

Міністерство освіти і науки України
Кам'янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп'ютерних наук

Дипломна робота
магістра

з теми: **“ДОСЛІДЖЕННЯ МЕТОДІВ СЕМАНТИЧНОЇ ОБРОБКИ
ЗОБРАЖЕНЬ”**

Виконала: студентка 2 курсу,
групи KN1-M21,
спеціальності 122 Комп'ютерні науки
Сокальська Діана Сергіївна

Керівник: старший викладач кафедри,
кандидат технічних наук, доцент
Слободянюк Олександр Васильович

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ПОНЯТТЯ СЕМАНТИЧНОЇ ОБРОБКИ ЗОБРАЖЕННЯ	7
1.1. Сприйняття зображення комп'ютером	7
1.2. Поняття обробки зображення	10
1.3. Сегментація зображення	12
1.4. Математична постановка задачі	17
1.5. Завдання сегментації	19
1.6. Проблематика	20
1.7. Застосування	22
РОЗДІЛ 2. ОГЛЯД МЕТОДІВ СЕМАНТИЧНОЇ ОБРОБКИ ЗОБРАЖЕНЬ	28
2.1. Підготовка даних	29
2.2. Пороги	32
2.3. Алгоритм K-means	34
2.4. Сегментація країв	35
2.5. Методи на основі стиснення	36
2.6. Нейронна мережа	38
2.7. CNN(convolutional neural network)	39
2.8. Повністю згорточні мережі	43
2.9. U-Net	44
2.10. SegNet	46
РОЗДІЛ 3. Дослідження ефективності методів	50
3.1. Обґрунтування вибору ПЗ	50
3.1.1. Google Collaboratory	54
3.1.2. TensorFlow	55
3.1.3. Keras	56
3.1.4. NumPy	57
3.2. ImageNet	58
3.3. Метрики оцінки ефективності семантичної обробки	59

3.4. Ефективність обробки зображень з будівельними об'єктами із аерофотознімків	63
3.5. Ефективність обробки зображень з водними об'єктами	67
ВИСНОВОК	70
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	71

ВСТУП

У сучасному світі ми постійно взаємодіємо з цифровими зображеннями. Зображення стало одним з основних типів інформації, який допомагає нам легко зберігати інформацію про навколишній світ. Це й не дивно, адже, як відомо, понад 80% інформації людина сприймає за допомогою зору. За даними агентства маркетингових досліджень InfoTrends, у 2023 році пристрої зберігатимуть близько 7,4 трильйонів цифрових зображень [1]. Здебільшого такі зображення мають суто творчий та естетичний характер, але навколо нас все ще є багато областей, і зображення містять інформацію, яку потрібно ще певним чином опрацювати. Будь то наука, медицина, криміналістика і т. д. Питання яке постало давно все ще залишається актуальним і в наш час, адже зображення є природним засобом спілкування людини і машини в будь-яких системах обробки, аналізу та управління.

Розвиток технологій дозволяє створювати системи комп'ютерного зору, здатні не тільки доповнювати, але в деяких випадках і замінювати людину в задачах розпізнавання образів у різних галузях науки і техніки.

Сьогодні комп'ютерний зір широко використовується в найрізноманітніших галузях промисловості. Наприклад, він набув широкого застосування в галузі медицини, де з його допомогою можна визначити діагноз або виявити проблему на основі рентгенівського знімка чи будь-якого іншого фото- та відеоматеріалу, у багатьох випадках комп'ютери помічають те, що не помічають навіть найдосвідченіші лікарі. Вони виступають своєрідними помічниками, чий «технічний» висновок підтверджує гіпотезу лікаря або дає привід для більш глибоких досліджень.

Інша галузь — фізика, де комп'ютерний зір набув широкого поширення для вимірювання та аналізу теплового або електромагнітного випромінювання. Але найчастіше комп'ютерний зір йде рука об руку зі штучним інтелектом і машинним навчанням. Найпоширенішими сферами є

розпізнавання образів і методи навчання. За допомогою штучного інтелекту можна проводити такі вимірювання та виконувати завдання, які раніше неможливо було уявити. Технології машинного навчання в комп'ютерному зорі швидко покращують розуміння комп'ютером високорівневих зображень, відкриваючи тим самим нові можливості для завдань, які раніше вимагали ручного втручання людини

З 2010 року ми спостерігаємо прискорення вдосконалення методів і технологій глибинного навчання. Завдяки глибокому навчанню тепер ми можемо програмувати суперкомп'ютери на навчання, самовдосконалення з часом і надання деяких із цих бізнес-можливостей у вигляді онлайн-додатків, наприклад хмарних додатків. Все це дало різкий поштовх для розвитку комп'ютерного зору разом зі штучним інтелектом. Багато великих компаній, таких як Facebook або Microsoft, витрачають багато часу та ресурсів на вивчення цієї сфери. Вони розробляють велику кількість програмного забезпечення або просто вдосконалюють існуючі програми за допомогою комп'ютерного зору для вирішення найрізноманітніших завдань.

Область цифрової обробки зображень фокусується на перетворенні одного зображення на інше, незалежно від його змісту, тоді як область комп'ютерного зору фокусується на перетворенні тривимірної сцени, спроектованої на одне або кілька зображень. Але розв'язання задачі безпосередньо залежить від припущень про зміст сцени. Першим етапом обробки зображення є сегментація, оскільки необхідно розрізняти різні об'єкти, що з'являються на сцені.

Завдання семантичної сегментації є високорівневим завданням обробки зображень, що належить до групи задач машинного розпізнавання. Вона є більш складною, ніж завдання класифікації зображень та пошуку об'єктів, що обумовлено не лише необхідністю визначення класів об'єктів, а й виявлення їхньої структури, правильного виділення частин об'єктів на зображенні.

Саме тому для дослідження було обрано найбільш широко застосовані методи обробки зображення і сегментації для подальшого аналізу і порівняння на основі якого можна буде зробити висновок щодо доцільності і результативності застосування до певної групи зображень.

Предметом дослідження є результати отримані після застосування обробки до кожної з обраних груп зображень для встановлення якості роботи методів за обраними критеріями.

Об'єктом дослідження методи обробки зображень та отримані сегменти зображень і те, наскільки вони відрізняються від реальних та їх порівняльний аналіз.

Мета дослідження полягає в тому, щоб розглянути та проаналізувати найбільш поширені способи семантичної обробки (сегментації) зображень. Дослідити продуктивність методів, вибраних для роботи на кількох наборах даних, для подальшого порівняння ефективності їх застосування.

Практичне значення одержаних результатів полягає в встановленні доцільності застосування методів до семантичної обробки зображень.

Робота складається з трьох розділів. У першому розділі з'ясуємо як саме комп'ютер сприймає зображення, що розуміємо під визначенням семантичної обробки зображення, які є основні елементи обробки зображення на підготовчому етапі, а також визначили головну задачу даної області, проблематику та актуальність. Другий розділ присвячений огляду існуючих методів семантичної обробки зображення та їх поділу, а також критерії оцінки роботи методів. У третьому розділі обґрунтовано вибір самих методів для дослідження методів обробки зображень та програмного забезпечення для їх реалізації та проведення порівняльного аналізу.

РОЗДІЛ 1. ПОНЯТТЯ СЕМАНТИЧНОЇ ОБРОБКИ ЗОБРАЖЕННЯ

1.1. Сприйняття зображення комп'ютером

Більшість інформації, що нас оточує, ми сприймаємо завдяки своїм очам. І найшвидше ми сприймаємо інформацію у вигляді зображень - нейробіологи з Массачусетського технологічного інституту виявили, що мозок може розпізнавати зображення всього за 13 мілісекунд [2]. Отже, в сучасному насиченому інформацією світі зображення є одним із найзручніших способів обміну інформацією. Ми знаємо, що інформація накопичується досить швидко і з кожною секундою її стає більше [3]. Ми використовуємо комп'ютери та їхні можливості для пошуку, фільтрації та сортування необхідної нам інформації. Комп'ютери все ще не можуть сприймати зображення так швидко й так само, як людський мозок, і через це комп'ютерам важче фільтрувати та сортувати зображення. Оскільки зображення становлять значну частку всієї інформації в Інтернеті, питання комп'ютерного розуміння зображень стає все більш актуальним з кожним днем. Розглянувши, як саме людина розуміє зображення і як це робить комп'ютер, ви можете зрозуміти, в якому напрямку ми повинні рухатися, щоб вирішити цю проблему. Люди та комп'ютери все ще сприймають зображення по-різному. Коли людина дивиться на зображення, він інтерпретує його на основі свого досвіду — він розуміє зображене завдяки раніше накопиченим знанням про різноманітні об'єкти реального світу та їх співвідношення. Крім того, люди здатні зрозуміти зображений об'єкт, навіть якщо він розмитий або не повністю видимий, тому що ми здатні думати про необхідні елементи за допомогою уяви. Людина може легко визначити, що саме зображено, що відбувається на зображенні, чи пов'язані елементи на ньому між собою і як саме, і в результаті, на основі власного досвіду, дати якусь інтерпретацію. до всього цього

Ми здатні так швидко інтерпретувати що знаходиться на зображенні в основному завдяки своєму досвіду – наш мозок за всі роки нашого життя

накопичує багато прикладів як виглядає той або інший об'єкт, які у об'єктів можуть бути взаємозв'язки та як вони зазвичай виглядають, тощо. У той же час, комп'ютерна інтерпретація зображень набагато примітивніша. Для комп'ютера зображення – це набір чисел, кожне з яких є представленням пікселя зображення. І це за умови що зображення чорно-біле, а якщо воно кольорове то його представлення набуває вигляду трьох наборів чисел, кожен з яких відповідає за один з каналів RGB. Таке представлення кольорового зображення тільки ускладнює його сприйняття комп'ютером – навіть задача пошуку простих геометричних фігур стає нетривіальною, вже не кажучи про довільні об'єкти.

Отже, у зв'язку зі стрімким збільшенням обсягів даних у світі, в тому числі зображень, а також через необхідність їх аналізу та фільтрації, зрештою виникла окрема галузь інформатики – комп'ютерний зір. Ця галузь поставила собі за мету автоматизувати обробку візуальної інформації та наблизитися до того, як це робить людський мозок. В даний час методи комп'ютерного зору досягли значних успіхів у вирішенні деяких завдань обробки зображень – класифікації, визначення меж об'єктів, сегментації та ін. Це, звичайно, значне покращення порівняно з відносно простим сприйняттям комп'ютером зображення як масиву пікселів, але це не наближає нас до рівня людського розуміння. Але з іншого боку, ці розробки є важливою і потужною базою для подальших досліджень і вдосконалення результатів.

