп'ютерному зір сегментація зображення є однією з найстаріших проблем, що широко вивчаються.

Мета сегментації полягає у спрощенні та/або зміні уявлення зображення, щоб його було простіше та легше аналізувати. Сегментація зображень використовується для того, щоб виділити об'єкти та межі (лінії, криві, тощо) на зображеннях.

Результатом такого процесу є множина сегментів, які разом покривають все зображення або множина контурів, виділених із зображення. Всі пікселі в сегменті схожі за деякою характеристикою (обчисленою властивістю), наприклад, за кольором, яскравістю або текстурою. Сусідні сегменти значно відрізняються за цією характеристикою. Деякі практичні

застосування сегментації зображень: медичні зображення; виявлення пухлин та ін. патологій; визначення обсягів тканин; хірургія за допомогою комп'ютера; діагностика; планування лікування; вивчення анатомічної структури; розпізнавання осіб; розпізнавання відбитків пальців; системи керування дорожнім рухом; виявлення стоп-сигналів; машинний зір; розпаралелювання інформаційних потоків при передачі зображень високої роздільної здатності; виділення об'єктів на супутникових знімках.

Таким чином, сегментація може допомогти у моніторингу залісення, вирубування лісів. Процес створення масок для зображень ще недосконалий, проте він значно покращився. Незважаючи на бурхливий розвиток алгоритмів комп'ютерного зору виявлення об'єктів на зображенні, завдання сегментації зображень для додатків дистанційного зондування щодо земної поверхні не автоматизована тією мірою, в якій точність спостерігається при ручній розмітці. Якщо цей процес продовжити на кілька років за допомогою супутникових зображень, зроблених із тих самих місць, можна порівняти попередні маски і легко побачити вплив на розмір лісів.

1.3 Методи сегментації зображень

Для сегментації зображень розроблено кілька універсальних алгоритмів та методів. Оскільки загального рішення завдання сегментації зображень немає, часто ці методи доводиться поєднувати зі знаннями з предметної області, щоб ефективно вирішувати це завдання у її предметної області.

Ітеративний метод «*k*-середніх» ґрунтується на кластеризації. Він використовується, щоб розділити зображення на k кластерів. Його суть полягає у наступному.

1. Вибрати до центрів кластерів, випадково (на підставі евристики).

2. Помістити кожен піксель зображення в кластер, центр якого найближче до цього пікселя.

3. Заново обчислити центри кластерів, усереднюючи всі пікселі в кластері.

4. Повторювати кроки 2 і 3 до збіжності (наприклад, коли пікселі залишатимуться у тому кластері).

Тут, як відстань, зазвичай береться сума квадратів або абсолютних значень різниць між пікселем і центром кластера. Різниця заснована на кольорі, яскравості, текстурі та розташуванні пікселя (виваженій сумі цих факторів). При цьому k вибирається вручну, випадково або евристично.

Методи з використанням гістограми дуже ефективні, коли порівнюються з іншими методами сегментації зображень, тому що вони потребують лише одного проходу пікселів.

Метод виділення країв – це добре вивчена область для обробки зображень. Межі та краї областей сильно пов'язані, оскільки часто існує сильний перепад яскравості на межах областей. Тому це використовується як основа іншого методу сегментації.

Виявлені краї часто бувають розірваними. Щоб виділити об'єкт на зображенні, потрібні замкнуті межі області.

Методи розростання областей. Першим був метод розростання областей із насіння. Як вхідні дані цей метод приймає зображення і набір насіння. Насіння відзначає об'єкти, які потрібно виділити. Області поступово розростаються, порівнюючи всі незайняті сусідні пікселі з областю. Різниця Δ між яскравістю пікселя та середньою яскравістю області використовується як міра схожості. Піксель з меншою такою різницею додається у відповідну область. Процес триває, доки всі пікселі не будуть додані до одного з регіонів. Метод розростання областей із насіння вимагає додаткового введення.

Результат сегментації залежить від вибору насіння. Шум на зображенні може спричинити те, що насіння погано розміщене.

У методах розрізу графа зображення подається як зважений неорієнтований граф та розрізається згідно з правилом «гарних» кластерів.

У сегментації методом вододілу розглядається абсолютна величина градієнта зображення як топографічної поверхні. Пікселі, що мають найбільшу абсолютну величину градієнта яскравості, відповідають лініям вододілу, які становлять межі областей. Вода, розміщена на будь-який піксель усередині загальної лінії вододілу, тече вниз до загального локального мінімуму яскравості. Пікселі, від яких вода стікається до загального мінімуму, утворюють водозбір, який є сегментом.

Сегментація за допомогою моделі. Можна знайти ймовірну модель для пояснення змін форми та, сегментуючи зображення, накладати обмеження, використовуючи її як апріорну. Сучасні методи для сегментації, що засновані на знанні, містять активні моделі форми та зовнішності, активні контури, деформовані шаблони та методи встановлення рівня.

Багатомасштабна сегментація виконується за різними масштабами. Критерій сегментації може бути довільно складним і може брати до уваги як локальні та глобальні критерії.

Одновимірна ієрархічна сегментація передбачає, що одновимірний сигнал може однозначно сегментуватись на області, використовуючи лише один параметр, що управляє масштабом сегментації.

У більш ранніх техніках використовується розщеплення та злиття регіонів, що відповідає роздільним та агломераційним алгоритмам у роботах із кластеризації. Сучасні алгоритми найчастіше оптимізують деякі глобальні критерії, такі як внутрішньо-регіональна узгодженість та міжрегіональні довжини кордонів.

Також використовують ще два алгоритми – графо-орієнтована сегментація та метод нормалізованих зрізів. Перший є базовим алгоритмом сегментації, який простий і зрозумілий у реалізації, але повільний і результати недостатньо хороші. Другий алгоритм є просунутою версією першого з безліччю евристик. Що відображає, як на продуктивності, так і на результатах вказаних процедур.

1.4 Аналіз методів бібліотеки комп'ютерного зору OpenCV

OpenCV (Open Source Computer Vision Library) – це відкрита бібліотека для роботи з алгоритмами комп'ютерного зору, машинним навчанням та обробкою зображень [3]. Написана на C++, але існує також Python, JavaScript, Ruby та ін.. Працює на Windows, Linux, MacOS, iOS та Android.

OpenCV може використовуватися скрізь, де потрібний комп'ютерний зір. Ця галузь IT працює з технологіями, які дозволяють пристрою «побачити», розпізнати та описати зображення. Комп'ютерний зір дає точну інформацію про те, що зображено на зображенні, з описом, характеристиками та розмірами (з певним ступенем достовірності).

OpenCV застосовується: у робототехніці – для орієнтування робота у просторі, розпізнавання об'єктів та взаємодії з ними; медичні технології – для створення точних методів діагностики, наприклад, 3D-візуалізації органу при МРТ; промислові технології – для автоматизованого контролю якості, зчитування етикеток, сортування продуктів тощо; безпеці – для створення «розумних» камер відеоспостереження, що реагують на підозрілі дії, для зчитування та розпізнавання біометрії; мобільній фотографії – для створення б'юті-фільтрів, що змінюють особу додатків; на транспорті – для розробки автопілотів. OpenCV має наступні функції.

Робота із структурами даних. Для зберігання та роботи із зображеннями OpenCV використовує вектори та скаляри, матриці та діапазони. Вони

дозволяють проводити математичні перетворення, орієнтуватися на зображення і виконувати безліч інших дій.

Видозміна зображень. За допомогою OpenCV з картинкою можна працювати як у графічному редакторі: обрізати, збільшувати чи зменшувати, обертати. Здебільшого програмісти використовують цю можливість для попередньої підготовки картинки перед її розшифровкою – наприклад, обрізають непотрібні частини.

Додавання ефектів. Картинку можна зробити у відтінках сірого або повністю чорно-білого. Це важливо для алгоритмів розпізнавання, які працюють із знебарвленими зображеннями. Можна змінювати тон кольору, розмивати, згладжувати або геометрично змінювати картинку.

Рисування поверх зображення. На картинку можна нанести лінії та геометричні фігури, зробити підпис, наприклад, щоб виділити знайдене програмою обличчя. Часто це використовується в мобільних додатках для камери: квадрат навколо людини під час зйомки означає, що програма розпізнала його.

Розпізнавання об'єктів. Для розпізнавання елементів OpenCV використовуються обриси об'єктів, сегментація за кольорами, вбудовані методи розпізнавання, які можна настроювати залежно від об'єкта і чутливості алгоритму.

Робота із відеороликами. Нові версії бібліотеки підтримують роботу не лише з картинками, а й з відео. Вони можуть зчитувати ролики з використанням кодеків, аналізувати те, що відбувається в них, відстежувати рухи та елементи. Це корисно, наприклад, при програмуванні робота, що рухається, або створенні ПЗ для камери відеоспостереження.

Структура OpenCV – це множинні модулі для різних цілей: зберігання математичних функцій та обчислень, алгебри, збереження структур даних; зберігання моделей для машинного навчання; введення та вивід картинок або відео, читання та запису у файл; обробки зображення; розпізнавання примітивів; детектування об'єктів – осіб, предметів та інших; відстеження та

аналізу рухів на відео; обробки тривимірної інформації; прискорення роботи бібліотеки; зберігання застарілого або ще не готового коду та інших. Кожен модуль вузькоспеціалізований. Їх не потрібно завантажувати окремо: у пакет установки включена вся основна функціональність бібліотеки.

Серед недоліків OpenCV можна вказати таке.

Складність у освоєнні. Щоб добре розуміти всі можливості OpenCV, потрібно знати теорію комп'ютерного зору та машинного навчання. Тому поріг входу в галузь вищий, ніж у інших популярних напрямках IT.

Відсутність кодів обробки помилок. Якщо виникла помилка, OpenCV буває складно зрозуміти, де саме. Тому при налагодженні програм у новачків можуть бути проблеми.

Орієнтованість на великі платформи. OpenCV працює на масштабних платформах. Якщо запустити її на мікроконтролері, одноплатному комп'ютері, продуктивність буде невисокою.

Далі є доцільним розглянути основні варіанти сегментації зображень на основі бібліотеки OpenCV.

Алгоритм сегментації за вододілами (WaterShed). Алгоритм працює із зображенням як із функцією від двох змінних:

$$f = I(x, y),$$

де x, y – координати пікселя.

Значення функції може бути інтенсивність або модуль градієнта. Для найбільшого розмаїття беруть градієнт від зображення. Якщо по осі Oz відкладати абсолютне значення градієнта, то місцях перепаду інтенсивності виникають хребти, а однорідних регіонах – рівнини. Після знаходження мінімумів функції f йде процес заповнення «водою», який починається з глобального мінімуму. Як тільки рівень води досягає значення чергового локального мінімуму, починається його заповнення водою. Коли два регіони починають зливатися, створюється перегородка, щоб запобігти об'єднанню

областей. Вода продовжить підніматися доти, доки регіони не відокремлюватимуться лише штучно побудованими перегородками (рис. 1.2).

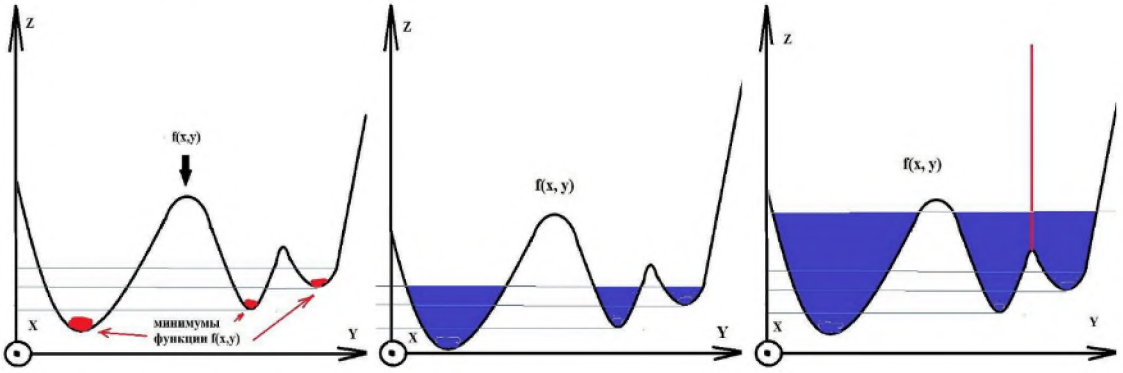


Рис. 1.2 – Ілюстрація процесу заповнення водою

Такий алгоритм є корисним, коли на зображенні невелика кількість локальних мінімумів, у разі їх великої кількості виникає надмірне розбиття на сегменти. Наприклад, якщо безпосередньо застосувати алгоритм до рис. 1.3, отримаємо багато дрібних деталей (рис. 1.4).

Рис. 1.3 – Вихідне зображення

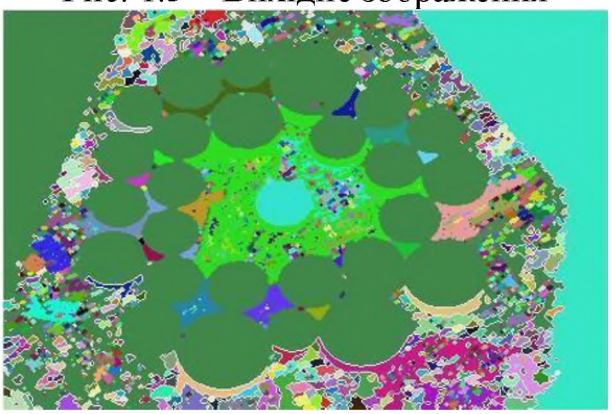


Рис. 1.4 – Зображення після сегментації алгоритмом WaterShed

Рис. 1.5 – Формування маски

Основним недоліком даного алгоритму є використання процедури попередньої обробки для картинок з великою кількістю локальних мінімумів (зображення зі складною текстурою та різноманіттям різних кольорів).

Алгоритм сегментації MeanShift. Він групує об'єкти із близькими ознаками. Пікселі зі схожими ознаками поєднуються в один сегмент, на виході отримуємо зображення з однорідними областями.

Наприклад, як координати в просторі ознак можна вибрати координати пікселя (x, y) та компоненти RGB пікселя (рис. 1.6). Зображуючи пікселі у просторі ознак, можна побачити згущення [4]. Щоб легше було описувати згущення точок, вводиться функція щільності:

$$f(\vec{x}) = \frac{1}{Nh^d} \sum_{i=1}^{N} K(\frac{\vec{x} - \vec{x}_i}{h}), \tag{1.1}$$

де $\vec{x}$ – вектор ознак i-го пікселя;

d – кількість ознак;

N – кількість пікселів;

h – параметр гладкості,

$K(\vec{x})$ – ядро.

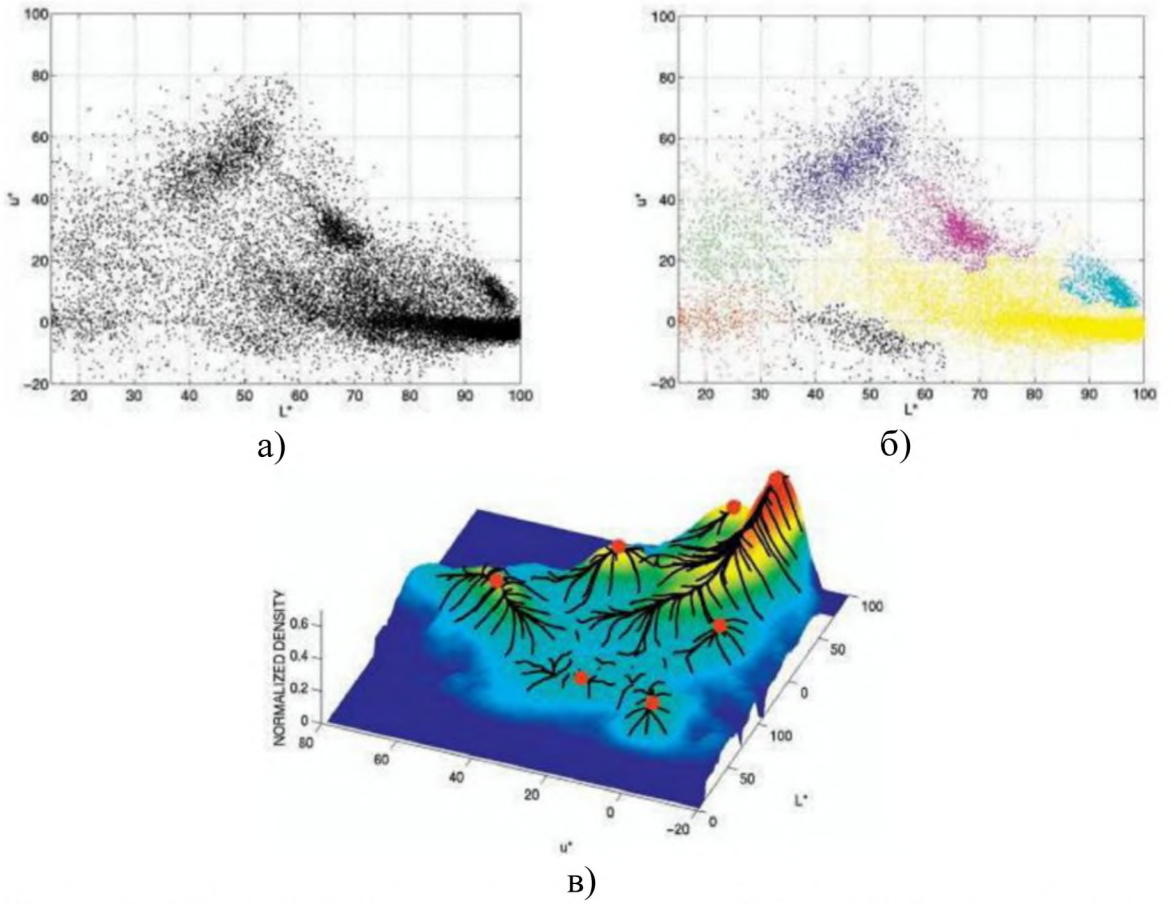


Рис. 1.6 – Простір ознак алгоритму сегментації MeanShift: а) – пікселі у 2-мірному просторі ознак; б) – пікселі, що прийшли в один локальний максимум, забарвлені в один колір; в) – функція щільності $f(\vec{x})$ для якої максимуми відповідають місцям найбільшої концентрації пікселів

Максимуми $f(\vec{x})$ розташовані в точках згущення пікселів зображення у просторі ознак. Пікселі одного локального максимуму, поєднуються в один сегмент. Щоб знайти до якого з центрів згущення відноситься піксель, треба крокувати градієнтом $f(\vec{x})$ для пошуку найближчого локального максимуму. При виборі ознак координат пікселів та інтенсивностей за кольорами в один сегмент об'єднуватимуться пікселі з близькими кольорами і розташовані недалеко один від одного. Якщо вибрати інший вектор ознак, то об'єднання пікселів у сегменти вже йтиме по ньому. Якщо прибрати з ознак

координати, то небо та озеро будуть вважатися одним сегментом, оскільки пікселі цих об'єктів у просторі ознак потрапили б до одного локального максимуму.