Кольорове зображення представляється як група одноколірних зображень. В колірному форматі RGB представляються фотографії які складається з 3 матриць: Red, Green, Blue. Для створення різних кольорів всі три матриці поєднуються. Фотографії шахт безумовно, монотонні. Всі три матриці мають однакове розподіл кольору пікселів і забезпечують аналогічну інформацію, і однієї матриці досить, щоб уявити властивості на зображенні. Кожне однокольорове цифрове зображення представлене у вигляді матриці яскравості. На зображенні кожен піксель має унікальну

адресу (x, y) в матриці, і йому присвоюється значення, відповідне яскравості пікселя. Яскравість пікселя зображення $N \times N$ можна представити в матричній формі, як зображено на Рис.1

$f(1,1)$	$f(1,2)$	$f(1,3)$...	$f(1,n-1)$
$f(2,1)$	$f(2,2)$	$f(2,3)$
$f(3,1)$	$f(3,2)$	$f(3,3)$
...
$f(n-1,1)$	$f(n-1,n-1)$

Рис.1.1 Приклад представлення пікселів в матричній формі.

Процес обробки зображень на початковій стадії ділиться на 3 етапи:

- Формування зображення. Етап формування зображення включає в себе завантаження зображення в програму і його попередню обробку.
- Введення зображень шару здійснюється за допомогою оптичної камери. При роботі системи формується безперервна послідовність зображень з частотою 60 кадрів/с. При цьому оптична вісь камери, яка формує зображення, розташовується перпендикулярно смузі конвеєра з матеріалом.
- Підготовчий етап. На цьому етапі зображення перетворюється в форму, зручну для подальшого аналізу, і готується до цього аналізу. Параметричний аналіз. На цьому етапі аналізу розраховуються основні технологічні характеристики матеріалу.

Одним із основних елементів обробки зображення на підготовчому етапі є процес виділення його складових ділянок (сегментацій), оскільки на цьому етапі виділяються об'єкти для подальшого розпізнавання та аналізу [3]. Сегментація є сполучною ланкою між попередньою обробкою зображення та його аналізом [4].

1.2.Поняття обробки зображення

Цифрова обробка зображень — це використання комп'ютером алгоритмів обробки цифрових зображень [5, 6]. Сегментація зображення є важливим і складним процесом обробки зображення.

У цифровій обробці зображень і комп'ютерному зорі сегментація зображення — це процес поділу цифрового зображення на кілька сегментів зображення, також відомих як області зображення або об'єкти зображення (набори пікселів). Мета сегментації полягає в тому, щоб спростити та/або змінити подання зображення на щось, що є більш значущим і легшим для аналізу.[7][8] Сегментація зображення зазвичай використовується для визначення місцезнаходження об'єктів і меж (ліній, кривих тощо) на зображеннях. Точніше, сегментація зображення — це процес призначення мітки кожному пікселю зображення таким чином, щоб пікселі з однаковою міткою мали певні характеристики.

Результатом сегментації зображення є набір сегментів, які разом покривають усе зображення, або набір контурів, виділених із зображення (див. визначення країв). Кожен із пікселів у регіоні схожий щодо певної характеристики або обчисленої властивості[9], такої як колір, інтенсивність або текстура. Суміжні регіони мають суттєво різний колір щодо тієї самої характеристики(й).[7] При застосуванні до стека зображень, типових для медичної візуалізації, отримані контури після сегментації зображення можна використовувати для створення 3D-реконструкцій за допомогою алгоритмів інтерполяції, як-от крокуючих кубів.[10]

Деякі з практичних застосувань сегментації зображення:

- Пошук зображень на основі вмісту
- Машинний зір
- Медична візуалізація, у тому числі об'ємні зображення з комп'ютерної та магнітно-резонансної томографії.
- Виявляти пухлини та інші патології

- Вимірювання об'ємутканини
- Діагностика, дослідження анатомічної будови
- Планування хірургічного втручання
- Симуляція віртуальної хірургії
- Внутрішньо операційна навігація
- Виявлення об'єктів
- Виявлення пішоходів
- Розпізнавання обличчя
- Виявлення стоп-сигналу
- Знаходження об'єктів на супутникових знімках (дороги, ліси, посіви тощо)
- Завдання на розпізнавання
- Розпізнавання обличчя
- Розпізнавання відбитків пальців
- Розпізнавання райдужки
- Системи керування дорожнім рухом
- Відеоспостереження
- К-сегментація відеооб'єктів і локалізація дії

Сегментація зображення є необхідним першим кроком у аналізі зображення. Метою сегментації зображення є розділення зображення на кілька частин/сегментів, які мають подібні характеристики (атрибути).

Основними застосуваннями сегментації зображень є:

- пошук вмісту зображень;
- медична візуалізація;
- завдання на виявлення та розпізнавання предметів;
- системи автоматичного контролю дорожнього руху та відеоспостереження, тощо.

Сегментацію зображення можна класифікувати за двома основними типами: локальну сегментацію (пов'язану з певною частиною або областю зображення) та глобальну сегментацію (пов'язану із сегментацією всього зображення, що складається з великих кількості пікселів). Підходи до сегментації зображення можна розділити на два типи на основі властивостей зображення.

Виявлення несучільності (розриву) — це підхід, за якого зображення сегментується на області, засновані на розривах. Сегментація, заснована на виявленні меж, належить до категорії, в якій межі, утворені внаслідок розриву в інтенсивності, виявляються та з'єднуються для формування меж регіонів [11]. Це підхід, при якому зображення сегментується на регіони на основі подібності. Методи, які підпадають під цей підхід, це порогові методи, методи збільшення регіонів, а також поділ і злиття регіонів. Усі вони поділяють зображення на області з однаковим набором пікселів. Методи кластеризації також використовують цю методологію. Вони ділять зображення на набір кластерів, які мають схожі характеристики на основі деяких заздалегідь визначених критеріїв.

Тобто до сегментації зображення можна підійти з трьох точок зору: регіональний підхід, граничний підхід і кластеризація даних. Регіональний підхід підпадає під виявлення подібності, виявлення меж — під виявлення розриву, а методи кластеризації також підпадають під виявлення подібності.

У минулому сегментація зображень була неефективною у великих масштабах, але завдяки графічним процесорам, хмарним GPU і периферійним обчисленням її додатки тепер стали більш доступними для громадськості.

1.3. Сегментація зображення

Для сегментації зображення розроблено кілька алгоритмів і методів загального призначення. Щоб бути корисними, ці методи зазвичай повинні

поєднуватися зі специфічними знаннями домену, щоб ефективно вирішити проблеми сегментації домену.

Методи сегментації зображень активно розвивалися разом із швидким прогресом штучного інтелекту протягом останніх років. У результаті тепер є три різні підходи до сегментації зображення (Рис.1.2), кожен зі своїми плюсами та мінусами.

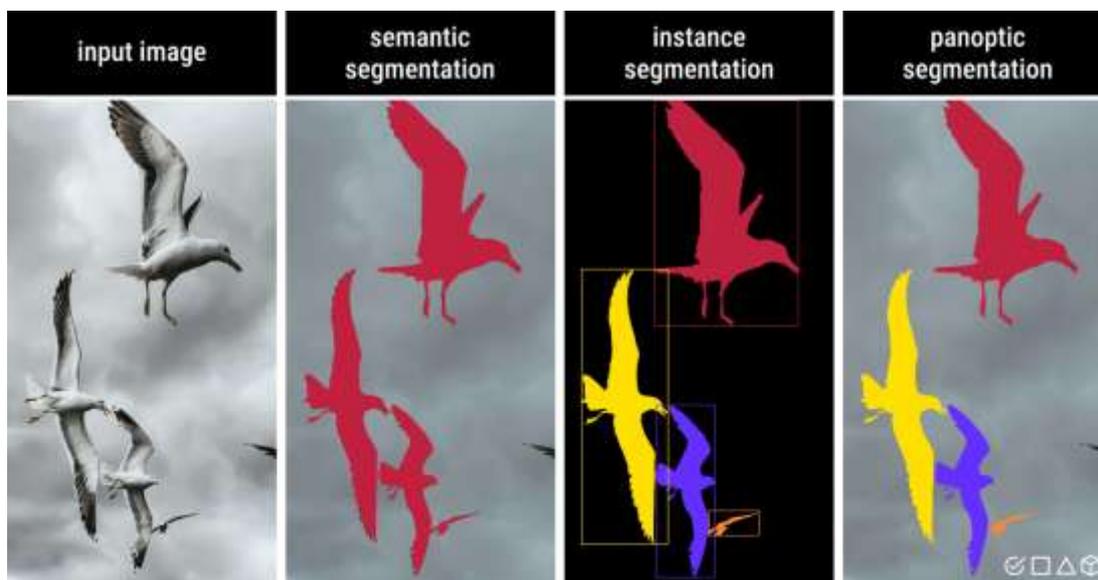


Рис.1.2 Підходи до сегментації зображення

Семантична сегментація

Семантична сегментація є одним із найскладніших, але важливих завдань маркування даних у машинному навчанні, особливо в області комп'ютерного зору. По суті, це той самий старий процес навчання машин розпізнаванню різних об'єктів і сцен у семантичних зображеннях або відео, що є природною здатністю для людей.

Стандартна семантична сегментація, також повна піксельна семантична сегментація, має на меті призначити відповідну унікальну мітку класу кожному пікселю на зображенні, вказуючи, що цим пікселем представлено. Це завдання також відоме як щільне прогнозування, оскільки ми прогнозуємо для кожного пікселя зображення. Результатом цього методу є, по суті, високоякісне зображення з кожним пікселем, призначеним до певного класу,

часто такого ж розміру, як вихідне зображення. Отже, це класифікація зображень на рівні пікселів.

Семантична сегментація складається з трьох кроків:

- Класифікація: класифікація певного об'єкта на зображенні.
- Локалізація: пошук об'єкта та малювання рамки навколо нього.
- Сегментація: групування пікселів у локалізованому зображенні шляхом створення маски сегментації.

По суті, завдання семантичної сегментації можна назвати класифікацією певного класу зображення та відокремленням його від решти класів зображення шляхом накладання на нього маски сегментації.

Це також можна розглядати як класифікацію зображень на піксельному рівні.