Якщо об'єкт, який хочемо виділити, складається з областей, які сильно відрізняються за кольором, MeanShift не зможе об'єднати ці регіони в один, і об'єкт складається з кількох сегментів. Але добре впоратися з однорідним за кольором предметом на строкатому тлі. Ще MeanShift використовують при реалізації алгоритму стеження за об'єктами, що рухаються.

Алгоритм сегментації FloodFill (заливання або метод «повені»). З його допомогою можна виділити однорідні за кольором регіони. Для цього потрібно вибрати початковий піксель та задати інтервал зміни кольору сусідніх пікселів щодо вихідного. Інтервал може бути несиметричним. Алгоритм об'єднуватиме пікселі в один сегмент (заливаючи їх одним кольором), якщо вони потрапляють у вказаний діапазон. На виході буде сегмент, який залитий певним кольором, та вказано його площу у пікселях.

Такий алгоритм буде корисним для заливання області із слабкими перепадами кольору однорідним тлом. Одним із варіантів використання FloodFill може бути виявлення пошкоджених країв об'єкта. Нижче на рис. 1.7 можна помітити, що цілісність меж областей зберігається.

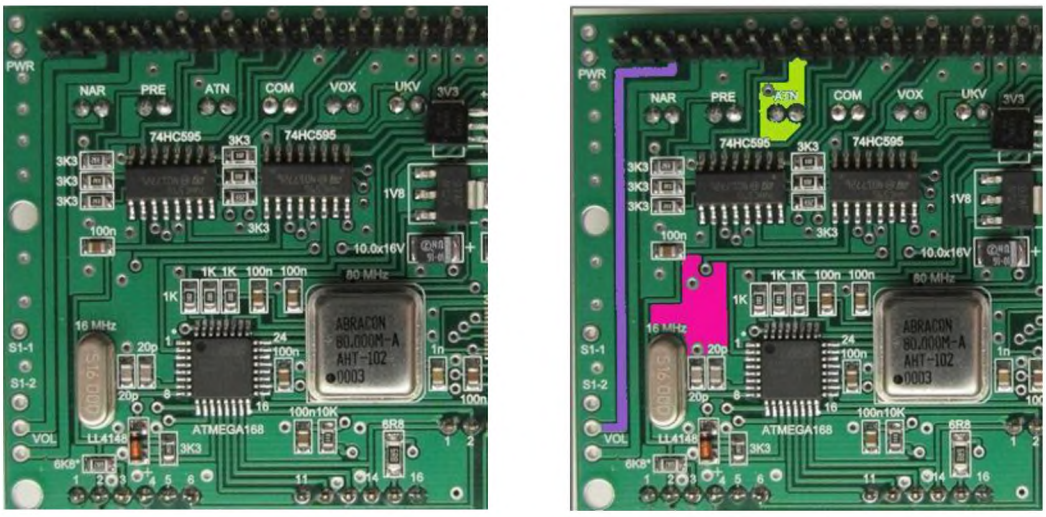


Рис. 1.7 – Вихідне зображення та результат після заливання кількох областей

На рис. 1.8 показано варіант роботи FloodFill у разі пошкодження однієї з меж у попередньому зображенні.

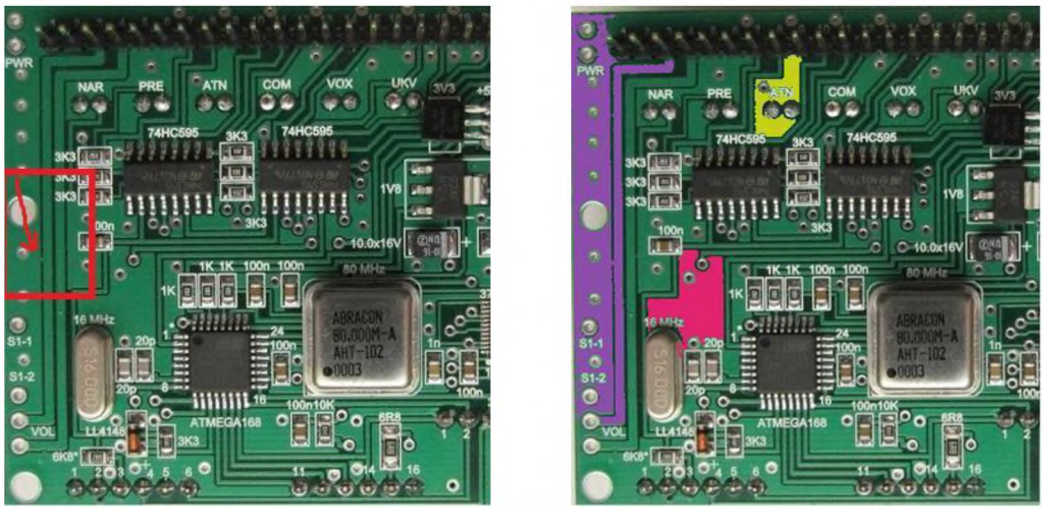


Рис. 1.8 – Ілюстрація роботи FloodFill при порушенні цілісності кордону між областями, що заливаються

GrabCut – інтерактивний алгоритм виділення об'єкта, що розроблявся як зручніша альтернатива магнітному ласо (щоб виділити об'єкт, потрібно було обвести контур). Для роботи алгоритму достатньо укласти об'єкт разом із частиною тла у прямокутник (grab). Сегментування об'єкта відбудеться автоматично (cut) – рис. 1.9.

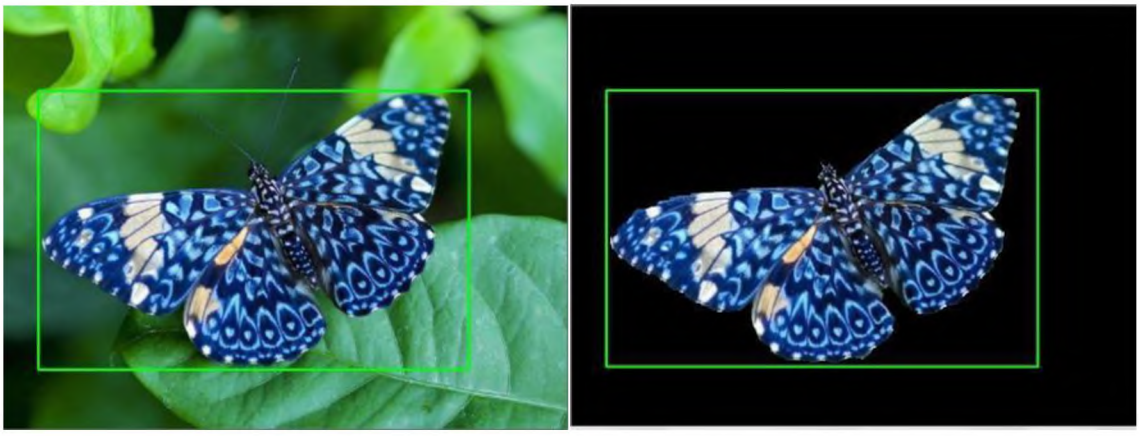


Рис. 1.9 – Робота алгоритму сегментації GrabCut

Можуть виникнути складності при сегментації, якщо всередині прямокутника, що обмежує, присутні кольори, які зустрічаються у великій

кількості не тільки в об'єкті, але і на тлі. У цьому випадку можна поставити додаткові позначки об'єкта (червона лінія) та фону (синя лінія) – рис. 1.10.

Розглянемо ідею алгоритму. За основу взято алгоритм інтерактивної сегментації GraphCut, де користувачу треба поставити маркери на тлі та на об'єкт. Зображення розглядається як масив $z(z_1,\ldots,z_n,\ldots,z_N)$. Z – значення інтенсивності пікселів, N-загальна кількість пікселів. Для відділення об'єкта від фону алгоритм визначає значення елементів масиву прозорості $a(a_1,\ldots,a_n,\ldots,a_N)$, причому a_n може приймати два значення, якщо $a_n = 0$ піксель належить фону, якщо $a_n = 1$ – об'єкту.

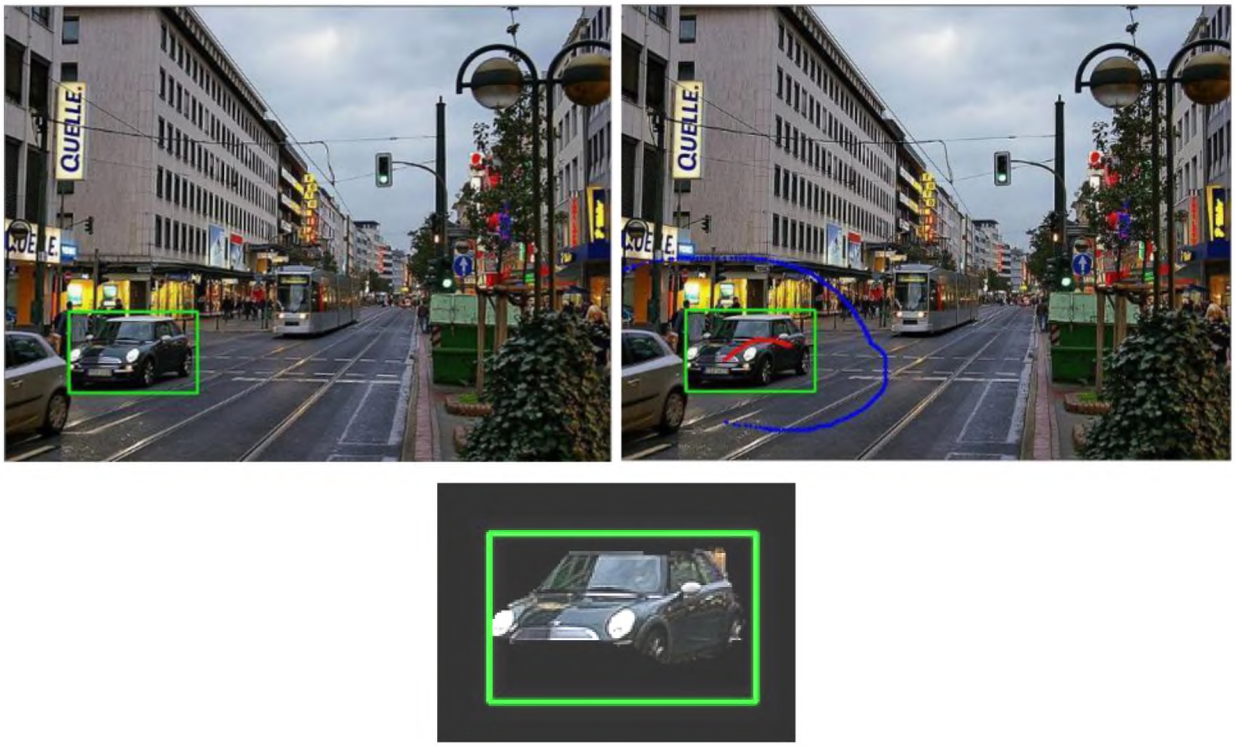


Рис. 1.10 – Встановлення додаткових позначок об'єкта

Внутрішній параметр θ містить гістограму розподілу інтенсивності переднього плану та гістограму фону:

$$\theta = \{h(z;a), a = 0,1\}. \quad (1.2)$$

Завдання сегментації – знайти невідомі. Розглядається функція енергії:

$$E(a,\theta,z) = U(a,\theta,z) + V(a,z). \quad (1.3)$$

Причому мінімум енергії відповідає найкращій сегментації:

$$U(a,\theta,z) = -\sum_{n} \log h(z_n, a_n),$$

$$V(a,z) = \sum_{(m,n)\in C} \frac{1}{dis(m,n)} [a_n \neq a_m] \exp(-\beta(z_m - z_n)^2),$$

де $U(a,\theta,z)$ – відповідає за якість сегментації, тобто. поділ об'єкта від фону;

$V(a,z)$ – доданок відповідає за зв'язок між пікселями (сума йде по всіх парах пікселів, які є сусідами);

$dis(m,n)$ – евклідова відстань;

$[a_n \neq a_m]$ – відповідає за участь пар пікселів у сумі, якщо $[a_n = a_m]$, то ця пара не враховуватиметься.

Знайшовши глобальний мінімум функції енергії *E*, отримаємо масив прозорості:

$$\hat{a} = \arg\min_a E(a,\theta).$$

Для мінімізації функції енергії зображення описується як граф і відшукується мінімальний розріз графа. На відміну від GraphCut в алгоритмі GrabCut пікселі розглядаються в просторі RGB, тому для опису колірної статистики використовують суміш Gaussian Mixture Model.

В результаті сегментації на зображенні виділяються області, які об'єднуються пікселі за вибраними ознаками. Для заливання однорідних за кольором об'єктів підходить FloodFill. З завданням відокремлення конкретного об'єкта від фону добре впорається GrabCut. Якщо використовувати реалізацію MeanShift із OpenCV, то пікселі, близькі за кольором та координатами, будуть кластеризовані. WaterShed підходить для зображень із простою текстурою. Отже, алгоритм сегментації слід вибирати, звісно, з конкретної завдання.

1.5 Вибір інструментарію для сегментації лісу

Застарілі методи комп'ютерного зору, такі як колірна гістограма та оцінка частоти кольору, а також інші моделі вилучення карти ознак, такі як випадкові ліси (RF) (Breiman, 2001) та умовні випадкові поля (CRF) (Lafferty et al., 2001) та ін., використовувалися раніше. Однак такі алгоритми в основному неефективні, оскільки зображення місцевості схильні до кліматичних та місцевих змін. Крім того, колірні характеристики в межах одного класу роблять їх невідмінними один від одного, іноді навіть для людського ока.

Для вирішення цього завдання все частіше застосовують штучний інтелект. Машинне навчання – це методи штучного інтелекту, які дозволяють побудувати учні для різних цілей: наприклад, автоматизації процесів, автоматичного перекладу текстів, розпізнавання зображень. Супутникові зображення дуже допомагають ефективному моніторингу Землі, а методи глибокого навчання з урахуванням нейронних мереж допомагають автоматизувати цей процес моніторингу.

Використання нейронних мереж [5-9] забезпечує більш точні оцінки автоматичної ідентифікації особливостей об'єктів та класифікації місцевості.

При цьому розв'язання задач дистанційного зондування Землі засобами аерокосмічного базування в основному реалізується на основі нейромережевих технологій Semantic Segmentation [10] (визначає належність наборів пікселів на зображенні до певних класів об'єктів).

Семантична сегментація надає кожному пікселю зображення певну мітку класу, що є основною вимогою нашої задачі класифікації. Семантична сегментація ділить дані в домені на дрібніші одиниці, такі як суперпікселі, супервокселі, одиниці на основі сітки та ін. Виявлення об'єктів, навпаки, використовує алгоритм зіставлення шаблонів і створює рамку, що обмежує, над одиницями на основі кореляції між відповідний шаблон і дані пікселів. Така обмежувальна рамка ніколи не відповідає виявленим класам та не може

використовуватися на рельєфах місцевості, знятих за допомогою аерофотознімків із дронів, де розмір таких об'єктів надзвичайно малий. Глибокі нейронні мережі перевершують будь-які інші структури, які використовуються в комп'ютерному зорі для вирішення проблем у галузях, як розпізнавання образів, вилучення ознак та виявлення/класифікація. Через складність візерунків біля поточна проблема класифікації вимагає правильного вилучення ознак із зображень для класифікації.

Крім семантичної сегментації існує Instance Segmentation (кожен об'єкт усередині одного класу виділяється окремими сегментами). У свою чергу, поєднання цих підходів (Semantic і Instance segmentation) породжує технологію Panoptic segmentation.

Для цього набір даних повинен мати всі можливі закономірності, що впливають на кліматичні умови. Також важливо збирати та ідентифікувати дані, які не повинні бути прив'язані до одного класу, оскільки одні моделі можуть бути більшими на місцевості, ніж інші. Наскільки нам відомо, такого набору даних для ідентифікації лісової місцевості поки що немає. Для ідентифікації місцевості потрібні зображення, які мають бути отримані з висоти пташиного польоту, щоб охопити більшу площу. Деякі з наявних доступних наборів даних є CityScapes (Cordts et al., 2016), PASCAL (Everingham et al., 2015), UAVID (Lyu et al., 2020) та ін. Однак жоден з вищенаведених наборів даних не містить зображень, пов'язаних із зображенням лісу, і в основному зосереджені в міських районах, що містять будівлі, дорожні доріжки та ін.

Таким чином, все це свідчить про доцільність використання зображень, які одержують на основі дистанційного зондування Землі, для оперативної оцінки екосистеми лісів. Класичні методи сегментації не гарантують отримання необхідного результату. Тому необхідно дослідити можливість семантичної сегментації на основі нейронних мереж.

Висновки до розділу 1

Для відстеження динаміки змін стану лісових екосистем можливо застосовувати дистанційне зондування Землі, що проводиться штучними супутниками та різноманітними висотними платформи, в тому числі і на основі БПЛА.

Отриманні таким чином зображення можуть мати низьку роздільну здатність та різноманітні однорідні області, обробку яких можливо автоматизувати за допомогою використання комп'ютерного зору та штучного інтелекту.

Таке завдання відноситься до реалізації процедури сегментації зображення. Найбільш відомим прикладом, що містить потрібний інструментарій є бібліотека OpenCV, наприклад, реалізації GrabCut, MeanShift та WaterShed. Однак, алгоритм сегментації слід вибирати, виходячи з конкретного завдання.

Проведені дослідження свідчать про необхідність використання замість класичних методів семантичної сегментації перспективних рішень на основі нейронних мереж. Такий підхід забезпечує більш точну оцінку автоматичної ідентифікації особливостей об'єктів та класифікації місцевості.

РОЗДІЛ 2
ДОСЛІДЖЕННЯ АРХІТЕКТУРИ НЕЙРОННИХ МЕРЕЖ ДЛЯ СЕМАНТИЧНОЇ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ

2.1 Складові елементи нейронних мереж

Алгоритми глибоких нейронних мереж сьогодні набули великої популярності, яка багато в чому забезпечується продуманістю архітектур.

Одним із їх елементів є повнозв'язаний шар (Dense). Відповідно, мережі, які повністю складаються з шарів Dense, називаються повнозв'язними. В складних архітектурах серед згорткових шарів, десь у середині, цілком може бути повнозв'язаний шар. Від його наявності мережа не стає повною. Тож у чому недоліки повнозв'язних мереж?

По-перше, у ресурсоємності таких шарів.

По-друге, є таке поняття «локалізація ознаки». Воно означає, що й у картинці ми почнемо міняти місцями розташування пікселів, це буде інша картинка тієї самої розміру, так як у зображенні важливе взаємне розташування даних.

По-третє, з використанням повнозв'язного шару кожен нейрон пов'язані з кожним нейроном попереднього шару, тобто кожен нейрон аналізує все зображення і може знайти хибні залежності.

Звичайно, це не означає, що їх використовувати не можна або вони не спрацюють. Можуть бути навіть двомірні шари Dense, але є більш ефективні типи шарів для аналізу зображень. До таких відносяться згорткові шари. Основна відмінність таких верств у тому, що вони зберігають просторову структуру зображення. Відповідно зі згорткових шарів будується мережа згортки (Convolutional Neural Networks, CNN) [11-15].

Свою назву CNN отримала за назвою операції. Проведені дослідження дозволяють виділити три основні принципи CNN: локальне сприйняття; ваги, що розділяються; зменшення розмірності.

Локальне сприйняття. На відміну від повнозв'язних мереж, які аналізують всю картинку повністю, CNN аналізують картинку частинами, проходячи по ній ядром згортки (рис. 2.1). Ядро є фільтром або вікном, яке ковзає по всій області зображення і знаходить певні ознаки об'єктів. Наприклад, якщо мережу навчали на безлічі осіб, то одне з ядер могло б у процесі навчання видавати найбільший сигнал у одній сфері, інше ядро могло б виявляти інші ознаки. Розмір фільтра згортки визначається в налаштуваннях шару, так само як і кількість таких фільтрів.

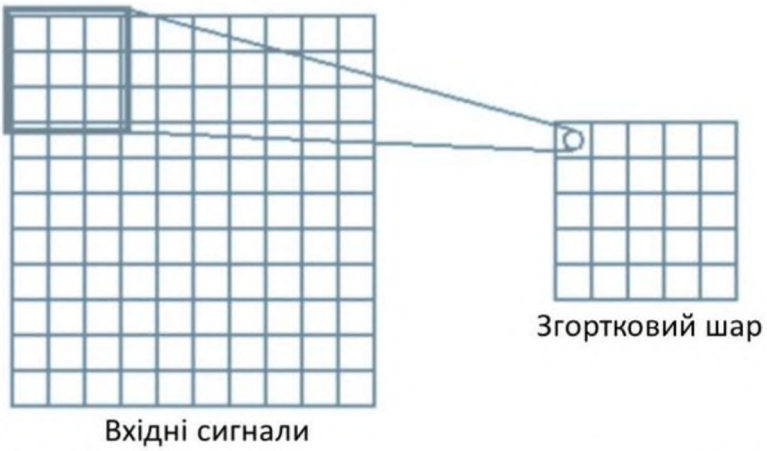


Рис. 2.1 – Сутність використання згортки в CNN

Таким чином, ядро є системою ваг, що розділяються, це одна з головних особливостей CNN. У повнозв'язаній мережі дуже багато зв'язків між нейронами, що дуже уповільнює процес детектування. У CNN – навпаки, загальні ваги дозволяють скоротити зв'язки і дозволити знаходити одну ознаку по всій області зображення. Залежно від методу обробки країв вихідної матриці, результат може бути меншим від вихідного зображення (valid), того ж (same) або більшого розміру (full) – рис. 2.2. У спрощеному вигляді цей шар можна описати формулою:

$$x^l = f(x^{l-1} * k^l + b^l), \tag{2.1}$$

де x^l – вихід шару l;

$f()$ – функція активації;

b^l – коефіцієнт зсуву шара l;

* – операція згортки x з ядром k.

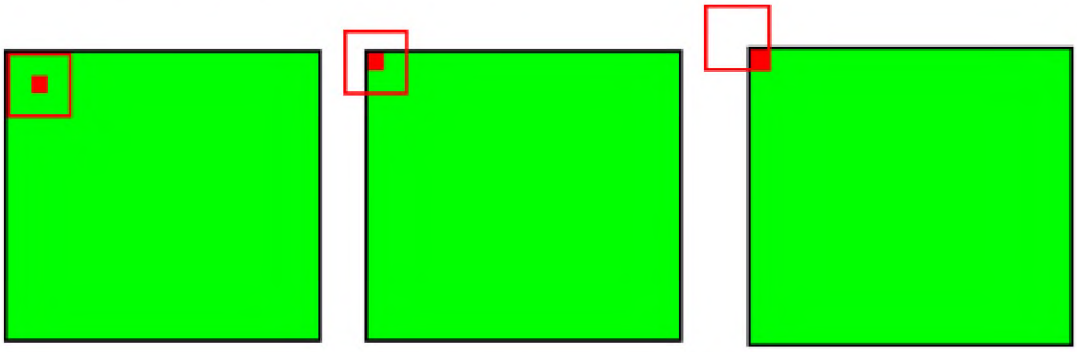


Рис. 2.2 – Види згортки вихідної матриці

За рахунок крайових ефектів розмір вихідних матриць зменшується:

$$x_j^l = f\left(\sum_i x_i^{l-1} * k_j^l + b_j^l\right), \tag{2.2}$$

де x_j^l – карта ознак j (вихід шару l);

b_j^l – коефіцієнт зсуву шару l для картки ознак j.

Тепер використовуватимемо ваги у вигляді невеликих фільтрів — просторових матриць, які проходять по всьому зображенню та виконують скалярний твір на кожній його ділянці. Розмір фільтра завжди відповідає розмірності вихідного знімка. В результаті проходу по зображенню отримуємо карту активації, також відому як карта ознак (рис. 2.3). Цей процес називається просторової згорткою.

До зображення можна застосовувати велику кількість фільтрів і отримувати на виході різні карти активації. Так формується один згортковий шар. Щоб створити цілу нейронну мережу, шари чергуються один за одним, а між ними додаються функції активації (наприклад, ReLU) і спеціальні шари Pooling, що зменшують розмір карт ознак (рис. 2.4). Принцип роботи сучасних згорткових нейронних мереж пояснюється схемою, яка представлена на рис. 2.5.

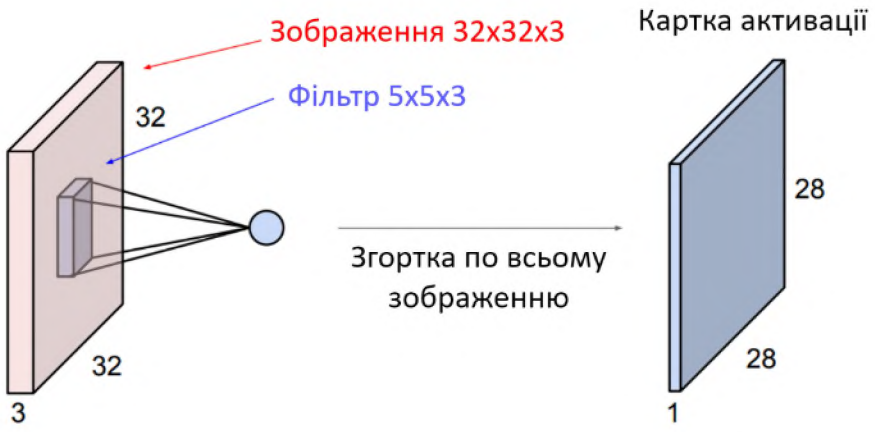


Рис. 2.3 – Формування картки активації

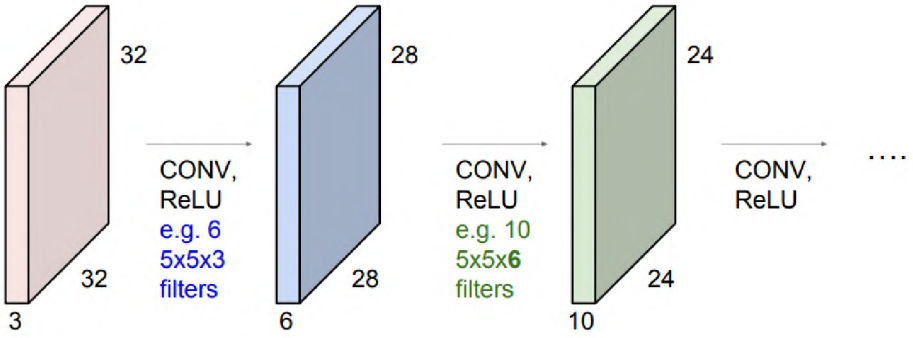


Рис. 2.4 – Зменшення розміру карт ознак

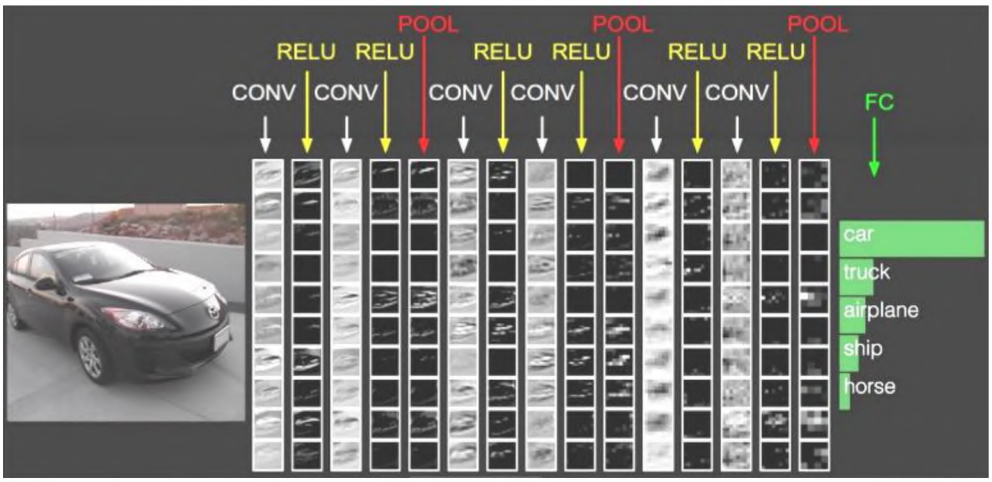


Рис. 2.5 – Схема роботи згорткової нейронної мережі

Розглянемо докладніше, що являють собою згорткові фільтри. У перших шарах вони зазвичай співвідносяться з низькорівневими ознаками зображення, наприклад, з краями та межами. У середині присутні більш складні особливості, такі як кути та кола. І на фінальних шарах фільтри вже більше нагадують деякі специфічні ознаки, які можна інтерпретувати ширше.

Склавши все разом, ми отримаємо приблизно таку картину: взявши вихідну фотографію, проводимо її через згорткові шари, що чергуються, функції активації і Pooling-шари. Наприкінці використовуємо звичайний повнозв'язний шар, з'єднаний з усіма висновками, який показує підсумкові оцінки кожного класу.

2.2 Особливості реалізації згорткових шарів

Існують одновимірні, двовимірні і навіть тривимірні згорткові шари. Але переважно використовуються двовимірні. По суті, ядро згортки виконує поелементне множення частини зображення маску і складає отримані значення. Далі робить крок і повторює операцію. Найпоширеніші ядра двовимірних згорткових шарів 3x3, 5x5 і 7x7, а кількість ядер, як правило, використовують кратне двом (8, 16, 32, 64 та ін.). Але ніхто не забороняє використовувати ядро 3x7 або 27x27 і кількість ядер 315. Наприклад, такі ядра згортки використовуються у фільтрах Photoshop (рис. 2.6).

Розмиття			Виділення границь			Підвищення чіткості		
1/9	1/9	1/9	0	-1	0	0	-1	0
1/9	1/9	1/9	-1	4	-1	-1	5	-1
1/9	1/9	1/9	0	-1	0	0	-1	0

Рис. 2.6 – Варіанти фільтрів

Так формується нова картинка. Кожне ядро/нейрон згортки формує свою картинку. Не важливо, скільки шарів прийшло на вхід. Кожен нейрон/ядро видає лише одну картинку. Тому, незалежно від того, подали одношарову чорно-білу картинку або 3-шарову RGB, кількість картинок, сформованих згортковим шаром, дорівнює кількості нейронів/ядер згортки згорткового шару.

У кожного ядра згортки у процесі навчання формується своя матриця, і це ядро починає відповідати пошук певного ознаки. У спрощеному варіанті

одне ядро шукає горизонтальну пряму, інше вертикальну, третє діагональ та ін. На наступному шарі відшукуються кути. У нейронних мережах ядра згортки визначаються автоматично.