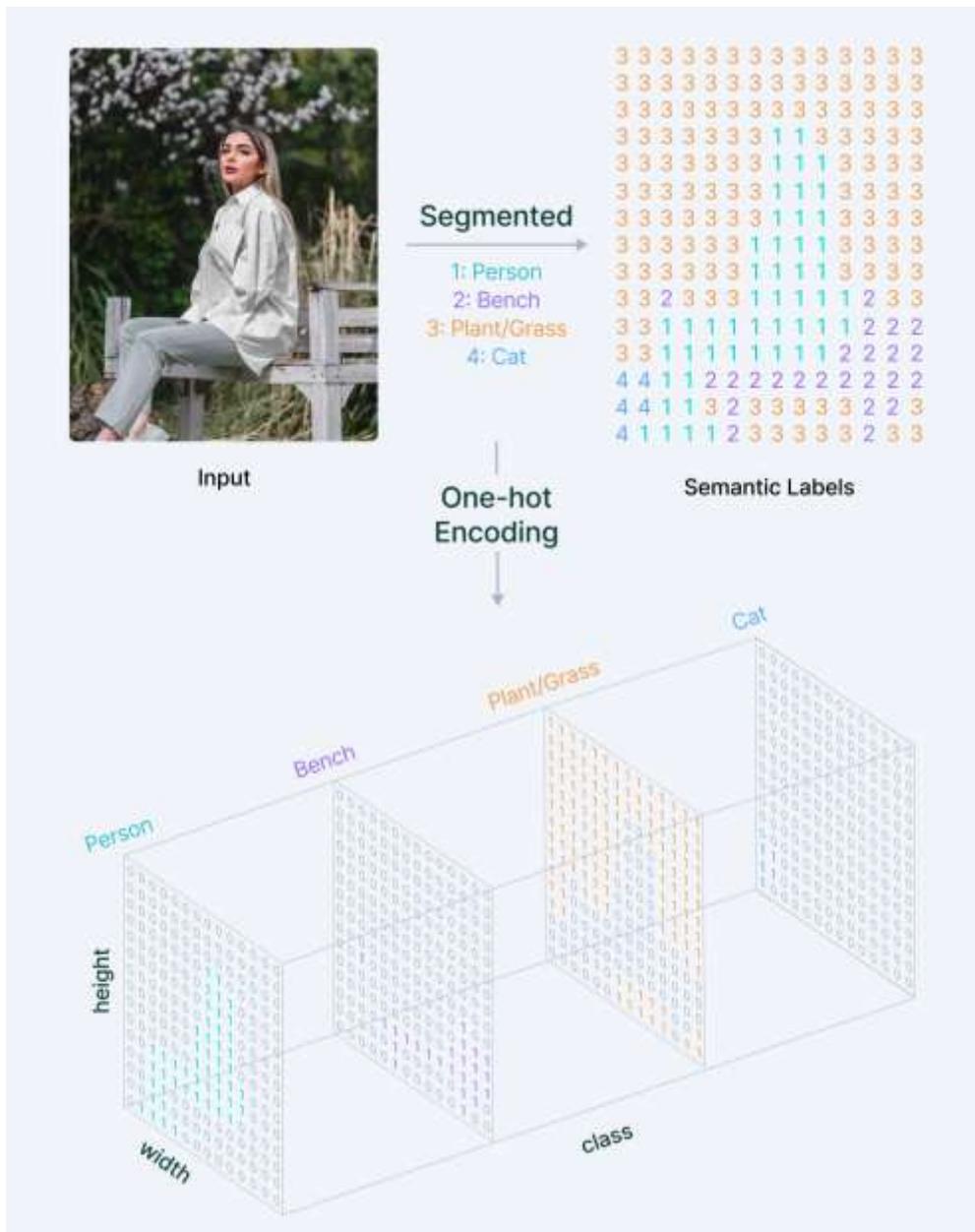


Рис.1.3 Приклад роботи семантичної сегментації

Примітно, що семантична сегментація шукає речі, які незліченні на зображенні. Кожен піксель зображення перевіряється, і залежно від текстури або категорії, яку він представляє, надається окрема мітка класу. Методи семантичної сегментації, які часто використовуються, включають SegNet, U-Net, DeconvNet і FCN.

Сегментація екземпляра

Завдання, що включають обчислювані елементи, зазвичай є центром сегментації екземплярів. Цей метод має можливість ідентифікувати кожен

елемент або екземпляр класу, видимий на зображенні, і надає йому чітку маску або обмежувальну рамку зі спеціальною ідентифікацією.

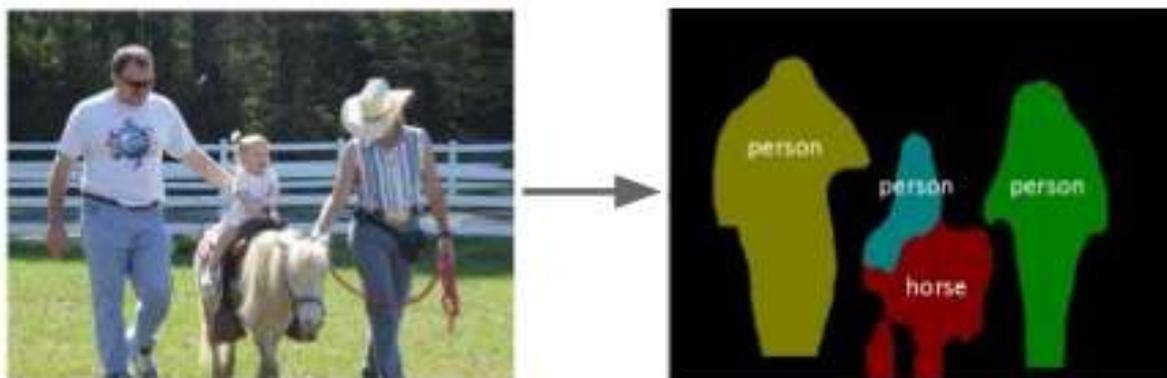


Рис.1.4 Приклад сегментації екземплярів

Припустімо, у вас є фотографія дороги з транспортними засобами та пішоходами. У семантичній сегментації всі пішоходи матимуть одну текстуру (і одну мітку класу), а всі транспортні засоби матимуть іншу мітку (наприклад, автомобіль). Однак у випадку сегментації кожен транспортний засіб буде позначено окремо (наприклад, car_1, car_2), а також пішоходи.

Але ці два методи мають одну спільну рису, а саме когерентну обробку сцени.

Паноптична сегментація

Завжди є щасливе середовище, навіть у методах сегментації зображень. У цьому випадку ми говоримо про панорамну сегментацію. Він пропонує уніфікований метод, у якому кожному пікселю в сцені надається семантична мітка (семантична сегментація) та спеціальна ідентифікація екземпляра (сегментація екземпляра).



Рис.1.5 Приклад паноптичної сегментації

Лише одна пара семантичних міток і ідентифікація екземпляра надається кожному пікселю в панорамній сегментації. Однак пікселі можуть накладатися. Щоб вирішити цей конфлікт, метод панорамної сегментації надає перевагу екземпляру об'єкта.

1.4.Математична постановка задачі

У [11-14] пропонується математична постановка задачі сегментації:

Нехай задано $P = \{p_1, p_2, \dots, p_n\}$ – кінцева множина елементів площини зображення (моделі), іконічні ознаки яких відображені функцією $f(a)$, а також предикат LP , що визначає правило сегментації $f(a)$.

Для детермінованої моделі розв'язанням задачі сегментації $f(a)$ є розбиття P на $S = \{s_1, s_2, \dots, s_k\}$ непустих зв'язкових областей, для яких LP набуває справжніх значення тоді і тільки тоді, коли будь-яка пара елементів з будь-якого підмножини $s_i \in S$ задовольняє цілі. В альтернативних умовах зазвичай метою є оцінка динаміки зміни густини розподілу.

У випадку сегментацію можна як

$$\text{Seg: } f(a) \xrightarrow{LP} S^{(\text{Sem.})},$$

де $s^{(\text{Sem.})}$ - Семантичні відносини.

Зокрема, $s_i(\text{Sem.})$ – ім'я області $s_i \in S$, а $LP(s_i^{\text{Sem.}})$ – вказівка на модель сусідства, що характеризує об'єкт.

Задовільний результат сегментації при заданій множині S є ідеальним випадком і підходить для «класичного» синтаксису детермінованого відображення:

1) модель дозволяє апіорі встановити значення іконічної функції $f(a)$ для області «фон». Область «фон» заповнює все зображення, в якому розташовані області S , які мають властивості об'єкта. Область «фон» характеризується найбільш контрастним значенням яскравості щодо сегментованих областей, зазвичай близьким до максимального або мінімального значення;

2) сегментовані області s_i мають відмінності за унарними ознаками (зазвичай за градієнтом: $\nabla f(a)$) та сусідством, те саме справедливо щодо області «фон». Тобто. області мають кордон у вигляді замкнутого контуру.

У детермінованих умовах робота правила LP є комплексною оцінкою «відстань» (близькості) $\nabla f(a)$ поточного $p_i \in P$ та сусідства (суміжності) деякого $p_j \in s_i$ виду

$$|p_i: \nabla f(a) - s_i| \rightarrow \min., \text{ и } p_i \cap p_j = 1, \text{ где } p_j \in s_i. \quad (1)$$

Так, кожен поточний $p \in P$ відноситься лише до однієї області $s_i \in S$. Відповідно, сегментація виконується на основі однорідності та зв'язності p_i .

Поза ідеальними умовами відображення властивостей об'єкта виникають закономірні порушення форми сегментованих областей за рахунок спотворень у вигляді незакінчених, хибних, потовщених контурів, «накладання» областей та інших відомих помилок [11, 15, 16].

Один із способів боротьби з помилками – модернізація правила (1) семантичної складової як «радіуса», що налаштовується, що дозволяє поліпшити якість сегментації. Таке правило має такий вигляд:

$$|p_i: \nabla f(a) - LP(si^{Sem})| < r_i^{Sem} \text{ и } p_i \cap p_j = 1, \text{ где } p_j \in s_i, \quad (2)$$

де r_i^{Sem} – радіус однорідності для si^{Sem} .

Так, оцінка однорідності класу може бути змінена в процесі сегментації [16, 18] шляхом зміни r_i^{Sem} щодо аналізу однорідності суміжних класів. Крім цього, може бути виконане коригування потужності S на основі аналізу однорідності s_i^{Sem} та потужності області «фон».

Використання принципу адаптації (2), з одного боку, покращує якість, а з іншого – закономірно обмежує можливості застосування правила $LP(si^{Sem})$ для роботи систем аналізу зображень з іншими властивостями. Причому $LP(si^{Sem})$ зазвичай є апіорними знаннями [19-22], заснованими на іконічній інформації, що обмежує практичне використання формалізації [11-14] у вирішенні задачі сегментації.

1.5.Завдання сегментації

Метод сегментації змінює або спрощує представлення зображення, щоб полегшити його подальший аналіз шляхом розкладання зображення на частини.

Як згадувалося вище, сегментація зображення часто є першим кроком у обробці цифрового зображення. Якщо розглядати задачу класифікації та розпізнавання образів, то формування ознак об'єкта дуже сильно залежить від результатів сегментації, яка за своїм напрямом впливає на якість кінцевого результату. Це пов'язано з тим, що на цьому етапі зображення може бути сильно спотворено, тому використовуються методи сегментації, які відповідають поставленому завданню.

Завдання сегментації можна поставити різними способами. З одного боку, це поділ всього зображення на схожі за деякими характеристиками сегменти. У цьому випадку метод обробки може відрізнитися для кожного сегмента під час подальшої обробки. У такій установці сегментація

використовується як перший крок у вирішенні проблеми стиснення зображення. З іншого боку, постановку завдання сегментації можна розглядати як виділення деяких сегментів, що цікавлять у вихідному зображенні, з метою подальшої детальної роботи з ними. З цим параметром сегментація використовується для вирішення завдань аналізу сцени або пошуку заданих об'єктів.

1.6.Проблематика

Дослідники часто стикаються з реальними проблемами при розробці методів сегментації зображень. Якщо одна частина проблеми пов'язана зі специфікою зберігання та обробки цифрових зображень, то інша частина проблеми виникає через певну нечіткість самої задачі сегментації.