При створенні шару Conv1D вказується кількість фільтрів (filters=20) та розмір ядра (kernel_size=5). До переваг одновимірної згорткової нейронної мережі слід віднести те, що час навчання значно нижче, ніж у рекурентних нейронних мереж, а як недолік – немає можливості «запам'ятати» потрібні дані на тривалий термін. Але його усунути за допомогою механізму уваги. Формат завдання такого шару в синтаксисі фреймворку Keras має вигляд:

$$\text{model.add}\left(\text{Conv1D}\left(20,\ 5,\ \text{activation='relu'}\right)\right)$$

Двовимірна (просторова) згортка визначається за допомогою шару Conv2D. Цей шар створює ядро згортки, яке згортається із входом шару для отримання тензора виходів. Якщо значення параметра use_bias дорівнює True, то створюється та додається до вихідних даних вектор зміщення. Нарешті, якщо активація не параметр None, він застосовується до вихідних даних. При використанні цього шару в якості першого шару моделі, необхідно задати ключовий аргумент input_shape (кортеж цілих чисел, не включає вісь партії), наприклад, $\text{input\_shape}=\left(128,\ 128,\ 3\right)$ для 128×128 RGB-картинок data_format=«channels_last» . У модель мережі він додається просто:

$$\text{model.add}\left(\text{Conv2D}\left(32,\ \left(3,\ 3\right),\ \text{padding='same'},\ \text{activation='relu'}\right)\right)$$

У цьому: перший параметр (32) – визначає кількість ядер згортки; другий параметр (3,3) – розмір ядра згортки; параметр padding='same' – необхідний збереження розміру картинки, інакше пікселі з обох боків «обрізаються», за умовчанням значення 'valid'; параметр activation='relu' – звичайна вказівка функції активації; strides – не обов'язковий параметр, ставить зрушення, тобто. на скільки має зсунути вікно згортки.

Підвибірковий шар (Subsampling) також, як і згортковий має карти, але їх кількість збігається з попереднім (згортковим) шаром, їх 6. Мета шару – зменшення розмірності карт попереднього шару. Якщо на попередній

операції згортки вже були виявлені деякі ознаки, то для подальшої обробки настільки детальне зображення не потрібно і воно ущільнюється до менш докладного. Фільтрація вже непотрібних деталей допомагає не перевчитися.

У процесі сканування ядром підвибіркового шару (фільтром) карти попереднього шару, скануюче ядро не перетинається на відміну від згорткового шару. Зазвичай кожна карта має ядро розміром 2x2, що дозволяє зменшити попередні карти згорткового шару в 2 рази. Вся карта ознак поділяється на комірки 2x2, у т. ч. вибираються максимальні за значенням.

Шар підвиборки трохи схожий на згортковий, у нього також є ядро згортки. Але на відміну від згорткового шару він зменшує розмір зображення (зазвичай у 2 рази), вибираючи максимальне (MaxPooling) або середнє (AveragePooling) значення вікна згортки.

При зміщенні вікна згортки не перетинаються. Певний шар робить інформацію більш сконцентрованою. Для двовимірної згортки вони записуються як MaxPooling2D та AveragePooling2D. Шар MaxPooling2D здійснює операцію стиснення (зменшення розмірів) зображення шляхом вибору максимального значення у блоці пікселів (рис. 2.7). Відповідно, роботу AveragePooling2D можна пояснити рис. 2.8.

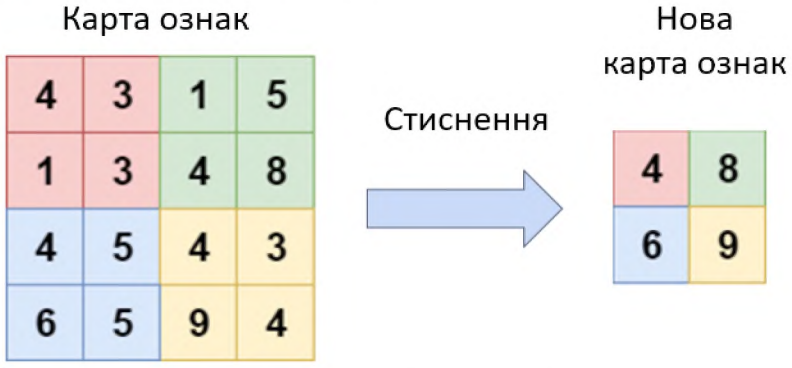


Рис. 2.7 – Шар MaxPooling2D

Таким чином, шари Pooling забезпечують розпізнавання об'єктів незалежно від масштабу, а також визначають факт наявності ознаки важливіше

знання місця його точного становища на зображенні. Формат запису таких шарів має вигляд:

model.add.MaxPooling2D(pool_size=(2, 2), strides=None, padding='valid', data_format=None)

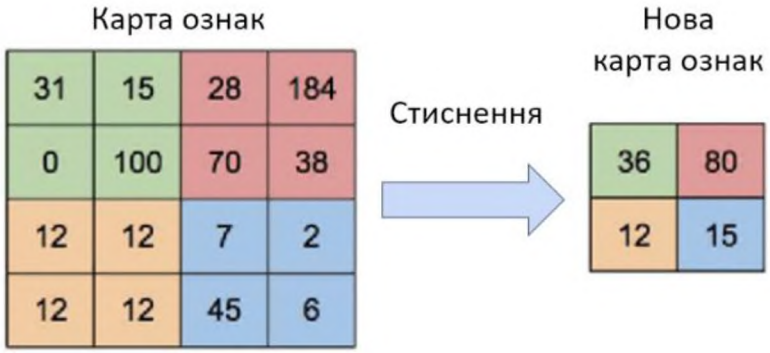


Рис. 2.8 – Шар AveragePooling2D

При цьому аргументи мають такі значення:

– **pool_size**: ціле число чи кортеж із 2-х цілих чисел, чинники, якими слід зменшувати масштаб (вертикальний, горизонтальний). (2, 2) зменшить вхідне значення в обох просторових вимірах наполовину. Якщо вказано лише одне ціле число, то для обох вимірювань буде використана та сама довжина вікна;

– **strides**: ціле число, кортеж з двох цілих чисел або None. Значення кроків. Якщо None, то за промовчанням буде використано значення pool_size;

– **padding**: одне з 'valid' або 'same' (без урахування регістру);

– **data_format**: рядок, один з channels_last (за замовчуванням) або channels_first. Порядок прямування розмірів на входах. channels_last відповідає входам із формою (batch, height, width, channels), тоді як channels_first відповідає входам із формою (batch, channels, height, width). За замовчуванням параметром image_data_format, знайденим у конфігураційному файлі Keras, є ~/.keras/keras.json. Якщо він ніколи не встановлювався, це буде «channels_last». В цілому, краще намагатися використовувати такі дозволи зображень, щоб обидві розмірності ділилися на 2 без залишку. Існують шари,

які виконують зворотну операцію. Але якщо взяти зображення 15x15, застосувати MaxPooling, то на виході буде зображення 7x7. І якщо потрібно застосувати Conv2DTranspose (один з «розтискаючих» шарів), то вийде зображення 14x14, що не співпадатиме з вихідними розмірами.

Окрім цих базових шарів у CNN використовують «вирівнюючий» шар Flatten. Це досить простий шар, він витягує дані в рядок, робить їх плоскими. Наприклад, для датасету з MNIST зображення 28x28x1 перетворюється на 784. Цей шар не потребує параметрів:

```
model.add(Flatten())
```

Цю ж дію можна виконати за допомогою шару Reshape, але потрібно вважати нові розмірності. З шаром Flatten нічого не потрібно вираховувати. Рис. 2.9 може дати уявлення про те, що відбувається в нейронах CNN.

Рис. 2.9 – Використання шару Reshape

Перший шар, умовно, шукає геометричні фігури, далі слідує зменшення розмірності. Другий шар із фігур попереднього шукає вже окремі елементи обличчя, знову зменшення розмірності. Третій шар з елементів осіб складає обличчя.

2.3 Функції активації CNN

Одним із етапів розробки нейронної мережі є вибір функції активації нейронів. Вигляд функції активації багато в чому визначає функціональні

можливості нейронної мережі та метод її навчання. Простими словами, штучний нейрон зважує виважену суму на входах, додає зсув (bias) і вирішує, чи слід це значення виключати або використовувати далі. Функція активації визначає вихідне значення нейрона в залежності від результату виваженої суми входів та порогового значення:

$$Y = \sum(wight \cdot input) + bias.$$

Тепер значення Y може бути будь-яким у діапазоні від $-\infty$ до $+\infty$. Насправді нейрон не знає межу, після якої слідує активація.

Для цього вирішили додавати активаційну функцію. Вона перевіряє вироблене нейроном значення Y щодо того, чи повинні зовнішні зв'язку розглядати цей нейрон як активований, чи його можна ігнорувати.

Далі більш докладно розглянемо властивості функцій активації, які найбільше частот використовують у CNN.

ReLu (Rectified Linear Unit) представлена на рис. 2.10. Стає зрозуміло, що ReLu повертає значення x, якщо x позитивно, і 0 інакше:

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

$$f'(s) = \begin{cases} 1, & s > 0 \\ rand(0.01, 0.05), & s \le 0 \end{cases}.$$

На перший погляд здається, що ReLu має ті самі проблеми, що й лінійна функція, оскільки ReLu лінійна в першому квадранті. Але насправді ReLu нелінійна за своєю природою, а комбінація ReLu також нелінійна. Тобто така функція є хорошим апроксиматором, оскільки будь-яка функція може бути апроксимована комбінацією ReLu. Це означає, що можна формувати стеки шарів. Область допустимих значень ReLu – $[0, \inf)$, тобто активація може вибухнути. Наступний пункт – розрідженість активації. Уявимо велику нейронну мережу з безліччю нейронів. Використання сигмоїди або гіперболічного тангенсу спричинятиме активацію всіх нейронів аналоговим способом. Це означає, що майже всі активації мають бути

оброблені для опису виходу мережі. Інакше кажучи, активація щільна, але це затратно. В ідеалі ми хочемо, щоб деякі нейрони не були активовані, це зробило б активації розрідженими та ефективними.

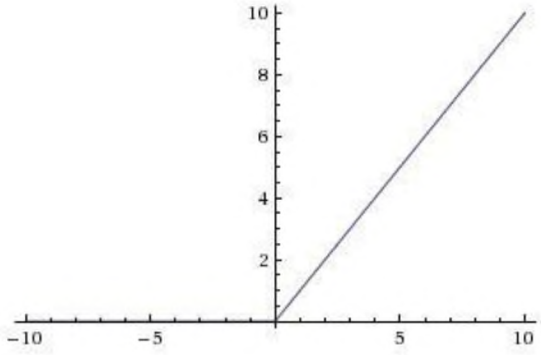


Рис. 2.10 – Функція активації ReLu

ReLu дозволяє це зробити. Уявімо мережу з випадково ініціалізованими вагами (або нормалізованими), в якій приблизно 50% активацій дорівнюють 0 через характеристики ReLu (повертає 0 для негативних значень x). У такій мережі включається менша кількість нейронів (розріджена активація), а мережа стає легше. Відмінно здається, що ReLu підходить за всіма параметрами. Але ніщо не ідеальне, зокрема і ReLu. Через те, що частина ReLu є горизонтальною лінією (для негативних значень x), градієнт на цій частині дорівнює 0. Через рівність нулю градієнта, ваги не будуть коригуватися під час спуску. Це означає, що нейрони, що знаходяться в такому стані, не будуть реагувати на зміни в помилці або вхідних даних (просто тому, що градієнт дорівнює 0, нічого не змінюватиметься). Таке явище називається проблемою вмираючого ReLu $\left(\text{Dying ReLu problem}\right)$. Через цю проблему деякі нейрони просто вимикаються і не відповідатимуть, роблячи значну частину пасивної нейромережі. Однак є варіації ReLu, які допомагають уникнути цієї проблеми. Наприклад, є сенс замінити горизонтальну частину функції на лінійну. Якщо вираз для лінійної функції задається як $y = 0.01x$ для області $x < 0$, то лінія трохи відхиляється від горизонтального положення. Існують інші способи уникнути нульового градієнта. Основна ідея тут – зробити градієнт нерівним нулю та поступово відновлювати його під час тренування.

ReLu менш вимогливо до обчислювальних ресурсів, ніж гіперболічний тангенс або сигмоїда, оскільки робить простіші математичні операції. Тому є сенс використовувати ReLu при створенні глибоких нейронних мереж.

Таким чином, серед переваг використання ReLu можна вказати такі положення: позбавлена ресурсомістких операцій; відсікає непотрібні деталі; відсутнє розростання (загасання) градієнта; швидке навчання. Оданк ReLu не завжди надійна, у процесі навчання може вмирати; сильно залежна від ініціалізації вагів.

Окрім ReLu у CNN також використовується функція активації Softmax, в основному, на виході мережі. Вона має забезпечити спосіб моделювання мережею імовірнісного розподілу. Softmax – це узагальнення логістичної функції багатовимірного випадку. Функція перетворює вектор $\vec{z}$ розмірності k в вектор $\vec{\delta}$ тієї ж розмірності, де кожна координата $\vec{\delta}(i)$ отриманого вектора представлена речовим числом в інтервалі [0, 1] та сума координат дорівнює 1.

2.4 Архітектури нейронних мереж для задач сегментації

Сегментація – це виділення об'єктів у вихідних даних. Як окремий клас, сегментація зображень полягає в підсвічуванні об'єктів на зображенні, тобто можна сказати, що сегментація зображень – це класифікація пікселів, де кожному пікселю об'єкта присвоюється певний клас. Розбір сцени, що базується на семантичній сегментації, є фундаментальною темою комп'ютерного зору. Мета полягає в тому, щоб присвоїти кожному пікселю зображення мітку категорії. Аналіз сцени забезпечує повне розуміння сцени. Він передбачає мітку, місцезнаходження, а також форму для кожного елемента. Ця тема представляє широкий інтерес для потенційних програм автоматичного водіння, виявлення роботів та багатьох інших.

Складність аналізу сцен тісно пов'язана з різноманітністю сцен і міток (рис. 2.11). Відповідно до [16], існує кілька загальних проблем для розбору складних сцен.

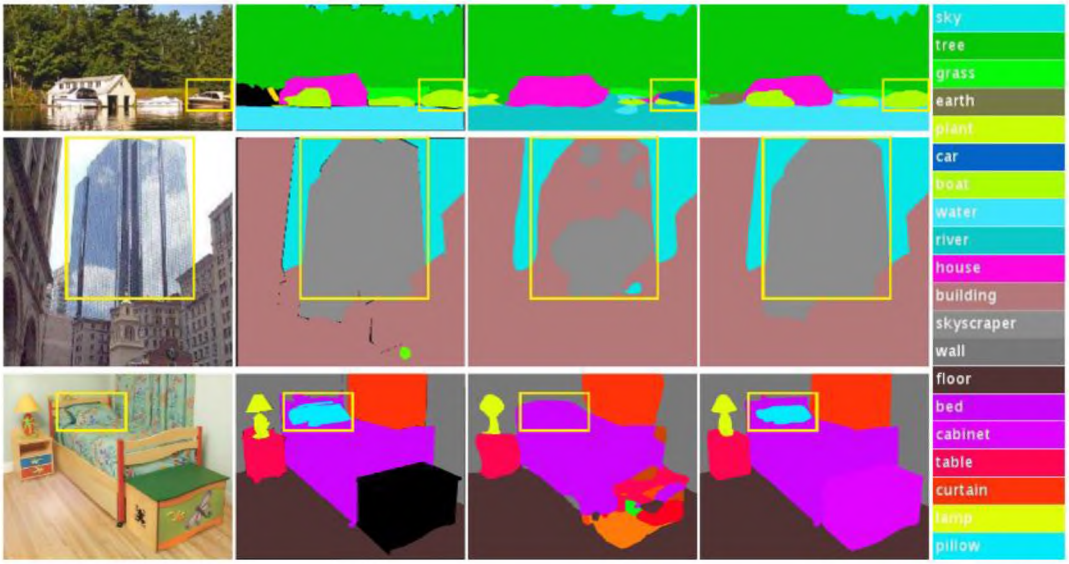
а) б)

Рис. 2.11 – Ілюстрація складних сцен у наборі даних ADE20K:

а) – зображення;

б) – справжній фон.

Відповідні співвідношення. Контекстні співвідношення універсальні та важливі, особливо розуміння складних сцен. При цьому існують супутні візуальні патерни, наприклад, літак швидше за все перебуватиме на злітно-посадковій смузі або літатиме в небі, але не над дорогою. Наприклад, на рис. 2.12 перший рядок показує проблему невідповідності віднощень – автомобілі рідко бувають над водою, ніж човни. Другий рядок показує категорії плутанини, де клас «будівлю» легко сплутати з «хмарочосом». Третій рядок ілюструє непомітні класи. У цьому прикладі подушка за кольором і фактурою дуже схожа на простирадло. FCN легко неправильно класифікує ці непомітні об’єкти [17].



а) б) в) г) д)

Рис. 2.12 – Проблеми аналізу сцени в наборі даних ADE20K [43]:

а) – зображення;

б) – справжній фон;

в) – маска сформована FCN;

г) – маска сформована PSPNet;

д) – карта кольорів.

У першому рядку FCN передбачає човен у жовтому полі як «автомобіль» на основі його зовнішнього вигляду. Але відомо, що автомобіль рідко переїжджає річку. Відсутність можливості збирати контекстуальну інформацію збільшує можливість неправильної класифікації.

Категорії плутанини. У наборі даних ADE20K є багато пар міток класів, класифікація яких призводить до плутанини. Приклади: поле та земля; гора та пагорб; стіни, будинок, будинок і хмарочос. Вони зі схожим зовнішнім виглядом. Експерт-анотатор, що розмітив весь набір даних, як і раніше, робить піксельну помилку 17,60% [16]. У другому рядку рис. 2.12 FCN передбачає об'єкт у рамці як частину хмарочоса та частину будівлі. Ці результати слід виключити, щоб весь об'єкт був або хмарочосом, або будинком, але не тим і іншим водночас. Цю проблему можна вирішити, використовуючи відносини між категоріями.

Непомітні класи. Сцена містить об'єкти (речі) довільного розміру. Декілька дрібних речей, таких як вуличний ліхтар та вивіска, важко знайти, хоча вони можуть мати велике значення. Навпаки, великі об'єкти або предмети можуть виходити за межі рецептивного поля FCN і викликати уривчасте передбачення. Як показано у третьому рядку на рис. 2.12 подушка має такий же зовнішній вигляд, що і простирадло. Пропуск категорії глобальної сцени може призвести до збою під час аналізу подушки. Щоб покращити продуктивність для дуже маленьких або великих об'єктів, слід приділяти багато уваги різним підобластям, які містять елементи непомітної категорії.

Підсумовуючи цим дослідженням, можна сказати, що багато помилок частково чи повністю пов'язані з контекстуальними відносинами та загальною інформацією для різних рецептивних полів. Таким чином, глибока мережа з відповідним апріорним рівнем глобальної сцени може значно підвищити продуктивність аналізу сцени.

Як відомо, однією з головних переваг CNN є використання спільної ваги у згорткових шарах, для кожного пікселя шару використовується один і той же фільтр (банк ваги); це як зменшує обсяг необхідної пам'яті, так і поліпшує продуктивність.

Для вирішення завдань сегментації зображень існують різні архітектури згорткових мереж нейронних мереж.

1. Автокодувальник (Autoencoder) – спеціальна архітектура штучних нейронних мереж, що дозволяє застосовувати навчання без вчителя під час використання методу зворотного поширення помилки. Мета навчання автокодувальників: знайти внутрішню структуру даних.

На сьогодні, автокодувальники застосовують для: поліпшення якості зображення (збільшення розміру, шумозаглушення та ін.); генерації нових даних за заданим зразком (наприклад, розфарбовування чорно-білих зображень у кольорові); пошук викидів.

Найпростіша архітектура автокодувальника – це мережа прямого поширення без зворотних зв'язків, найбільш схожа з персептроном і містить вхідний шар, проміжний шар та вихідний шар. На відміну від персептрона вихідний шар автокодувальника повинен містити стільки ж нейронів, як і вхідний шар (рис. 2.13). Автокодувальник складається з двох частин: енкодер X (кодує вибірку X у своє внутрішнє уявлення Z) і декодера (відновлює вихідну вибірку) X'.

Таким чином, автокодувальник намагається поєднати відновлену версію кожного об'єкта вибірки з вихідним об'єктом.

Одне із основних призначень таких автокодувальників – зниження розмірності вихідного простору. Коли маємо справу з автокодировщиками, сама процедура тренування нейронної мережі змушує автокодировщик запам'ятовувати основні ознаки об'єктів, якими буде простіше відновити вихідні об'єкти вибірки.

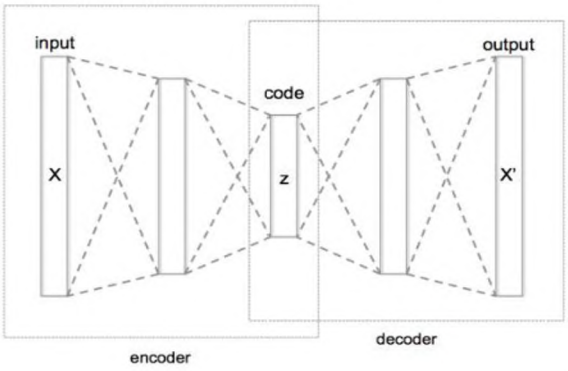


Рис. 2.13 – Архітектура Autoencoder

Можна налаштувати архітектуру мережі таким чином, що вона збільшуватиме розмір зображень, наприклад, зі 100x100 пікселів у 200x200 без втрати якості. Також мережа непогано справляється із завданням придушення шуму на фотографіях.

2. U-Net. По структурі мережа схожа з автокодировщиком, у якому мережа стискає дані у прихований простір, цим виявляючи основні ознаки, і потім відновлює зображення з прихованого простору. Архітектура U-Net є послідовністю блоків, спочатку зменшуючи розмірність зображення (блоки

включають Pooling-шари), а потім збільшуючи, попередньо об'єднавши з виходами початкових блоків з відповідною розмірністю (рис. 2.14) [19-22].

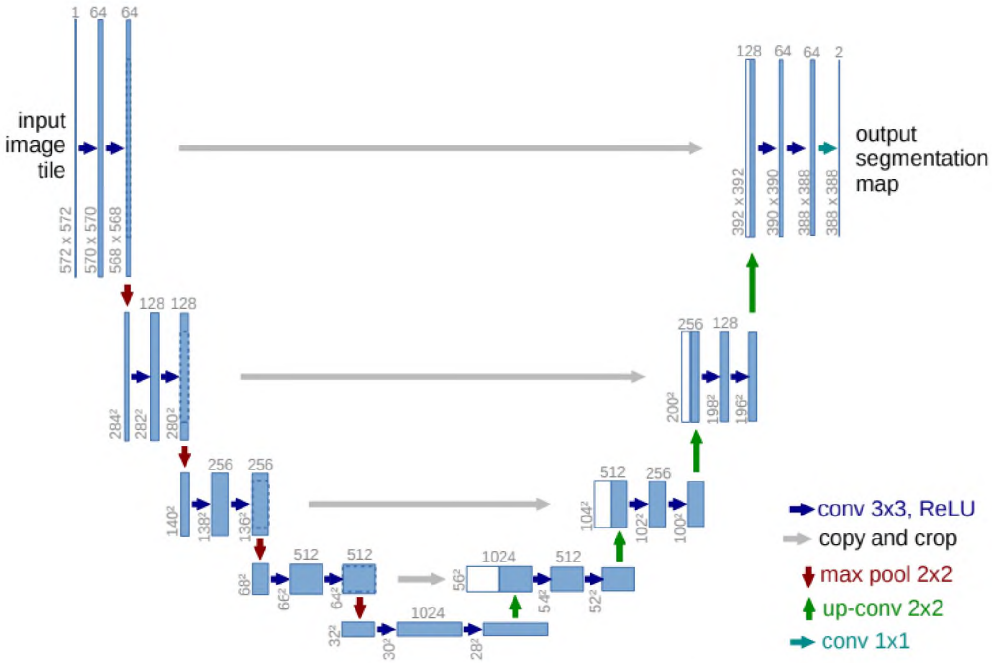


Рис. 2.14 – Архітектура U-Net

До основних переваг такої архітектури належить таке: обчислювально ефективна; навчається на невеликому датасеті; спочатку – для біомедичних зображень.

При цьому U-Net має недоліки: ускладнення архітектури при багатокласовій сегментації; проблема кордонів.

3. SegNet – це згортковий автокодувальник, останній шар якого – шар попіксельної класифікації (рис. 2.15). Архітектура складається з згорткових блоків, спочатку зменшуючи розмірність зображення (блоки включають Pooling-шари), а потім збільшуючи до вихідної розмірності.

4. LinkNet. За своєю структурою LinkNet – це U-Net, де шари об'єднання (Concatenate) замінені на шари додавання (рис. 2.16).

5. Архітектура Pyramid Scene Parsing Network (PSPNet) побудована на моделі пірамід Pooling і, зазвичай, складається з блоку кількох згорткових шарів (рис. 2.17) [23].

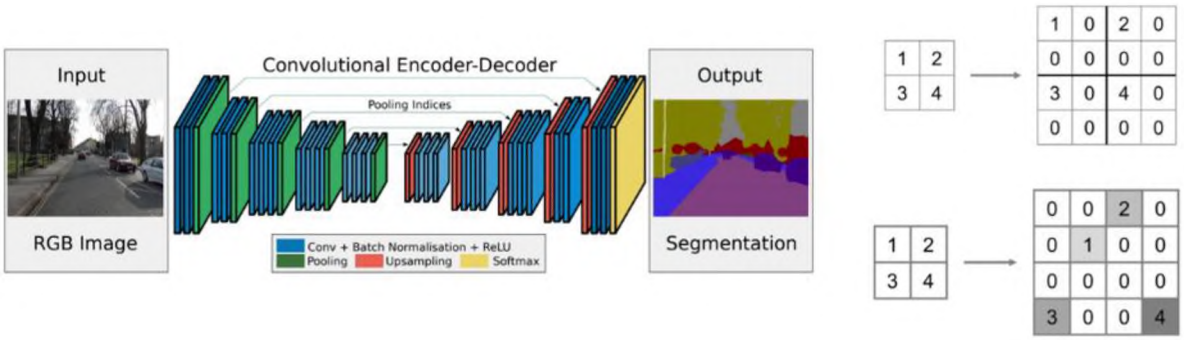


Рис. 2.15 – Архітектура SegNet

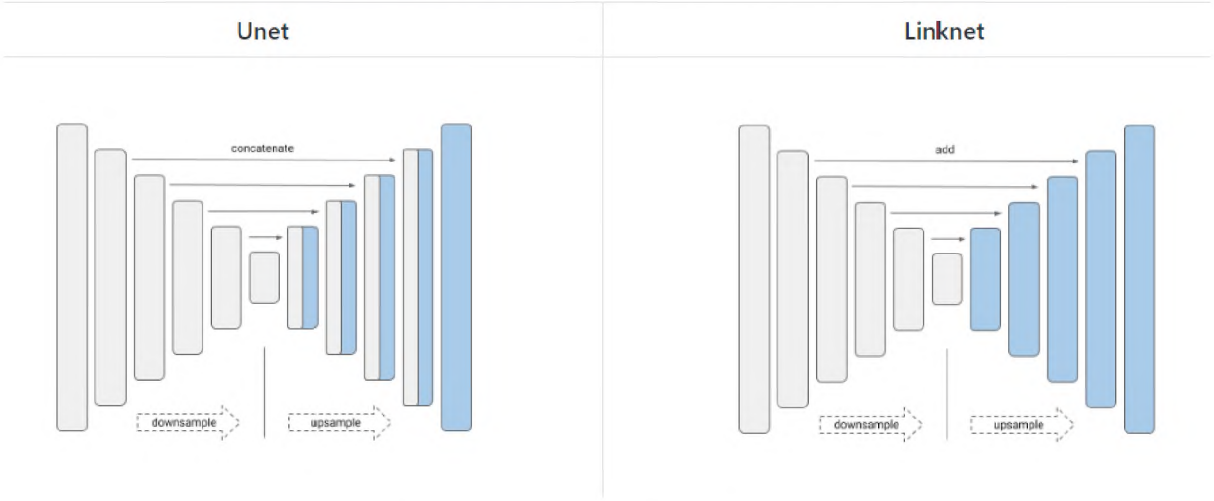
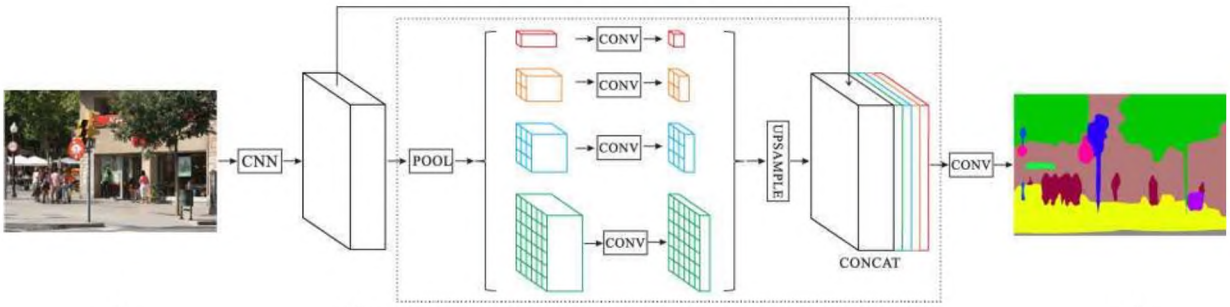


Рис. 2.16 – Архітектура LinkNet



а) б) в) г)

Рис. 2.15 – Структура PSPNet:

а) – вхідне зображення;

б) – карта характеристик;

в) – модуль об’єднання пірамід;

г) – сформована маска.

Карта ознак на виході блоку проходить через Pooling піраміду. Тобто карта ознак проходить через кілька шарів Pooling з різним ядром. Після цього кожен шар окремо проходить через згорткові шари та повертається відповідна

розмірність відповідного шару (Upsampling, Conv2DTranspose), а вже потім всі шари об'єднуються (Concatenate) і подаються на вихідний згортковий шар.

Згідно з [12], у ході проведених досліджень для реалізації сегментації ділянок лісу як базові розглядаються мережі U-Net та PSPNet.

При цьому, у моделі U-Net архітектура покращена шляхом використання звужуються блоків згортки для захоплення контексту, блоків згортки для локалізації, що розширюються, а також прямих зв'язків між блоками згортки на однакових рівнях. У PSPNet комбінуються ознаки з кількох масштабів без значного збільшення кількості параметрів. Це дозволяє вивчати більш загальний контекст.

Проте, реалізації моделей глибокого навчання найчастіше потрібно як значний обсяг навчальних даних, а й відповідні обчислювальні потужності. Останній фактор може грати вирішальну роль при їх впровадженні на роботи з безпілотних роботизованих платформах або інших рішень AI IoT.

Як наслідок, доцільно орієнтуватися на зображення з низькою роздільною здатністю. При цьому властивості різних типів CNN, що використовуються, впливають на рівень якості одержуваних результатів. Проведений аналіз існуючих робіт свідчить про домінування досліджень U-Net, тоді як PSPNet потребує детальнішого вивчення.

2.5 Деталізація архітектури PSPNet

Сучасні системи синтаксичного аналізу сцен в основному засновані на повністю згортковій мережі (FCN) [24]. Методи на основі глибоких згорткових нейронних мереж (CNN) покращують розуміння динамічних об'єктів, але все-таки стикаються з проблемами, пов'язаними з різноманітними сценами та необмеженим словниковим запасом. Один приклад показаний у першому рядку рис. 2.12, де човен помилково прийнято за автомобіль. Ці помилки виникають через схожий зовнішній вигляд об'єктів. Але при перегляді

зображення в контексті, що передує тому, що сцена описана як елінг біля річки, має бути правильне передбачення.

Для точного сприйняття сцени графік знань спирається на попередню інформацію про контекст сцени. Виявлено, що основною проблемою сучасних моделей на основі FCN є відсутність стратегії для використання підказок глобальних категорій сцен. Раніше для розуміння складної сцени, щоб отримати глобальну характеристику рівня зображення, широко використовувалося об'єднання Pyramid Pooling [25], де просторова статистика забезпечує хороший дескриптор для загальної інтерпретації сцени. Мережа об'єднання Pyramid Pooling [26] ще більше розширює можливості.

На відміну від цих методів, щоб увімкнути відповідні глобальні функції, пропонується використовувати нейронну мережу PSPNet (див. рис. 2.15). Маючи вхідне зображення (рис. 2.15.а), спочатку використовується CNN для отримання карти ознак останнього згорткового шару (рис. 2.15.б), потім застосовується модуль синтаксичного аналізу піраміди для збору різних уявлень субрегіонів, після чого виконуються шари підвищуючої дискретизації та конкатенації для формування остаточного представлення функції, яке несе як локальну, і глобальну контекстну інформацію (рис. 2.15.в). Нарешті, подання передається у шар згортки, щоб отримати остаточну маску для кожного пікселя (рис. 2.15.г).

Крім традиційного розширеного FCN для передбачення пікселів розширюється функціонал лише на рівні пікселів до спеціально розробленої глобальної піраміди. Локальні та глобальні підказки разом роблять остаточний прогноз більш надійним. PSPNet дає перспективний напрямок для завдань прогнозування на рівні пікселів, які можуть навіть принести користь стерео зіставленню на основі CNN, оптичного потоку, оцінки глибини та ін. у наступній роботі. Таким чином, PSPNet підвищує продуктивність ідентифікації об'єктів та матеріалів з відкритим словником під час аналізу складних сцен.

CNN розмір рецептивного поля може приблизно вказувати, наскільки часто використовується контекстну інформацію. При цьому емпіричне рецептивне поле CNN набагато менше теоретичного, особливо на шарах високого рівня. Як наслідок, багато мереж недостатньо враховують важливі глобальні краєвиди. Об'єднання глобальних середніх значень є гарною базовою моделлю як глобальний контекстуальний апріорний базис, який зазвичай використовується в задачах класифікації зображень [25, 26]. У [27] він був успішно застосований до семантичної сегментації. Але щодо зображень складних сцен в ADE20K [16], цієї стратегії недостатньо для охоплення необхідної інформації. Пікселі у цих зображеннях сцени анотовані щодо багатьох речей та об'єктів. Безпосереднє злиття їх у єдиний вектор може призвести до втрати просторового відношення та викликати неоднозначність. Інформація про глобальний контекст разом із контекстом субрегіону корисна у цьому відношенні виявлення відмінностей різних категорій. Більш потужне уявлення могло бути об'єднане інформацією з різних субрегіонів з цими рецептивними полями. Аналогічний висновок було зроблено у класичних роботах [28, 29] щодо класифікації сцен/зображень.