При обробці зображень однією з труднощів є розробка алгоритму вирішення певної задачі. Як приклад можна навести класичну проблему класифікації зображень, а саме пошук певних класів із заздалегідь визначеного набору зображень. Простіше це можна було б описати як пошук відповіді на питання «що зображено на картинці?», при цьому варіанти відповідей відомі. Найпростішим випадком цього завдання є бінарна класифікація – коли вам просто потрібно вказати, чи належить зображення до певного класу чи ні. Якщо, наприклад, у нас є клас «яблуко», то виникає питання: як визначити, що воно є на зображенні? Який алгоритм ми для цього використаємо?

Людина скаже, що відповідь цілком очевидна: яблуко - це предмет кулястої форми, воно може бути зеленим, червоним або жовтим, воно росте на дереві. Кожен у своєму житті бачив яблуко і точно розуміє, як воно виглядає. Однак нам не обов'язково знати, як виглядає будь-який сорт яблук на цій планеті, щоб відповісти на питання «як виглядає яблуко?». Для людини алгоритм визначення, чи це яблуко, досить простий і базується на вже наявних знаннях про яблука та їх звичний вигляд.

Можна спробувати побудувати подібний алгоритм для комп'ютера - він перевірить зображення на наявність характерних ознак яблук, щоб визначити, чи присутні вони на зображенні. Але відразу виникає декілька запитань: скільки має бути цих ознак, які з них важливіші, що робити, якщо виявлена лише частина ознак, а не всі? Є багато інших фруктів і овочів, схожих на яблука, тому потрібно розуміти, як відрізнити, помідор це чи яблуко, чи взагалі апельсин. Крім того, зображення навряд чи буде містити тільки одне яблуко на білому фоні - яблук може бути кілька, вони можуть бути тільки в кутку, саме зображення може бути сильно деформованим - і це тільки ускладнює наше завдання.

Стає очевидним одне: створити такий алгоритм або неможливо, або дуже важко. Крім того, що нам робити, якщо ми потім захочемо навчити комп'ютер ще й класифікувати помідори? Для кожного нового класу потрібно буде створювати новий алгоритм, вибирати нові ознаки і т. д. Враховуючи все це, можна зробити висновок, що підхід з явним визначенням алгоритму не найкращий і потрібно шукати інший підхід до цієї проблеми.

Таким підходом є машинне навчання. Замість того, щоб явно вказувати алгоритм, машинне навчання створює власний на основі великих обсягів вхідних даних. Цей підхід імітує те, як люди набувають досвіду та вчаться вирішувати різні проблеми. Цей метод дає нам можливість не явно вказувати алгоритм, а просто надати багато прикладів як вхідні дані, щоб машина сама розробила алгоритм для розрізнення цих прикладів (якщо ми говоримо про класифікацію).

Оскільки машинне навчання добре підходить для вирішення проблем із зображеннями, його почали активно використовувати в обробці зображень. Завдяки їй було успішно вирішено багато нових завдань, які раніше не вдавалося вирішити. Але машинне навчання має свої складності – такий підхід вимагає великої кількості відповідних вхідних даних. Нові дані,

включаючи зображення, як згадувалося раніше, з'являються у великих кількостях щодня. З одного боку, це спрощує завдання - не виникне проблем з пошуком потрібних прикладів для конкретного завдання. З іншого боку, їх кількість ускладнює цей пошук і завдання створення повного набору ознак для вхідних даних.

Але, незважаючи на труднощі, машинне навчання наразі є найкращим підходом, який допоможе нам вирішити проблему комп'ютерного розуміння зображень і наблизити вирішення цієї проблеми до людського рівня.

При створенні цифрового зображення фактично з'являється цифровий шум. Цифровий шум — це дефект зображення, який виникає внаслідок недосконалості фотодатчика та електроніки камери та природи світла. Шум з'являється на зображенні як накладена маска з пікселів випадкового кольору та яскравості. Наявність шуму може сильно послабити результат подальшої обробки зображення.

Якщо зображення було стиснуте з втратами під час передачі, воно покаже так звані артефакти стиснення. Через те, що для достатнього стиснення зображення потрібно значно спростити, деякі дані відкидаються. Таким чином, зображення демонструє блокові спотворення, кільцеві артефакти та змазування. Якість зображення стає жахливою, а результати постобробки не будуть задовільними.

Також варто відзначити, що освітлення також відіграє важливу роль в обробці зображень. Яскраві відблиски світла або, навпаки, тіні на предметах можуть призвести до неправильного визначення відрізків.

1.7.Застосування

Автономна промисловість

Моделі семантичної сегментації забезпечують дуже корисні навички розуміння сцени для різних автономних платформ (наприклад, безпілотних

транспортних засобів, дронів, роботів тощо). Наприклад, семантична сегментація може ідентифікувати смуги руху та дорожні знаки, а також надавати інформацію про відкриті ділянки на магістралях.



Рис.1.6 Застосування в автономній промисловості

Мережа повинна забезпечувати плавну сегментацію, оскільки в типових дорожніх сценах більшість пікселів займають такі речі, як будівлі або дороги. Отже, існує як академічний, так і комерційний інтерес до точних моделей сегментації з невеликим використанням пам'яті та швидким висновком.

Охорона здоров'я

Щоб суттєво скоротити час, необхідний для виконання діагностичних тестів, семантична сегментація також проклала свій шлях у сферу діагностики медичних зображень. Класифікація аномалій під час КТ стає дуже корисним інструментом для радіологів. Складність комп'ютерної томографії та більшості медичних зображень ускладнює виявлення нерівностей. Семантичну сегментацію зображення можна використовувати як діагностичний інструмент для вивчення цих зображень і допомоги

рентгенологам і лікарям у прийнятті життєво важливих рішень щодо догляду за пацієнтом.

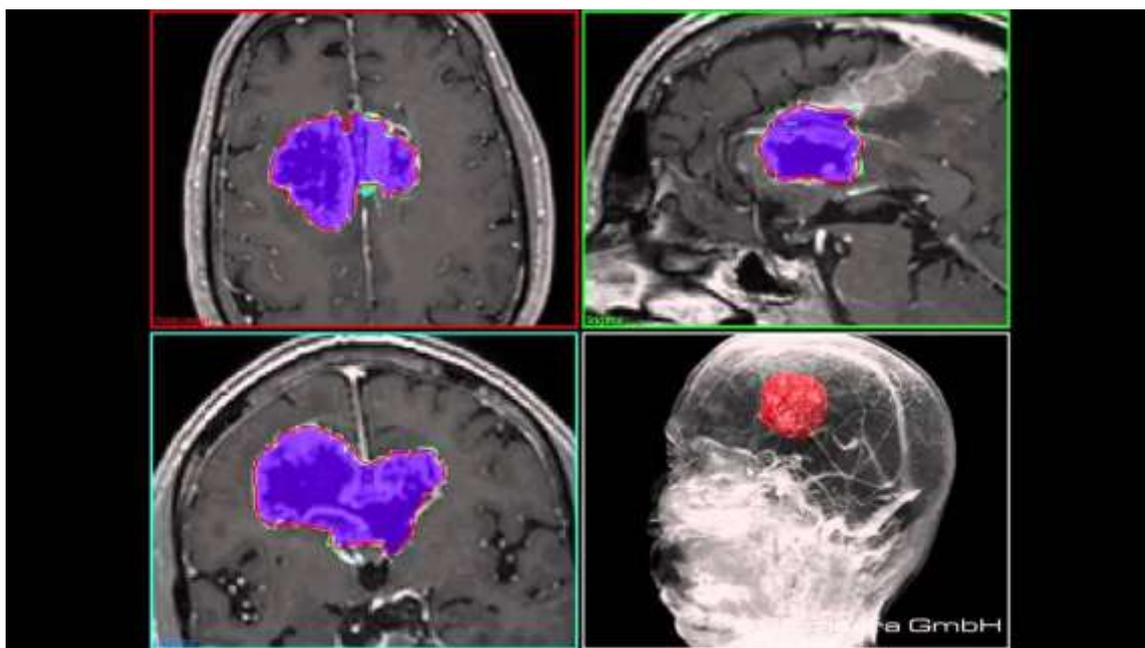


Рис.1.7 Приклад сегментації на знімку

Аерофотознімки

Подібно до процесу розуміння сцени, обробка аерофотознімків також вимагає семантичної сегментації місцевості, як її видно згори. Безпілотники (БПЛА) можуть розкидатися для обстеження різних місць під час надзвичайних ситуацій, таких як повінь або пошук людей і тварин, яких потрібно врятувати. Використання обробки аерофотознімків при доставці товарів є ще одним життєздатним прикладом застосування моделі семантичної сегментації.

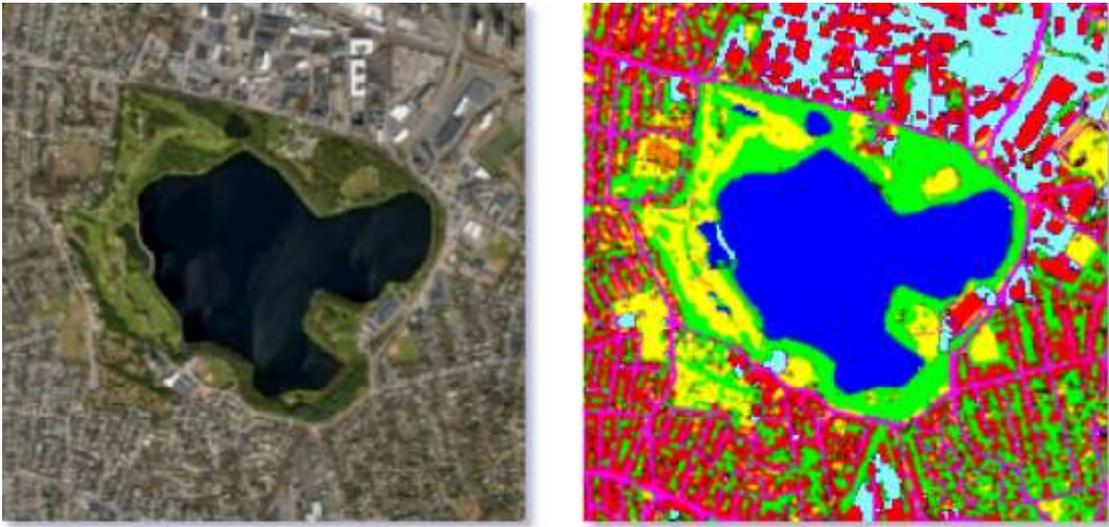


Рис.1.8 Приклад застосування сегментації

Геозондування

Проблеми семантичної сегментації є проблемами класифікації. Таким чином, існує варіант використання для картографування землекористування на супутникових знімках. Інформація про земний покрив має вирішальне значення для кількох цілей, включаючи відстеження урбанізації та вирубки лісів. Класифікацію земельного покриву можна розглядати як багатокласове завдання семантичної сегментації, оскільки воно визначає тип земельного покриву (тобто міські, сільськогосподарські або водні території) для кожного пікселя на супутниковому зображенні. Крім того, для управління дорожнім рухом, міського планування та моніторингу доріг виявлення доріг і будівель є важливою областю дослідження.