У [29] карти об'єктів на різних рівнях, генеровані Pyramid Pooling, були остаточно згладжені та об'єднані для подачі повнозв'язного шару для класифікації. Цей глобальний апріор призначений для усунення обмеження фіксованого розміру CNN для класифікації зображень. Щоб зменшити втрату контекстної інформації між різними субрегіонами, можна використовувати ієрархічний глобальний апріорний рівень, що містить інформацію з різним масштабом і різний для різних субрегіонів. Він називається модулем Pyramid Pooling для попередньої побудови глобальної сцени на карті ознак останнього шару CNN (див. рис. 2.15).

Модуль об'єднання пірамід поєднує функції у чотирьох різних масштабах піраміди. Найбільш грубий рівень, виділений червоним, є глобальне об'єднання створення вихідних даних одного біта. Наступний рівень піраміди поділяє карту об'єктів на різні субрегіони та формує об'єднане

уявлення для різних позицій. Вихідні дані різних рівнів у модулі об'єднання пірамід містять карту об'єктів із різними розмірами. Щоб зберегти вагу глобальної функції, використовується шар згортки 1x1 після кожного рівня піраміди, щоб зменшити розмір представлення контексту до 1/N від вихідного, де розмір рівня піраміди дорівнює N. Нарешті різні рівні функцій об'єднуються в остаточний глобальний Pyramid Pooling. Зазначено, що кількість рівнів піраміди та розмір кожного рівня можуть бути змінені. Вони пов'язані з розміром карти об'єктів, що завантажується у шар Pyramid Pooling. Структура абстрагує різні субрегіони, за кілька кроків приймаючи ядра пула різного розміру. Таким чином, багатоступінчасті ядра повинні підтримувати адекватну різницю в поданні. Наприклад, модуль Pyramid Pooling є 4-рівневим з розмірами осередків 1x1, 2x2, 3x3 та 6x6 відповідно.

Остаточний розмір картки ознак становить 1/8 вхідного зображення, як показано на рис. 2.15.б. У верхній частині карти використовуємо модуль поєднання пірамід (рис. 2.15.в), для збору контекстної інформації. Використовуючи 4-рівневу піраміду, ядра об'єднання охоплюють усі зображення, його половину та невеликі частини. Вони зливаються як глобальні попередні.

Потім поєднується апріорна карта з вихідною картою об'єктів у заключній частині (рис. 2.15.в). За ним слідує шар згортки для створення остаточної карти передбачення (рис. 2.15.г).

Таким чином, PSPNet надає ефективний глобальний контекстний апріор для аналізу сцени на рівні пікселів. Модуль Pyramid Pooling може збирати рівні інформації більш репрезентативні, ніж глобальний пул. З погляду обчислювальних витрат PSPNet ненабагато збільшує їх у порівнянні з вихідною розширеною мережею FCN. Під час наскрізного навчання модуль глобального об'єднання пірамід і локальна функція FCN можуть бути оптимізовані одночасно.

Крім того, модулю Pyramid Pooling необхідно аналізувати як об'єкти, так і інші елементи в сцені, що робить його складнішим за інші набори даних. Для

оцінки використовуються як піксельна точність (Pixel Acc.), так і середнє значення перетину за класами по об'єднанню (Mean IoU).

Щоб оцінити PSPNet, розробниками проводилися експерименти з декількома налаштуваннями, включаючи типи об'єднання максимальних та середніх значень, об'єднання лише однієї глобальної ознаки або 4-рівневих ознак, зі зменшенням розмірності та без нього після операції об'єднання та до конкатенації. Середній пул працює краще, ніж максимальний пул у всіх налаштуваннях. Об'єднання в пул з аналізом піраміди перевершує використання глобального пулу. Зі зменшенням розміру продуктивність ще більше підвищується. З запропонованої PSPNet найкраще налаштування дає результати 41,68/80,04 з погляду середнього IoU та Pixel Acc. (%). Введені допоміжні втрати допомагають оптимізувати процес навчання, не впливаючи при цьому на навчання в основній галузі.

Для подальшого аналізу PSPNet автори проводили експерименти різної глибини попередньо навченої мережі типу ResNet. Як показано на рис. 2.16, при тому ж налаштуванні збільшення глибини ResNet з 50 до 269 може покращити показник (середнє значення IoU + Pixel Acc.)/2 (%) з 60,86 до 62,35 з абсолютним покращенням на 1,49.

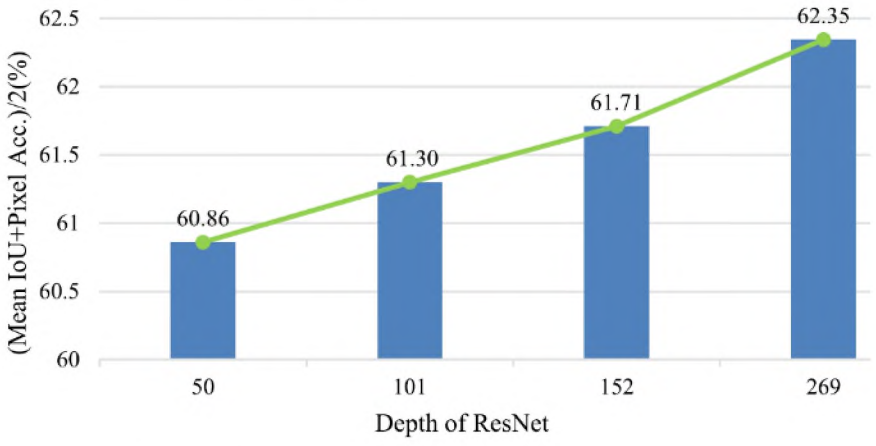
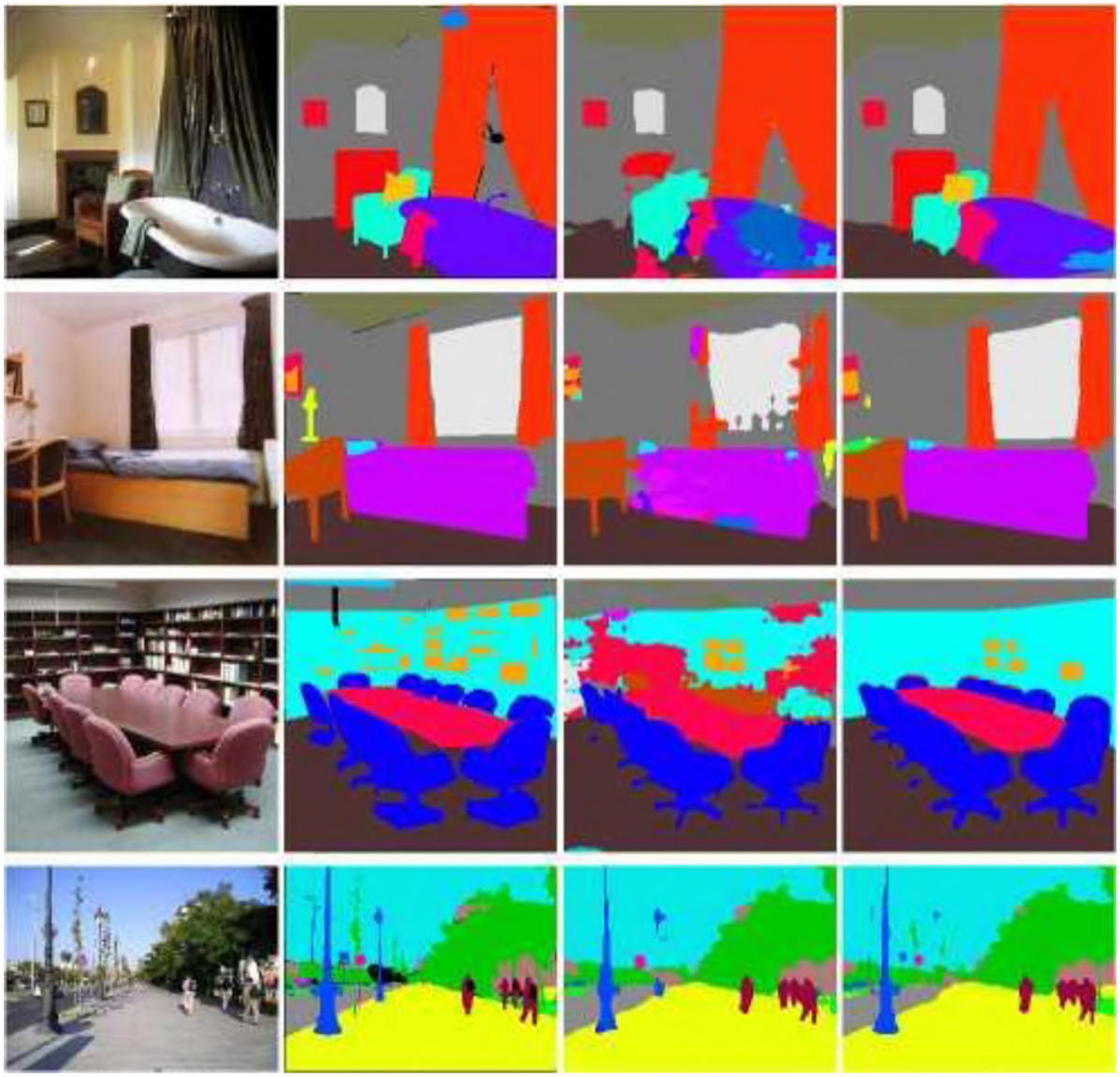


Рис. 2.16 – Залежність мережі від глибини (кількості прихованих шарів) при використанні набору валідації з одномасштабним входом

Докладні оцінки PSPNet, попередньо навчальні для моделей ResNet різної глибини, перераховані в табл. 2.1. Як показано на рис. 2.17.в, PSPNet

вирішує загальні проблеми FCN. На рис. 2.17 показані ще кілька результатів синтаксичного аналізу перевірочного набору ADE20K. Результати по мережі PSPNet містять більш точні та докладні структури відносно базового рівня.



а) б) в) г)

Рис. 2.17 – Робота PSPNet на датасеті ADE20K:

а) – зображення;

б) – справжній фон;

в) – маска сформована базовою моделлю;

г) – маска сформована PSPNet.

PSPNet також задовільно працює з семантичною сегментацією (рис. 2.18). Експерименти проводилися на набір даних сегментації PASCAL VOC 2012 [30], який містить 20 категорій об'єктів і один фоновий клас.

Таблиця 2.1 – Більш глибока попередньо навчена модель отримує вищу продуктивність

Варіант архітектури мережі	Mean IoU (%)	Pixel Acc.(%)
PSPNet(50)	41.68	80.04
PSPNet(101)	41.96	80.64
PSPNet(152)	42.62	80.80
PSPNet(269)	43.81	80.88
PSPNet(50)+MS	42.78	80.76
PSPNet(101)+MS	43.29	81.39
PSPNet(152)+MS	43.51	81.38
PSPNet(269)+MS	44.94	81.69

а) б) в) г)

Рис. 2.18 – Робота PSPNet на датасеті PASCAL VOC 2012:

а) – зображення;

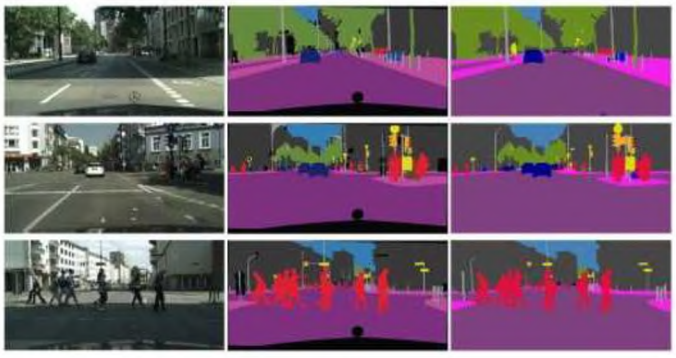
б) – справжній фон;

в) – маска сформована базовою моделлю;

г) – маска сформована PSPNet.

Наступні процедури [31, 32], автори використовували доповнені дані з анотацією [33], отримали 10582, 1449 і 1456 зображень для навчання, перевірки та тестування.

При цьому PSPNet порівнювався з попередніми найбільш ефективними методами набору тестів на основі двох налаштувань MS-COCO [34]: з попереднім навчанням або без нього. PSPNet перевершує попередні методи при обох параметрах. При навчанні тільки з даними VOC 2012 досягається точність 82,6% у всіх 20 класах. Коли PSPNet попередньо навчений з набором даних MS-COCO, точність досягає 85,4%, при цьому 19 із 20 класів отримують найвищу точність. Цікаво, що PSPNet, навчена лише на даних VOC 2012, перевершує існуючі методи, навчені за допомогою попередньо навченої моделі MS-COCO. Згідно рис. 2.18, для «корів» у першому рядку базова модель розглядає їх як «кінь» та «собаку», тоді як PSPNet виправляє ці помилки. Для слів «літак» і «стіл» у 2-му та 3-му рядках PSPNet знаходить відсутні частини. Для «людини», «пляшки» і «рослини» в наступних рядках PSPNet добре працює з цими класами об'єктів невеликого розміру на зображеннях порівняно з базовою моделлю. Також досліджувалась робота PSPNet (рис. 2.19) на датасеті Cityscapes [35].



а) б) в)

Рис. 2.19 – Приклади результатів PSPNet для набору даних Cityscapes:

а) – зображення;

б) – справжній фон;

в) – маска сформована PSPNet.

Це набір даних розуміння семантичної міської сцени. Він містить 5000 високоякісних анотованих зображень на рівні пікселів, зібраних у 50 містах у різну пору року. Зображення розділені на набори з номерами 2975, 500 та 1525 для навчання, перевірки та тестування. Він визначає 19 категорій, які містять як речі, і об'єкти.

З іншого боку, 20000 зображень з грубими анотаціями надаються для двох параметрів порівняння, тобто навчання тільки з точними даними або з точними та грубими даними. Використовуючи для навчання як точні, і грубі дані, PSPNet дає точність 80,2 %.

Висновки до розділу 2

Розбір сцени, що базується на семантичній сегментації, є фундаментальною темою комп'ютерного зору. Щоб присвоїти кожному пікселю зображення мітку категорії. Складність аналізу сцен тісно пов'язана з різноманітністю сцен і міток. В ході досліджень визначено основні проблеми для виконання семантичної сегментації.

Глибоке машинне навчання сьогодні набуло великої популярності. Це обумовлене продуманістю архітектур нейронних мереж. Для вирішення завдань семантичної сегментації найбільш вдало підходять згорткові мережі. В роботі досліджено принцип їх роботи та визначено особливості реалізації шарів, наприклад, Dense, Conv1D, Conv2D, Conv3D, MaxPooling2D, AveragePooling2D, Conv2DTranspose. Окрема увага приділена властивостям найбільш поширених функцій активації, що використовуються в згорткових мережах (ReLu і Softmax).

Для реалізації сегментації зображень існують різні архітектури згорткових мереж нейронних мереж, серед яких слід виділити: Автокодувальник, U-Net, SegNet, LinkNet, PSPNet. При цьому властивості різних типів CNN впливають на рівень якості одержуваних результатів. З

врахуванням того, що домінують дослідження U-Net, в роботі основна увага приділяється деталізації архітектурі PSPNet та проаналізована можливість створення гібридних архітектур на цій основі.

РОЗДІЛ 3
АНАЛІЗ МОДЕЛЕЙ ГЛИБОКОГО НАВЧАННЯ АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ СЕГМЕНТАЦІЇ ЛІСУ

3.1 Формування датасету

Як відомо, при реалізації моделі глибокого навчання важливу роль відіграє формування датасету. Для його створення використовувалися дані набору Kaggle [36]. Цей набір даних було отримано з треку класифікації земного покриву DeepGlobe Challenge. У [36] зображення в наборі даних були змінені на зображення розміром 256x256, щоб створити більше зразків зображень. Їхня кількість була збільшена до 5108 штук.

У цій роботі під час досліджень використовувалися ресурси Google Colab Pro+ з GPU Tesla V100-SXM2-16GB та Keras [37].

При цьому для відпрацювання технології навчання виконана модифікація зазначеного набору даних шляхом переходу до зображень 128x128. В результаті замість вихідного розміру 1,2 ГБ при форматі 256x256 розмір перетисненого набору становив 476,14 МБ, а також вдалося збільшити batch, який дорівнював 16. При підготовці датасету створювалися окремо дві папки. В одній з них є вихідні зображення, а в іншій – файли масок. У кореневий каталог розміщувався файл опису міток наступної структури:

```
# label:color_rgb:parts:actions
Forest:0,0,0::
Field:255,255,255::
```

Тренувальна вибірка формувалася на основі повного датасету, розділеного у співвідношенні 70:30.

При цьому, відсоток простору у тренувальній вибірці становив: 38% – перший клас (поле) та 62% – другий клас (ліс), а відсоток простору у перевірочній вибірці становив: 40% – перший клас (поле) та 60 % – другий клас (ліс). Загалом у тренувальній вибірці 1-ий клас був представлений на 3565 фото, а 2-й клас – на 3570. У перевірочній вибірці на частку 1-го класу довелося

1527 знімків, а 2-го класу – 1529. На жаль, деякі маски в датасеті були зроблені досить грубо і не завжди відповідали реальній картині розподілу лісонасаджень. Тому додатково виконувався вхідний контроль датасету відповідність масок. Для оптимізації функції втрат використовувався оптимізатор Адам.

3.2 Оцінка точності синтезованих архітектур PSPNet

У першому етапі оцінювалося кілька варіантів архітектури PSPNet (наприклад, рис. 3.1 і 3.2).

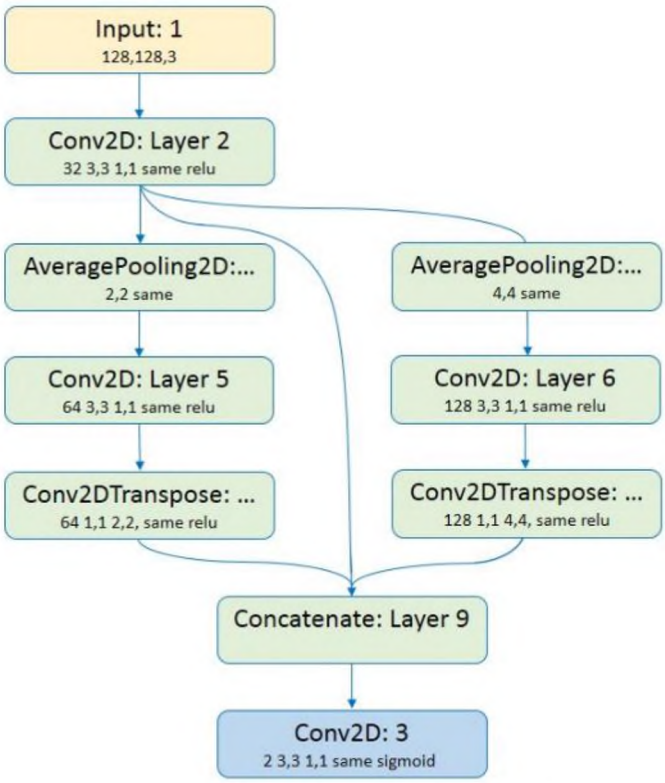


Рис. 3.1 – Архітектура «малої» PSPNet

На її основі створювалися модифікації, які дозволяли оцінити вплив архітектури на отриманий результат. Наприклад, поліпшення точності можна досягти, якщо: відразу після входу додати шар BathNormalization; усі шари MaxPool2D замінити на відповідні за розміром AveragePool2D та додати їх

одразу після входу або BathNormalization.

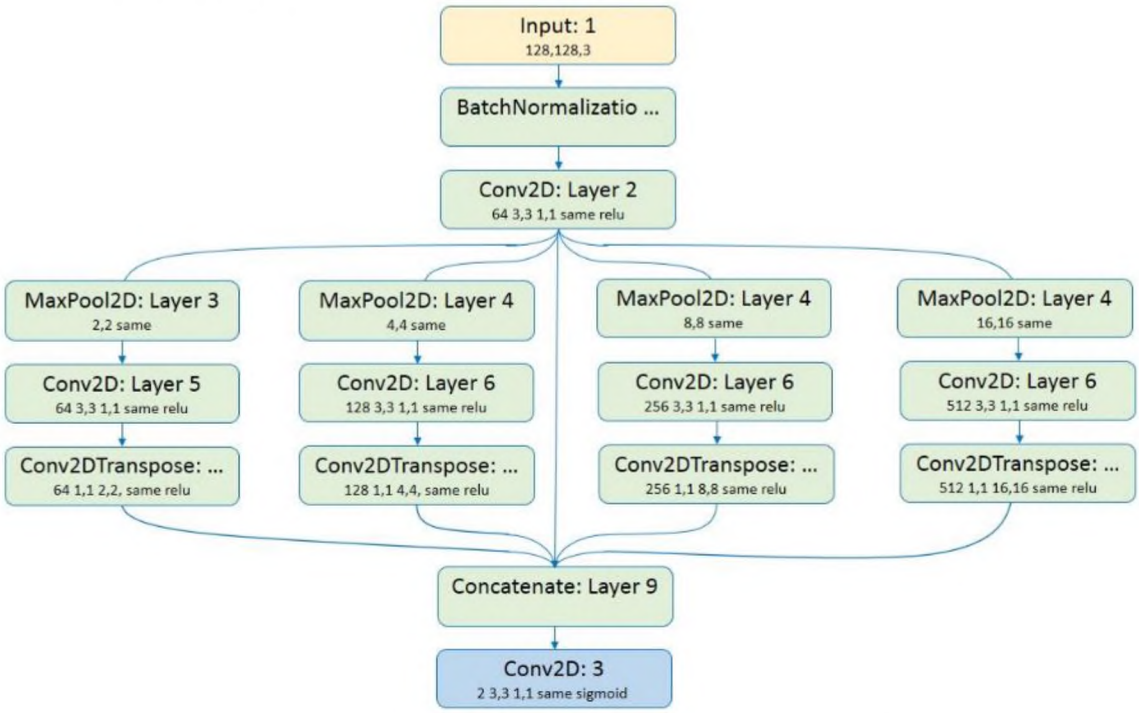


Рис. 3.2 – Архітектура «великої» PSPNet

Як відомо, MaxPool2D забезпечує зменшення розміру зображення. Він приймає параметр, званий розміром пула (ядра) і отримує перші p x p пікселів зображення. Потім він знаходить максимальне значення цих значеннях пікселів і зберігає його як перше значення пікселя для вихідного зображення. Шар продовжує процес для всього зображення, переміщаючись по зображенню, і виводить зображення з тими самими даними більш стислій формі. У свою чергу AveragePool2D замість максимального використовує середнє арифметичне значення. У ході навчання було отримано максимальну перевірочну точність на 40 епохах становила 77,2% при кроці 0,001 (рис. 3.3). Деякі результати використання PSPNet представлено на рис. 3.4.

На наступному етапі досліджувався вплив варіації розмірності ядер у шарах Conv2DTranspose. При цьому було встановлено, що збільшення розмірності згідно з коефіцієнтом масштабування призводить до більш рівномірного заповнення матриці даних.

У свою чергу, усунення незаповнених точок у Conv2DTranspose дозволило підвищити точність, але промальовування дрібних деталей

погіршилося. Загалом, такий підхід дозволив досягти точності 80 % на 98-ій епосі –та 92,8 % на 111-ій епосі перевірочної виборки (крок – 0,001).

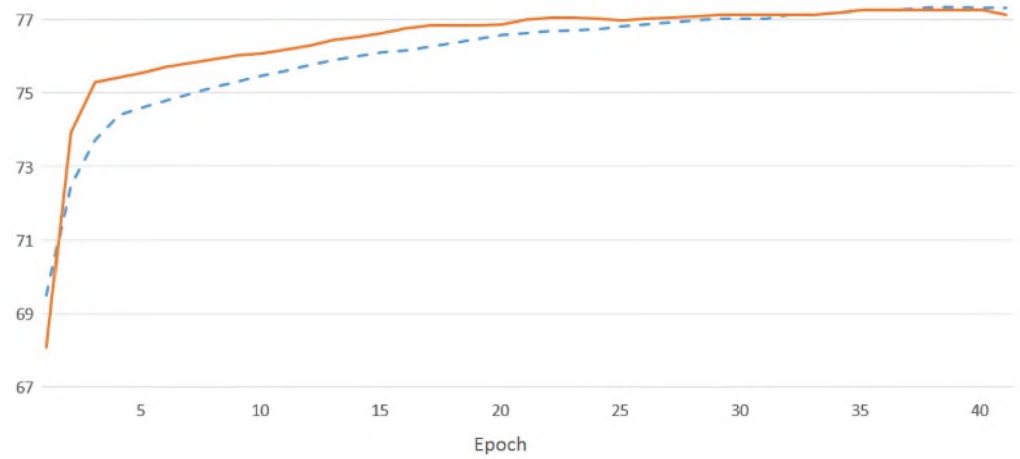


Рис. 3.3 – Результати навчання (суцільна лінія – перевіркова вибірка; пунктирна лінія – тренувальна вибірка)

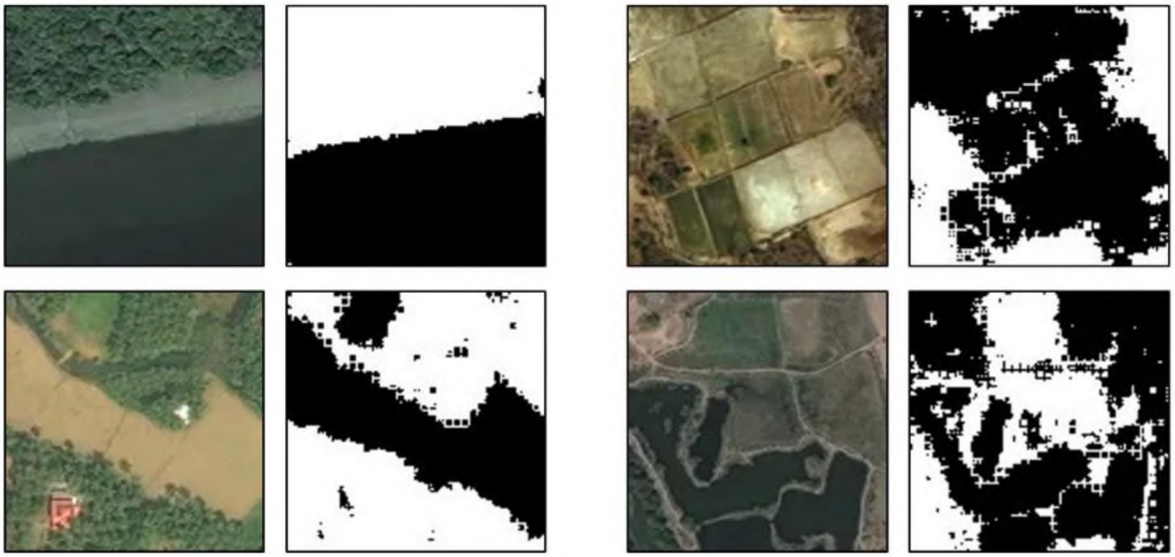


Рис. 3.4 – Результати використання моделей глибокого навчання сегментації лісу з урахуванням архітектури PSPNet

Таким чином, можна зробити висновок, що потрібно збільшити заповнюваність порожнин у Conv2DTranspose, але не піднімати до 16 або навіть обмежити на рівні 8 (4). Тобто, прийнятна точність зберігається при поверненні величин ядер у Conv2DTranspose до значень 1x1 і 2x2, тоді як у вихідній схемі використовувалися тільки ядра 1x1 (рис. 3.5).

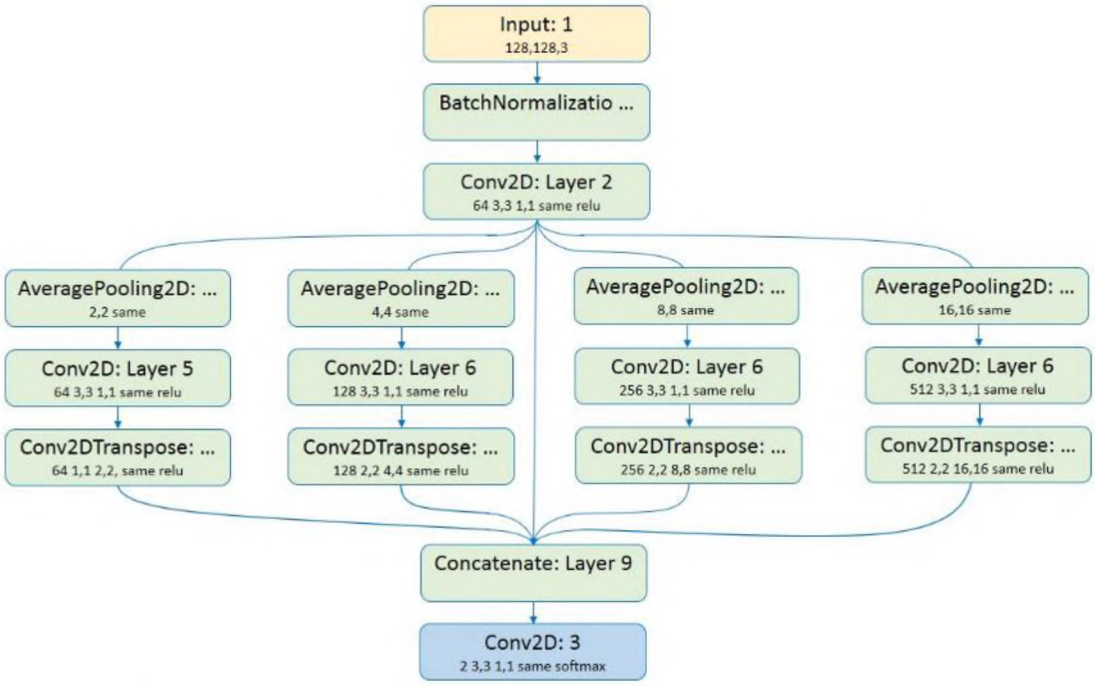


Рис. 3.5 – Архітектура PSPNet з ядрами в Conv2Dtransp 1x1 і 2x2

Наступним етапом досліджень стала модифікація архітектури PSP шляхом заміни шарів Conv2DTranspose на UpSampling з тим самим множником (рис. 3.6).

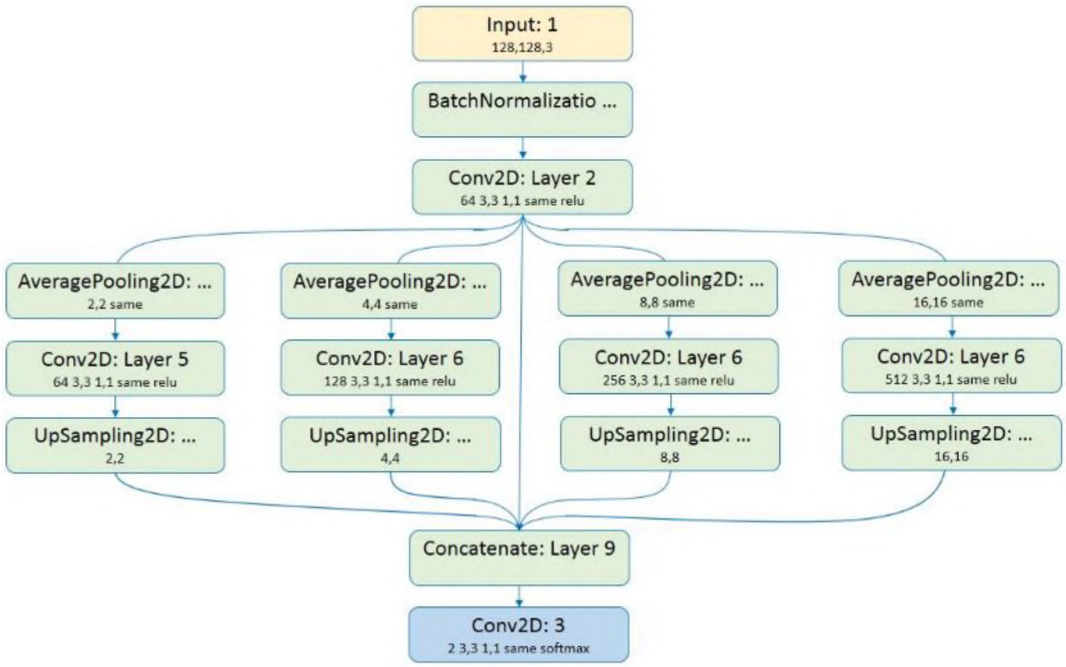


Рис. 3.6. – Заміна шарів Conv2Dtranspose на Upsampling з тим самим множником

Це два поширені типи шарів, які можна використовувати для збільшення розмірів масивів. Як відомо, Conv2DTranspose виконує підвищуючу дискретизацію та згортку (транспонована згортка).

При цьому теоретично він може призводити до появи артефактів «шахової дошки». UpSampling2D схожий на пул, у якому він повторює рядки та стовпці вхідних даних. Формально, перехід на UpSampling видав найкращі показники точності, майже 80 % вже на 20-ій епосі (рис. 3.7). Але така архітектура PSPNet дає більше візуальних розбіжностей і суворо слідує помилковим маскам, не виділяючи дрібні деталі (рис. 3.8).