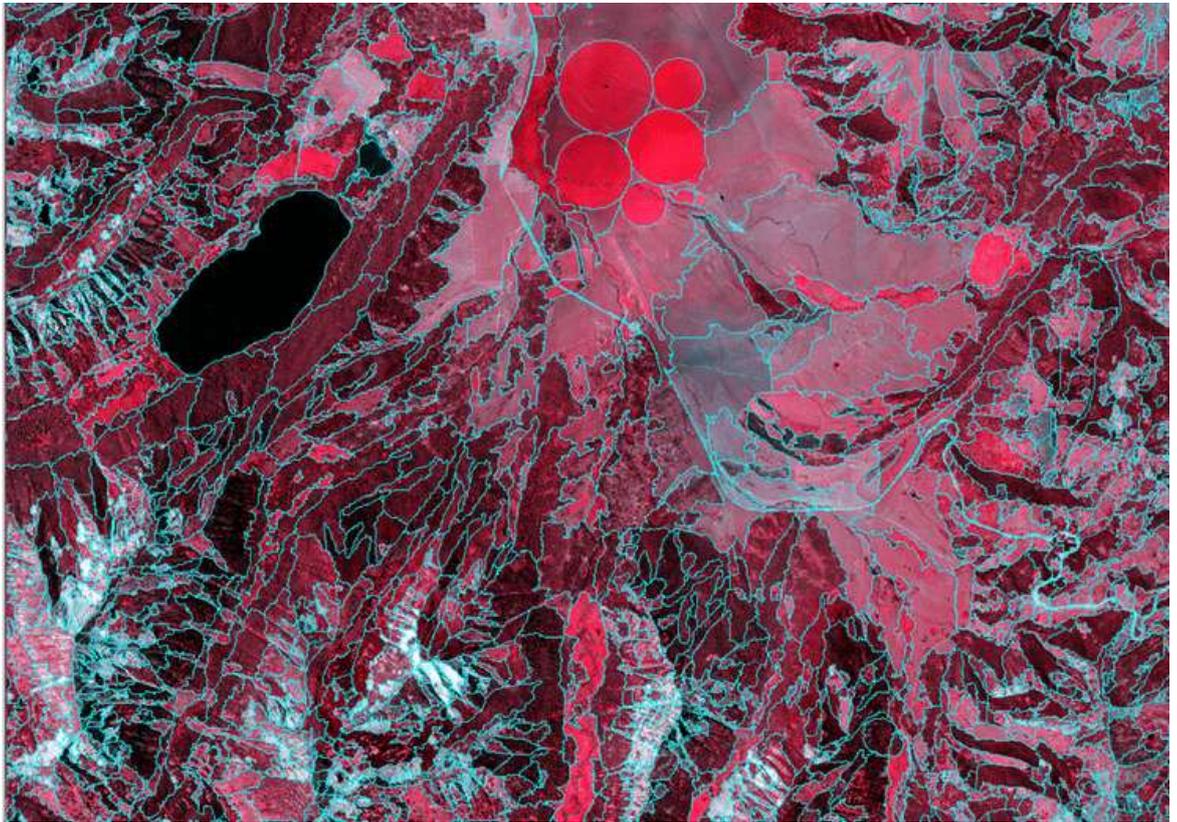


Рис.1.9 Приклад застосування сегментації

Точне землеробство

За допомогою семантичної сегментації сільськогосподарських культур і бур'янів роботи точного землеробства можуть мінімізувати кількість гербіцидів, які необхідно розпорошити на полях, і допомогти їм негайно розпочати операції з видалення бур'янів у реальному часі.

Сільськогосподарський сектор може використовувати ці передові інструменти бачення зображення, щоб автоматизувати більше завдань, які раніше вимагали ручної перевірки.

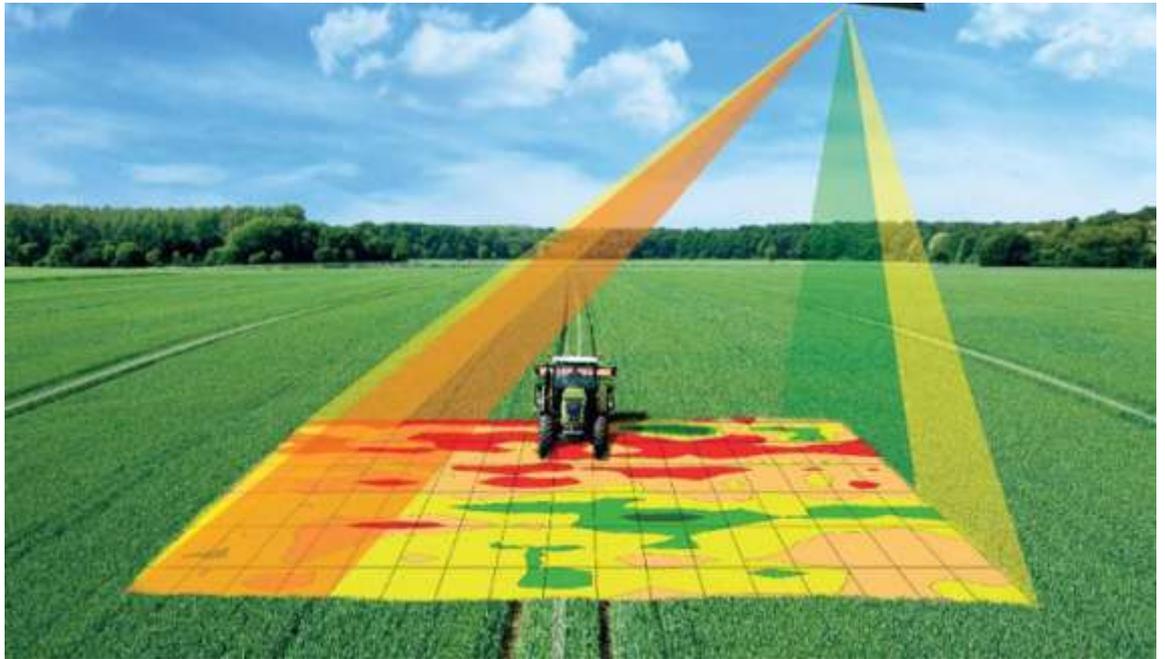


Рис.1.10 Приклад застосування сегментації

РОЗДІЛ 2. ОГЛЯД МЕТОДІВ СЕМАНТИЧНОЇ ОБРОБКИ ЗОБРАЖЕНЬ

Існує два класи методів сегментації: методи на основі класичних підходах комп'ютерного зору та методи на основі глибинного навчання.

Сегментація зображень спочатку почалася з цифрової обробки зображень у поєднанні з алгоритмами оптимізації. Ці примітивні алгоритми використовували такі методи, як вирощування області та алгоритм змій, де вони встановлювали початкові області, а алгоритм порівнював значення пікселів, щоб отримати уявлення про карту сегментів.

Ці методи використовували локальний погляд на особливості зображення та зосереджувалися на локальних відмінностях і градієнтах у пікселях.

Алгоритми, які брали глобальний погляд на вхідне зображення, з'явилися набагато пізніше, коли серед класичних методів обробки зображень були запропоновані такі методи, як адаптивне порогове визначення, алгоритм Оцу та алгоритми кластеризації.



Рис. 2.1 Поділ методів сегментації

2.1. Підготовка даних

Попередня обробка даних дуже важлива при аналізі будь-яких даних. Навіть найкращий алгоритм на невірних, погано підготовлених даних не зможе дати задовільного результату.

Перед початком роботи з цифровими зображеннями їх необхідно належним чином підготувати до подальшої роботи через наявність дефектів. До дефектів цифрового зображення належать цифровий шум, розмитість і дефекти кольору. Для усунення дефектів зображення використовуються різні підходи залежно від цілей дослідження. [23]

Для боротьби з цифровим шумом зазвичай використовуються такі методи, як лінійне усереднення пікселів за сусідами, розмиття за Гаусом і медіанна фільтрація.

Лінійне усереднення пікселів по сусідам є одним з найпростіших методів видалення шуму. Його основна ідея полягає в тому, щоб взяти за нове значення пікселя середнє арифметичне значень пікселів у деяких його околицях усіх тих сусідів, значення яких відрізняється від центрального значення не більше ніж на деякий поріг. Чим більше значення цього порогу, тим сильніше усереднення

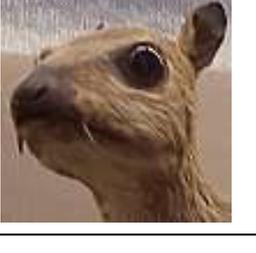
Коли до зображення застосовується розмиття за Гауссом, воно стає менш чітким, як кажуть, розмитим. Фізично така операція реалізується за допомогою операції згортки.

Згортка (ядро, матриця згортки) — матриця, зазвичай, малих розмірів, що використовується в оброблянні зображень як фільтр для розмиття, підвищення різкості, виділення границь тощо. Оброблення зображення полягає в обчисленні нового значення обраного пікселя з врахуванням значення навколишніх пікселів.

В залежності від елементів матриці згортка може викликати різні ефекти.

Це лише декілька прикладів ефектів(Табл.1), досягнутих за допомогою використання згортки.

Тотожне відображення	
Виділення границь	
	
	
Sharpen	

Розмиття квадратом	
Гаусове розмиття	
5×5 Нерівке розмиття	

Таблиця 1 Приклади ефектів згортки

Згортка - це процес додавання кожного елемента зображення до його сусідів, зважених ядром. Важливо зауважити, що виконувана матрична операція - згортка - це не звичайне множення, хоча й позначається *.

Наприклад, якщо ми маємо дві 3x3 матриці, перша - ядро, друга - шматок зображення, згортка - це процес транспонування рядків і стовпчиків ядра з наступним множенням і додаванням. Елемент з координатами [2, 2] (тобто, центральний елемент) отриманого зображення буде зваженою комбінацією всіх елементів матриці зображення, з вагами взятими з ядра:

$$\left(\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \right) [2,2] = (i \cdot 1) + (h \cdot 2) + (g \cdot 3) + (f \cdot 4) + (e \cdot 5) + (d \cdot 6) + (c \cdot 7) + (b \cdot 8) + (a \cdot 9).$$

Значення кожного пікселя у вихідному зображенні рівне сумі добутків значень матриці згортки і відповідних пікселів вхідного зображення.

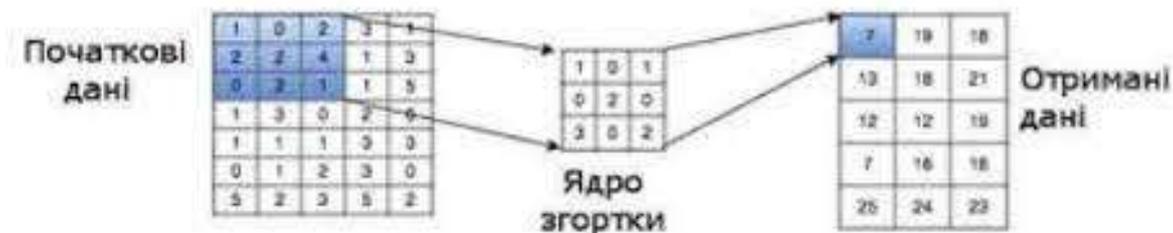


Рис. 2.2 Принцип роботи згортки

Це можна описати таким псевдокодом:

```

for each image row in input image:
  for each pixel in image row:

    set accumulator to zero

    for each kernel row in kernel:
      for each element in kernel row:

        if element position corresponding* to pixel position then
          multiply element value corresponding* to pixel value
          add result to accumulator
        endif

    set output image pixel to accumulator

```

2.2.Пороги

Порогове значення — це один із найпростіших методів сегментації зображення, коли встановлюється поріг для поділу пікселів на два класи. Для пікселів, які мають значення, вищі за порогове значення, встановлено значення 1, тоді як для пікселів зі значеннями, меншими за порогове значення, встановлено значення 0.

Таким чином зображення перетворюється на двійкову карту, що призводить до процесу, який часто називають бінаризацією. Порогове значення зображення дуже корисне, якщо різниця в значеннях пікселів між двома цільовими класами дуже висока, і легко вибрати середнє значення як порогове значення.



Рис. 2.3 Парк, повний снігу

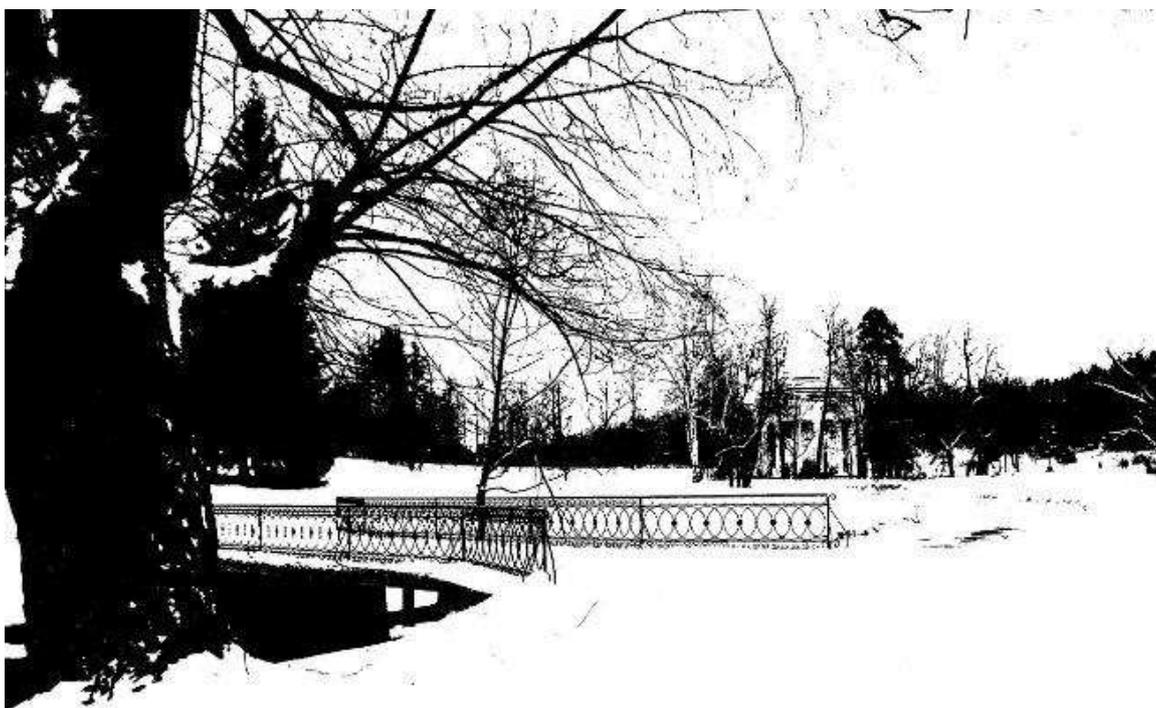


Рис. 2.4 Парк, повний снігу після застосування методу порогу

Порогове значення часто використовується для бінаризації зображення, щоб можна було використовувати додаткові алгоритми, такі як виявлення контурів та ідентифікація, які працюють лише з бінарними зображеннями.

2.3.Алгоритм K-means

Алгоритм K-means — це ітераційна техніка, яка використовується для поділу зображення на K кластерів.[23]

Опис алгоритму

Маємо масив спостережень (об'єктів), кожен з яких має певні значення по ряду ознак. Відповідно до цих значень об'єкт розташовується у багатовимірному просторі.

1. Дослідник визначає кількість кластерів, що необхідно утворити
2. Випадковим чином обирається k спостережень, які на цьому кроці вважаються центрами кластерів
3. Кожне спостереження «приписується» до одного з p кластерів — того, відстань до якого найкоротша
4. Розраховується новий центр кожного кластера як елемент, ознаки якого розраховуються як середнє арифметичне ознак об'єктів, що входять у цей кластер
5. Відбувається така кількість ітерацій (повторюються кроки 3-4), поки кластерні центри стануть стійкими (тобто при кожній ітерації в кожному кластері опиняться одні й ті самі об'єкти), дисперсія всередині кластера буде мінімізована, а між кластерами — максимізована

Вибір кількості кластерів відбувається на основі дослідницької гіпотези. Якщо її немає, то рекомендують створити 2 кластери, далі 3,4,5, порівнюючи отримані результати.

1. k початкових «середніх» (тут $k=3$) випадково згенеровані у межах домени даних (кольорові).
2. створено k кластерів, асоціюючи кожне спостереження з найближчим середнім. Розбиття відбувається згідно з діаграмою Вороного утвореною середніми.
3. Центроїд кожного з k кластерів стає новим середнім.
4. Кроки 2 і 3 повторюються до досягнення збіжності.

Мета методу — розділити n спостережень на k кластерів, так щоб кожне спостереження належало до кластера з найближчим до нього середнім значенням. Метод базується на мінімізації суми квадратів відстаней між кожним спостереженням та центром його кластера, тобто функції

$$\sum_{i=1}^N d(x_i, m_j(x_i))^2 .$$

де d — метрика, x_i — i -ий об'єкт даних, а $m_j(x_i)$ — центр кластера, якому на j -ій ітерації приписаний елемент x_i .

Алгоритм середнього зсуву — це техніка, яка використовується для поділу зображення на невідомий апріорний номер кластерів. Це має перевагу, оскільки не потрібно починати з початкового припущення такого параметра, що робить його кращим загальним рішенням для більш різноманітних випадків.

2.4. Сегментація країв

Сегментація країв, яка також називається виявленням країв, — це завдання виявлення країв на зображеннях.

З точки зору сегментації ми можемо сказати, що виявлення країв відповідає класифікації пікселів у зображенні як краєві пікселі та виділенню цих крайових пікселів відповідно до окремого класу.

Виявлення країв зазвичай виконується за допомогою спеціальних фільтрів, які дають нам краї зображення після згортання. Ці фільтри

обчислюються спеціальними алгоритмами, які працюють над оцінкою градієнтів зображення в координатах x і y у просторової площини.

Нижче наведено приклад виявлення краю за допомогою алгоритму виявлення краю Canny, одного з найпопулярніших алгоритмів виявлення краю.



Рис. 2.5 Сегментація країв

2.5.Методи на основі стиснення

Методи на основі стиснення стверджують, що оптимальна сегментація серед усіх можливих – це та, яка використовує найменшу кількість даних для кодування результуючого зображення.[6][7] Зв'язок між сегментацією та стисненням полягає в тому, що сегментація намагається знайти шаблони в зображенні, і будь-який зв'язок у зображенні можна використовувати для його стиснення. Підхід описує кожен сегмент за його текстурою та формою контуру. Кожен компонент сегмента моделюється за допомогою формули розподілу ймовірностей, а кількість даних для кодування обчислюється таким чином:

Контурне кодування засноване на тому факті, що області на звичайних зображеннях намагаються мати плавний контур. Це припущення використовується в кодуванні Хаффмана для кодування диференціального коду контурного ланцюга на зображенні. Так, чим плавніше контур, тим менше місця він займає.

Текстура кодується за допомогою стиснення з втратами, подібно до принципу мінімально прийнятної довжини опису, але тут обсяг даних для

моделі приблизно оцінюється кількістю вибірок ентропії моделі. Текстура в кожній області моделюється багатовимірним нормальним розподілом, ентропія якого найбільш схожа на представлення. Цікавою особливістю цієї моделі є те, що значення ентропії реальних даних текстури міститься в межах оціненої ентропії. Це тому, що серед усіх доступних функцій розподілу з відомими значеннями середнього арифметичного та моменту кореляції нормальний розподіл має найвищу ентропію. Звідси випливає, що реальний обсяг кодованих даних не може бути більшим за запропонований алгоритмом.

Для довільної сегментації ця схема обчислює кількість бітів, необхідних для кодування зображення за допомогою вибраної сегментації. Отже, серед усіх можливих сегментацій зображення вам потрібно знайти сегментацію, яка представлена кодованими даними найкоротшої довжини. Цього можна досягти за допомогою методів ієрархічної кластеризації. Спотворення при стисненні з втратами визначають похибку сегментації, і її оптимальне значення може відрізнитися для окремих зображень. Цей параметр можна оцінити евристично за показниками контрастності текстур на зображенні.

Метод вододілу

Метод вододілу базується на площах. метод математичної морфології. У географії вододіл — це хребет, який розділяє ділянки різних річкових систем.

Розглядаючи зображення як геологічний ландшафт, можна сказати, що лінії вододілу - це межі, які розділяють ділянки зображення. У топографічному представленні зображення числові значення (наприклад, рівні сірого) кожного пікселя діють як висота цієї точки. Трансформація вододілу обчислює вододіли та хребти, де вододіли є відповідними областями зображення, а хребти є межами цих регіонів. Основною

проблемою цього алгоритму є надмірна сегментація, оскільки всі межі та шуми представлені в градієнті, що робить необхідним процес видалення.

Першим етапом видалення шуму в оригінальному зображенні є застосування морфологічних операцій закриття/розкриття, потім обчислюється морфологічний градієнт безшумного зображення та виконується нелінійне перетворення для рівнів сірого на градієнті зображення за принципом Вебера, Останнім етапом є розрахунок вододілу за нелінійним, розбитим на регіони, градієнтним зображенням [20].

2.6.Нейронна мережа

Нейронна мережа - це мережа або ланцюг біологічних нейронів, або, в сучасному розумінні, штучна нейронна мережа, що складається зі штучних нейронів або вузлів.[23] Таким чином, нейронна мережа — це або біологічна нейронна мережа, що складається з біологічних нейронів, або штучна нейронна мережа, яка використовується для вирішення проблем штучного інтелекту (ШІ). Зв'язки біологічного нейрона моделюються в штучних нейронних мережах як ваги між вузлами. Позитивна вага відображає збудливий зв'язок, тоді як негативні значення означають гальмівні зв'язки. Усі вхідні дані модифікуються за вагою та підсумовуються. Ця діяльність називається лінійною комбінацією. Нарешті, функція активації контролює амплітуду вихідного сигналу. Наприклад, прийнятний діапазон вихідних даних зазвичай становить від 0 до 1, або він може бути від -1 до 1.