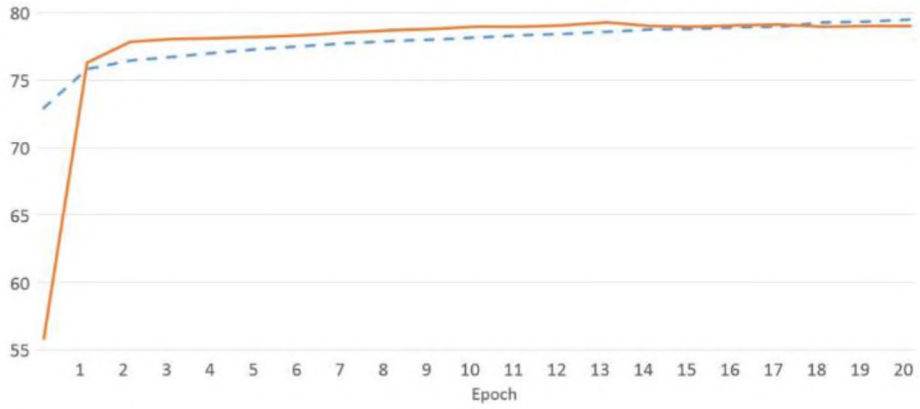


Рис. 3.7 – Результати навчання моделі PSPNet з Upsampling (суцільна лінія – перевірна вибірка; пунктирна лінія – тренувальна вибірка)

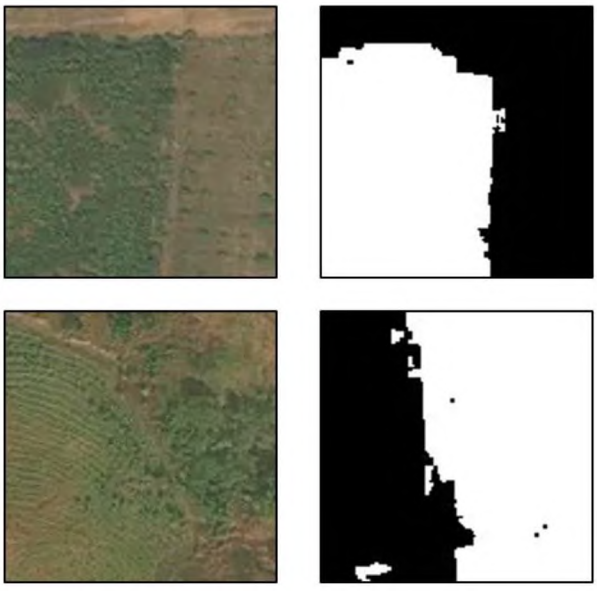


Рис. 3.8 – Результати використання моделей глибокого навчання для сегментації лісу на базі архітектури PSPNet із Upsampling

Це пояснюється тим, що UpSampling, на відміну від Conv2DTranspose, не містить ваги для навчання, і тому Conv2DTranspose більш адаптивні для вирішення задачі сегментації.

3.3 Порівняльна оцінка точності мереж на основі архітектур U-Net та PSPNet

На завершення, у роботі було проведено аналіз впливу на підсумковий результат розробленої архітектури U-Net та її порівняльна оцінка з PSPNet. Приклад архітектури «малої» U-Net представлений на рис. А.1. У ході досліджень було отримано такі результати: «мала» U-Net дає точність на тренувальній вибірці – 99% та на перевірочній – 80%; «велика» U-Net дає точність на тренувальній вибірці – 87 % та на перевірочній – 70 %.

Застосування складніших архітектур Unet++ (рис. А.2) і $Unet^2$ (рис. А.3) супроводжується швидким їх перенавчанням і дозволяє істотно підняти точність навчання. Зокрема, на вказаному модифікованому датасеті Unet++ дозволила досягти максимальної точності 79,7 %, а $Unet^2$ – 80,8 %. При цьому в архітектурі $Unet^2$ у міжкаскадних з'єднаннях замість Maxpooling використовувалися шари AveagePooling, а замість Upsampling – шари Conv2Dansopse.

В цілому, отримані результати дозволяє зробити висновок, що U-Net промальовує якісніше забарвлення, ніж мала PSPNet. Але, своєю чергою, PSPNet на розглянутому датасеті працює точніше. До певної міри PSPNet долає основний недолік структури FCN для семантичної сегментації зображень без урахування контекстної інформації, тим самим витягуючи повне уявлення ознак і додатково підвищуючи точність сегментації.

3.4 Техніко-економічне обґрунтування прийнятих рішень

Проведені дослідження свідчать про відсутність на ІТ-ринку готових рішень подібних програмну продукту, що розглядається в роботі. Найбільш близьким слід вважати інструментарій сегментації та класифікації розділу Spatial Analyst ГІС ArcGIS PRO (США). Однак, операції щодо сегментації ділянок лісу повинен виконувати оператор ГІС в ручному режимі для кожного зображення. Це свідчить про відсутність повної автоматизації даного процесу, яке може забезпечити нейронні мережі, що розглянуті в роботі та забезпечують точність 92,8 % (PSPNet) і 80,2 % (типу U-Net).

Якщо замовляти розглянутий в роботі програмний продукт з подібним функціоналом у ІТ-компанії (підприємства), то доцільно визначити орієнтовну його вартість. Визначальним чинником вартості розробки нейронної мережі є трудомісткість. Розглянемо типові роботи, які проводяться при розробці та інтеграції нейронної мережі.

1. Створення ПЗ для розмітки навчальної вибірки.
2. Збір матеріалу для навчальної вибірки.
3. Розмітка навчальної вибірки.
4. Розробка архітектури нейронної мережі.
5. Навчання нейронної мережі, формування ваг.
6. Створення інфраструктури введення даних та виведення результатів роботи нейронної мережі.
7. Інтеграція нейронної мережі у продукт чи програмний стек замовника.

В оцінці трудомісткості наведених пунктів враховувалися середні статичні дані, що отримані з відкритих джерел. При цьому, пункт 7 (інтеграція в продукт або програмний стек замовника) може як не коштувати нічого (якщо замовник має достатню компетенцію для інтеграції та планує використання програмного модуля самостійно), так і перевищити в ціні всі інші пункти разом узяті (іноді замовнику потрібно розробити цілу розподілену інфраструктуру, у

якій нейронна мережа є лише певним елементом). У розробці програмного забезпечення головними виробничими витратами є оплата праці програмістів, тому насамперед сформуємо перелік робіт, які необхідно виконати для розробки та визначимо їхню трудомісткість у людино-місяцях (табл. 3.1). Фонд заробітної плати та податки, що обчислюються виходячи з його розміру, є основною статтею витрат, їх розмір повинен бути визначальним у структурі ціни розробки. Розмір зазначеного фонду безпосередньо залежить від двох факторів: рівня заробітної плати розробників та трудомісткості роботи. Як припущення вважаємо, що рівень заробітної плати програмістів від регіону до регіону відрізняється незначно, наприклад, через можливість віддаленої роботи.

Таблиця 3.1 – Трудомісткість за видами робіт

Вид робіт	Трудомісткість люд./міс.	Виконавець
Проектування	0,5	Ведучий проектувальник
Розробка ПЗ для збору зображень за ключовим словом «ліс» у пошукових системах	0,5	Інженер-програміст
Розробка ПЗ для розмітки навчальної вибірки	0,5	Інженер-програміст
Розмітка навчальної вибірки	2,0	Технік
Розробка та реалізація архітектури нейронної мережі	0,8	Ведучий інженер-програміст
Навчання нейронної мережі	0,5	Інженер-програміст
Розробка модуля отримання зображення від джерел	0,8	Інженер-програміст
Розробка графічного інтерфейсу модуля спостереження, збирання, налагодження, тестування.	1,0	Інженер-програміст

Значення середньомісячного рівня «чистої» суми заробітної плати (S_1) поставимо з урахуванням середніх зарплат фахівців достатньої кваліфікації для виконання даних робіт. В розрахунку заробітної плати (S_2) використовуються діючі на поточний момент коефіцієнти зборів: (18 + 1,5) %. Далі порахуємо витрати на оплату праці програмістів. У табл. 3.2 у колонці «витрати часу» наведено сумарні трудовитрати у людино-місяцях кожного фахівця. S_3 відповідає фактичній сумі витрат для кожного фахівця. Склавши

суми основної заробітної плати всіх фахівців, отримаємо загальні витрати на оплату праці, що необхідні для виконання роботи по створенню і впровадженню нейронної мережі для сегментації ділянок ліса – 9613,8 у.о.

У табл. 3.3 наведено безпосередньо розрахунок ціни розробки програмного забезпечення. В якості допущення статті «матеріали», «спецобладнання», єдиний соціальний внесок та «відрядження» приймаємо рівними нулю. Накладні витрати приймаємо на рівні 15 % від витрат на заробітну платню. Витрати на оплату праці робітників беремо з табл. 3.2.

Таблиця 3.2 – Витрати на заробітну плату

Виконавець	Витрати часу	S_1	S_2	S_3
Ведучий проектувальник	0,5	1700	2031,5	1015,75
Ведучий інженер-програміст	0,8	1550	1852,25	1481,8
Інженер-програміст	3,3	1350	1613,25	5323,725
Технік	2,0	750	896,25	1792,5
Разом:	6,6			9613,8

Таблиця 3.3 – Орієнтовна оцінка вартості розробки ПЗ

Найменування статей витрат	Сумма, у.о.
Матеріали	0
Спецобладнання	0
Витрати на оплату праці виконавців	9613,8
Додаткова зарплатня	0
Єдиний соціальний внесок	0
Витрати на відрядження	0
Накладні витрати	1442,07
Разом внутрішні витрати	11055,87
Витрати сторонніх організацій	0
Усього собівартість	11055,87
Прибуток	2211
Ціна без ПДВ	13267,0
ПДВ, 20%	2653,4
Ціна з ПДВ	15920

Сума перерахованих вище статей становить внутрішні витрати підприємства, що виконує роботи, разом із витратами залучених сторонніх організацій (в даному випадку, дорівнюють нулю) вони становлять собівартість розробки. Прибуток організації встановимо лише на рівні 20% собівартості. Сума прибутку та собівартості утворюють ціну розробки. Далі до ціни додається ПДВ 20% і отримуємо ціну розробки ПЗ з ПДВ – 15920 у.о.

Стосовно проведених досліджень, слід враховувати кілька припущень, а саме: використовувався готовий датасет [36], за пунктами 6 і 7 не проводилися дослідження та розробки (виходять за рамки завдання на роботу). Як наслідок, це значно впливає на трудомісткість за видами робіт (табл. 3.4 – 3.6).

Таблиця 3.4 – Трудомісткість за видами робіт при використанні готового датасету [36]

Вид робіт	Трудомісткість люд./міс.	Виконавець
Проектування	0,3	Ведучий проектувальник
Розмітка навчальної вибірки	0,1	Технік
Розробка та реалізація архітектури нейронної мережі	0,8	Ведучий інженер-програміст
Навчання нейронної мережі	0,5	Інженер-програміст
Розробка модуля отримання зображення від джерел	0,8	Інженер-програміст
Розробка графічного інтерфейсу модуля спостереження, збирання, налагодження, тестування.	1,0	Інженер-програміст

Таблиця 3.5 – Витрати на заробітну плату при використанні готового датасету [36]

Виконавець	Витрати часу	S_1	S_2	S_3
Ведучий проектувальник	0,3	1700	2031,5	609,45
Ведучий інженер-програміст	0,8	1550	1852,25	1481,8
Інженер-програміст	2,3	1350	1613,25	3710,475
Технік	0,1	750	896,25	89,625
Разом:	6,6			5891,4

В такому випадку, витрати на розробку нейронної мережі скорочуються на 6164 у.о. (38,8% від початкової суми). В цілому, загальна ціна роботи може зрости тільки за рахунок робіт, що не стосуються безпосередньо синтезу нейронної мережі – розробки ПЗ, частиною якого є нейронна мережа (аналітика, звіти, бази даних, портали, інфраструктура та робочі місця користувачів та ін.).

Таблиця 3.6 – Орієнтовна оцінка вартості розробки ПЗ при використанні готового датасету [36]

Найменування статей витрат	Сумма, у.о.
Матеріали	0
Спецобладнання	0
Витрати на оплату праці виконавців	5891,4
Додаткова зарплатня	0
Єдиний соціальний внесок	0
Витрати на відрядження	0
Накладні витрати	883,71
Разом внутрішні витрати	6775,11
Витрати сторонніх організацій	0
Усього собівартість	6775,11
Прибуток	1355
Ціна без ПДВ	8130,1
ПДВ, 20%	1626,03
Ціна з ПДВ	9756

Доцільність (як наслідок, ефективність) застосування запропонованих рішень на основі нейронних мереж буде зростати при збільшенні обсягів баз зображень при масовій оцінці великих даних. Крім того, виконання процедур з автоматизованого аналізу лісистої місцевості на основі сегментації зображень дистанційного зондування Землі може відігравати важливу роль у таких областях, як екологія дикої природи, контроль вирубок лісів, оцінка рослинного покриву та геологічне картування. При цьому можливо будувати системи аналітики на базі автоматизованих звітів щодо змін площини, яку займають ліси, з вказівкою геолокації відповідних регіонів в інтересах розширення функціоналу існуючих ГІС або SCADA [38].

Інтенсивний розвиток засобів дистанційного зондування Землі аерокосмічного базування, збільшення обсягів та інформативності аерокосмічних даних призводить до безперервного розширення кола тематичних завдань, що вирішуються на їх основі.

Подальші дослідження можуть бути спрямовані на оцінку гібридних архітектур, наприклад, з використанням у гілках з різним масштабом попередньо навчених мереж [39]. Набори пропонованих моделей надають Tensorflow, PyTorch, Keras, Caffe2. Наприклад, у Keras ця номенклатура становить майже 4 десятки мереж [40]. Крім того, слід оцінити можливості

використання TensorFlow Lite [41] для розгортання розроблених архітектур на мобільних, вбудованих та периферійних пристроях тощо [42].

Висновки до розділу 3

Для адекватної оцінки якості моделі глибокого навчання важливу роль потрібно ретельно підготувати дані для тренування та перевірки нейронної мережі. В роботі для створення датасету використовується ресурс з вебпорталу Kaggle. Для збільшення кількості зображень змінені їх розміри. Для проведення навчання використовувався Google Colab Pro+ з GPU Tesla V100-SXM2-16GB та Keras. Тренувальна вибірка формувалася на основі повного датасету, розділеного у співвідношенні 70:30.

В ході досліджень виконувалась оцінка точності запропонованих архітектур PSPSNet, а також проводився їх порівняльний аналіз синтезованими мережами U-Net. Отримані результати підтвердили висунуті в роботі теоретичні положення щодо властивостей вказаних архітектур.

В рамках техніко-економічного обґрунтування прийнятих рішень проаналізовано витрати на типові роботи, які проводяться при розробці та інтеграції нейронної мережі. При цьому, зроблений висновок, що ефект від проведення семантичної сегментації буде зростати при збільшенні обсягів баз зображень при масовій оцінці великих даних. На основі проведених досліджень визначено пріоритетні напрями застосування нейронних мереж PSPNet і U-Net.

ВИСНОВКИ

Інтенсивний розвиток засобів дистанційного зондування Землі аерокосмічного базування, збільшення обсягів та інформативності аерокосмічних даних призводить до безперервного розширення кола тематичних завдань, що вирішуються на їх основі. Аналіз лісистої місцевості на основі нейромережевої сегментації зображень дистанційного зондування Землі може також відігравати важливу роль у таких галузях, як екологія дикої природи, контроль вирубок лісів, оцінка рослинного покриву та геологічне картування.

В ході досліджень були запропоновані та досліджені різні архітектури PSPNet та U-Net для сегментації лісу на зображеннях дистанційного зондування Землі.

Основні вдосконалення запропонованих архітектур базуються на використанні шарів BathNormalization, заміні шарів MaxPool2D на AveragePooling2D, зміні блоків Conv2DTranspose на UpSampling2D тощо.

Для навчання нейронних мереж використовувався модифікований набір даних із зображень 128x128 на основі набору даних від вебпорталу Kaggle. В результаті вдосконалення архітектури було отримано максимальну точність сегментації 92,8 % (PSPNet) і 80,2 % (U-Net).

В процесі дослідження отримано такі результати:

– моделі глибокого навчання архітектур згорткових нейронних мереж типу U-Net і PSPNet для завдань сегментації лісу на зображеннях з низьким розрізненням;

– порівняльна оцінка точності мереж на основі архітектур U-Net та PSPNet.

Вони можуть бути використані для подальших досліджень за даною тематикою та при проектуванні інтелектуальних інформаційних систем, а також в інтересах розширення функціоналу існуючих ГІС або SCADA. Одним з таких напрямів є масштабування отриманих результатів на більш складні за

структурою гібридні архітектури нейронних мереж для вирішення завдань сегментації зображень, а також визначити напрями розгортання запропонованих нейронних мереж при реалізації прикордонних та туманних обчислень систем AI IoT.