Ці штучні мережі можна використовувати для прогнозного моделювання, адаптивного керування та додатків, де їх можна навчати за допомогою набору даних. Самонавчання в результаті досвіду може відбуватися в мережах, які можуть робити висновки зі складного та, здавалося б, непов'язаного набору інформації.[24]

Біологічна нейронна мережа складається з групи хімічно або функціонально пов'язаних нейронів. Один нейрон може бути з'єднаний з багатьма іншими нейронами, а загальна кількість нейронів і зв'язків у мережі

може бути значною. З'єднання, звані синапсами, зазвичай утворюються від аксонів до дендритів, хоча можливі дендродендритичні синапси[24] та інші з'єднання. Крім електричної сигналізації, існують інші форми сигналізації, які виникають внаслідок дифузії нейромедіаторів.

Штучний інтелект, когнітивне моделювання та нейронні мережі — це парадигми обробки інформації, натхненні тим, як біологічні нейронні системи обробляють дані. Штучний інтелект і когнітивне моделювання намагаються імітувати деякі властивості біологічних нейронних мереж. У сфері штучного інтелекту штучні нейронні мережі успішно застосовуються для розпізнавання мовлення, аналізу зображень і адаптивного керування з метою створення програмних агентів (у комп'ютерних і відеоіграх) або автономних роботів.

Історично склалося так, що цифрові комп'ютери еволюціонували з моделі фон Неймана і працюють через виконання явних інструкцій через доступ до пам'яті кількома процесорами. З іншого боку, походження нейронних мереж ґрунтується на спробах моделювати обробку інформації в біологічних системах. На відміну від моделі фон Неймана, обчислення нейронної мережі не розділяють пам'ять і обробку.

Теорія нейронних мереж допомогла краще визначити, як функціонують нейрони в мозку, і стала основою для зусиль зі створення штучного інтелекту.

2.7.CNN(convolutional neural network)

У глибокому навчанні згорткова нейронна мережа (CNN або ConvNet) — це клас штучних нейронних мереж (ANN), які найчастіше застосовуються для аналізу візуальних зображень.[21] CNN також відомі як Shift Invariant або Space Invariant Artificial Neural Networks (SIANN), засновані на архітектурі зі спільною вагою згорткових ядер або фільтрів, які ковзають уздовж вхідних функцій і забезпечують еквіваріантні відповіді на переклад, відомі як карти функцій.[25][26] На противагу інтуїції більшість згорткових нейронних

мереж не є інваріантними до трансляції через операцію зменшення дискретизації, яку вони застосовують до вхідних даних. Вони мають застосування в розпізнаванні зображень і відео, системах рекомендацій, класифікації зображень, сегментації зображень, аналізі медичних зображень, обробці природної мови, інтерфейсах мозок–комп'ютер і фінансових часових рядах.

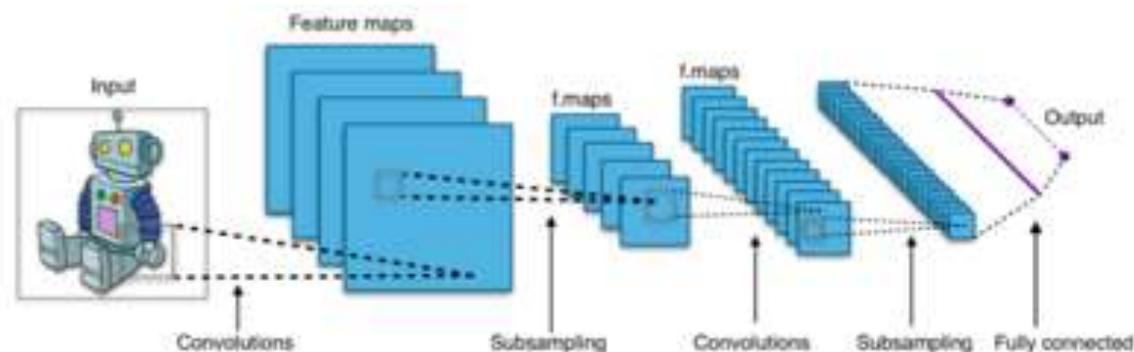


Рис. 2.6 CNN

CNN є регуляризованими версіями багат шарових перцептронів. Багат шарові перцептрони зазвичай означають повністю зв'язані мережі, тобто кожен нейрон одного шару з'єднаний з усіма нейронами наступного шару. «Повна зв'язність» цих мереж робить їх схильними до переобладнання даних. Типові способи регуляризації або запобігання надмірному оснащенню включають: штрафування параметрів під час навчання (наприклад, зниження ваги) або обрізання з'єднань (пропущені з'єднання, випадання тощо). CNN використовують інший підхід до регуляризації: вони використовують переваги ієрархічного шаблону в даних і збирають візерунки все більшої складності, використовуючи менші та простіші візерунки, тиснені у їхніх фільтрах. Таким чином, за шкалою зв'язності та складності CNN знаходяться на нижньому полюсі.

Згорткові мережі були натхненні біологічними процесами у тому, що схема зв'язку між нейронами нагадує організацію зорової кори тварин. Окремі нейрони кори реагують на стимули лише в обмеженій області поля

зору, відомої як рецептивне поле. Рецептивні поля різних нейронів частково перекриваються так, що охоплюють усе поле зору.

CNN використовують відносно невелику попередню обробку порівняно з іншими алгоритмами класифікації зображень. Це означає, що мережа вчиться оптимізувати фільтри (або ядра) за допомогою автоматичного навчання, тоді як у традиційних алгоритмах ці фільтри розробляються вручну. Ця незалежність від попередніх знань і людського втручання у вилучення ознак є головною перевагою.

Конструкція

ЗНМ складається з вхідного та вихідного рівнів, а також кількох прихованих шарів. Приховані шари ЗНМ зазвичай складаються зі згорткових шарів, шарів агрегації, повнозв'язаних шарів і шарів нормалізації.

Цей процес описується в нейронних мережах як згортка за домовленістю. З математичної точки зору, це взаємна кореляція, а не згортка. Це актуально лише для індексів у матриці, а отже, які вагові коефіцієнти розташовані на якому індексі.

Згорткові шари

Згорткові шари застосовують операцію згортки до вхідних даних, передаючи результат до наступного шару. Згортка імітує реакцію одного нейрона на візуальний стимул.

Кожен згортковий нейрон обробляє дані тільки для свого рецептивного поля.

Хоча повністю зв'язані нейронні мережі прямого зв'язку можна використовувати як для вивчення ознак, так і для класифікації даних, застосування цієї архітектури до зображень є недоцільним. Буде потрібна дуже велика кількість нейронів, навіть у поверхневій (на відміну від глибокої) архітектурі, через дуже великі розміри вхідних даних, пов'язаних із зображеннями, де кожен піксель є відповідною змінною. Наприклад, повністю пов'язаний шар для (маленького) зображення розміром 100×100

має 10 000 ваг. Операція згортки вирішує цю проблему, оскільки вона зменшує кількість вільних параметрів, дозволяючи мережі бути глибшою з меншою кількістю параметрів. Наприклад, незалежно від розміру зображення, 5×5 областей мозаїки, кожна з однаковими загальними вагами, вимагають лише 25 вільних параметрів. Таким чином, це вирішує проблему зникнення або розширення градієнтів у навчанні традиційних багатошарових нейронних мереж за допомогою зворотного поширення.

Шари агрегації

Згорткові мережі можуть включати локальні або глобальні рівні агрегації, які поєднують виходи кластерів нейронів на одному шарі в один нейрон на наступному рівні. Наприклад, максимізація агрегації (англ. max pooling) використовує максимальне значення від кожного з кластерів нейронів попереднього шару. Іншим прикладом є об'єднання середніх значень, у якому використовується середнє значення кожного з кластерів нейронів попереднього шару.

Повністю з'єднані шари

Повністю з'єднані шари з'єднують кожен нейрон одного шару з кожним нейроном наступного шару. Це, в принципі, те ж саме, що й традиційна багатошарова нейронна мережа перцептрона.

Ваги

ЗНМ використовують загальні ваги в згорткових шарах, що означає, що той самий фільтр (банк ваг) використовується для кожного рецептивного поля шару; це зменшує обсяг необхідної пам'яті та покращує продуктивність.

2.8. Повністю згорточні мережі.

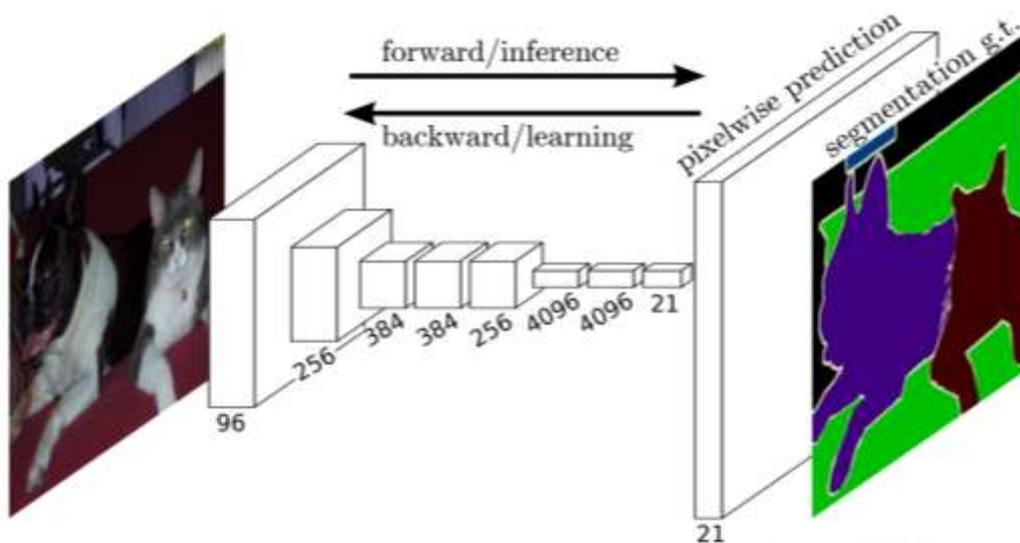


Рис. 2.7 Архітектура FCN

Кожен рівень даних у згортку є тривимірним масивом розміром $h \times w \times d$, де h і w – просторові розміри, а d – розмір функції або каналу.

Перший шар — це зображення з розміром пікселів $h \times w$ і d кольорових каналів. Розташування на вищих шарах відповідають місцям на зображенні, з якими вони пов'язані шляхом, які називаються їхніми рецептивними полями. Згортки побудовані на інваріантності перекладу. Їх основні компоненти (функції згортки, об'єднання та активації) працюють на локальних вхідних областях і залежать лише від відносних просторових координат. Записуючи x_{ij} для вектора даних у місці (i, j) на певному шарі та y_{ij} для наступного шару, ці функції обчислюють виходи

$$y_{ij} = f_{ks} (\{x_{si+\delta i, sj+\delta j}\}_{0 \leq \delta i, \delta j \leq k})$$

де k називається розміром ядра, s є кроком або коефіцієнтом субдискретизації, а f_{ks} визначає тип шару: множення матриці для згортки або середнього об'єднання, просторовий максимум для максимального об'єднання або поелементна нелінійність для активації і так далі для інших типів шарів. Ця функціональна форма зберігається при композиції, при цьому розмір ядра та крок підкоряються правилу перетворення

$$f_{ks} \circ g_{k's'} = (f \circ g)_{k'+(k-1)s', ss'}$$

Хоча згальна глибока мережа обчислює загальну нелінійну функцію, мережа лише з шарами цієї форми обчислює нелінійний фільтр, який ми називаємо глибоким фільтром або повністю згортковою мережею. FCN природним чином працює на вхідних даних будь-якого розміру та створює вихідні дані відповідних (можливо передискретизованих) просторових розмірів. Дійсна функція втрат, складена з FCN, визначає завдання. Якщо функція втрат є сумою просторових розмірів кінцевого шару,

$$\ell(\mathbf{x}; \theta) = \sum_{ij} \ell'(\mathbf{x}_{ij}; \theta),$$

її градієнт буде сумою градієнтів кожного з його просторових компонентів. Таким чином, стохастичний градієнтний спад на l , обчислений для цілих зображень, буде таким самим, як стохастичний градієнтний спад на L , приймаючи всі сприйнятливі поля кінцевого шару як міні-пакет. Коли ці сприйнятливі поля значно перекриваються, обчислення прямого зв'язку та зворотне розповсюдження є набагато ефективнішими, коли обчислюються шар за шаром для всього зображення, а не незалежно один від одного. Далі ми пояснюємо, як перетворити мережі класифікації на повністю згорточні мережі, які створюють грубі вихідні карти. Для попиксельного прогнозування нам потрібно з'єднати ці грубі виходи назад із пікселями.

Перетворення повністю зв'язаних шарів у згортку шари дозволяють мережі класифікації виводити теплову карту. Додавання шарів і просторових втрат (як на малюнку 1) створює ефективну машину для наскрізного щільного навчання.

2.9.U-Net

U-Net — це згорточна нейронна мережа, розроблена для сегментації біомедичних зображень на факультеті комп'ютерних наук Фрайбурзького університету. Мережа заснована на повністю згортковій мережі, а її архітектуру було змінено та розширено для роботи з меншою кількістю

навчальних зображень і для отримання більш точної сегментації. На сучасному графічному процесорі сегментація зображення 512×512 займає менше секунди.

Архітектура U-Net походить від так званої «повністю згорнутої мережі», вперше запропонованої Лонгом, Шелхамером і Дарреллом.

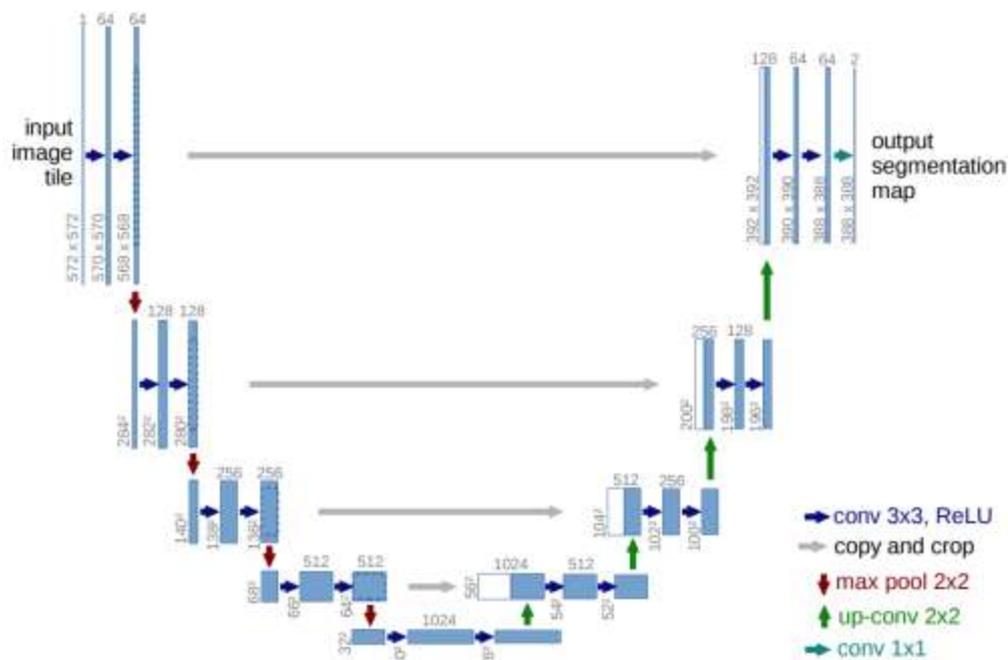


Рис. 2.8 Архітектура U-net

Основна ідея полягає в тому, щоб доповнити звичайну контрактну мережу послідовними шарами, де операції об'єднання замінюються операторами підвищення дискретизації. Отже, ці шари збільшують роздільну здатність виходу. Послідовний згортковий рівень може потім навчитися збирати точний результат на основі цієї інформації.

Однією з важливих модифікацій U-Net є наявність великої кількості функціональних каналів у частині підвищення дискретизації, які дозволяють мережі поширювати контекстну інформацію на рівні вищої роздільної здатності. Як наслідок, шлях розширення є більш-менш симетричним до частини, що стискається, і дає U-подібну архітектуру. Мережа використовує лише дійсну частину кожної згортки без будь-яких повноцінно пов'язаних

шарів. Щоб передбачити кількість пікселів у межі зображення, відсутній контекст екстраполюється шляхом дзеркального відображення вхідного зображення. Ця стратегія мозаїки важлива для застосування мережі до великих зображень, оскільки інакше роздільна здатність буде обмежена пам'яттю GPU.

Мережа складається зі звужувального шляху та розширеного шляху, що надає їй U-подібну архітектуру. Контрактний шлях є типовою згортковою мережею, яка складається з багаторазового застосування згортки, за кожною з яких слідує випрямлена лінійна одиниця (ReLU) і операція максимального об'єднання. Під час скорочення просторова інформація зменшується, а інформація про функції збільшується. Експансивний шлях об'єднує об'єкти та просторову інформацію через послідовність висхідних звивин і конкатенацій із об'єктами високої роздільної здатності зі звужуючого шляху.

2.10. SegNet

SegNet — це модель семантичної сегментації. Ця основна навчена архітектура сегментації складається з мережі кодувальника, відповідної мережі декодера, за якою слідує рівень класифікації по пікселях. Архітектура мережі кодера топологічно ідентична 13 згортковим шарам у мережі VGG16. Роль мережі декодера полягає у відображенні карт функцій кодера з низькою роздільною здатністю на карти функцій із повною роздільною здатністю вхідного сигналу для класифікації по пікселях. Новизна SegNet полягає в тому, як декодер підвищує дискретизацію карт вхідних функцій із нижчою роздільною здатністю. Зокрема, декодер використовує індекси об'єднання, обчислені на кроці максимального об'єднання відповідного кодера для виконання нелінійної підвищення дискретизації.

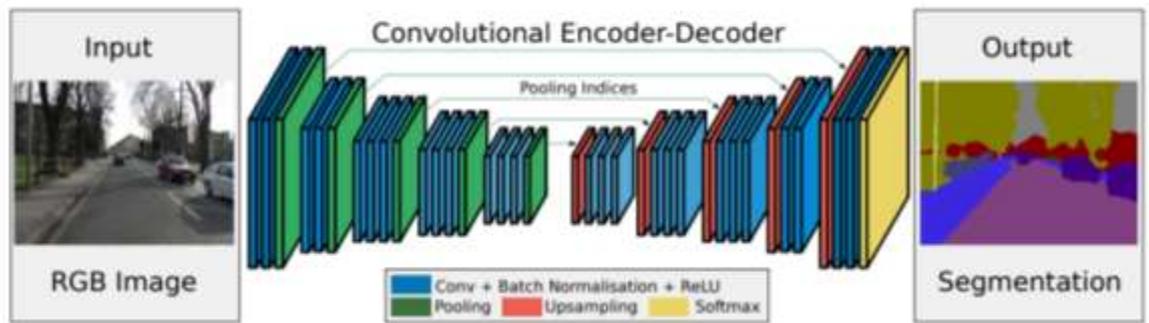


Рис. 2.9 Ілюстрація архітектури SegNet.

SegNet має мережу кодувальника та відповідну мережу декодера, за якими слідує остаточний попиксельний рівень класифікації. Мережа кодера складається з 13 згорткових шарів, які відповідають першим 13 згортковим рівням у мережі VGG16, призначеній для класифікації об'єктів. Тому ми можемо ініціалізувати процес навчання з ваг, навчених для класифікації на великих наборах даних. Ми також можемо відмовитися від повністю пов'язаних шарів на користь збереження карт функцій з вищою роздільною здатністю на найглибшому виході кодера. Це також значно зменшує кількість параметрів у мережі кодувальника SegNet (від 134М до 14,7М) порівняно з іншими останніми архітектурами. Кожен рівень кодера має відповідний рівень декодера, отже мережа декодера має 13 рівнів. Кінцевий вихідний сигнал декодера подається в багатокласовий soft-max класифікатор для створення ймовірностей класу для кожного пікселя незалежно. Кожен кодер у мережі кодувальників виконує згортку з банком фільтрів для створення набору карт функцій. Потім вони нормалізуються партіями). Потім застосовується поелементна випрямлена лінійна нелінійність (ReLU) $\max(0, x)$. Після цього виконується максимальне об'єднання з вікном 2×2 і кроком 2 (вікно, що не перекривається), а отриманий результат субсекретується з коефіцієнтом 2. Максимальне об'єднання використовується для досягнення інваріантності трансляції при невеликих просторових зсувах у вхідному зображенні. Підвибірка призводить до великого контексту вхідного зображення (просторового вікна) для кожного

пікселя на карті функцій. Хоча кілька рівнів максимального об'єднання та підвибірки можуть досягти більшої трансляційної інваріантності для надійної класифікації, відповідно існує втрата просторової роздільної здатності карт функцій. Відображення зображення з дедалі більшою втратою (деталі меж) не є корисним для сегментації, де окреслення меж є життєво важливим.

Таким чином, необхідно захопити та зберегти граничну інформацію в картах функцій кодера перед тим, як буде виконано підвибірку. Якщо пам'ять під час логічного висновку не обмежена, тоді можна зберегти всі карти функцій кодера (після підвибірки). Це зазвичай не так у практичних застосуваннях, тому ми пропонуємо більш ефективний спосіб зберігання цієї інформації. Це передбачає збереження лише максимальних індексів об'єднання, тобто розташування максимального значення функції в кожному вікні об'єднання запам'ятовується для кожної карти функцій кодера.

В принципі, це можна зробити, використовуючи 2 біти для кожного вікна об'єднання 2×2 , і, таким чином, набагато ефективніше зберігати в порівнянні із запам'ятовуванням карти (карт) функцій із точністю до плаваючої величини. Цей менший обсяг пам'яті призводить до невеликої втрати точності, але все ще підходить для практичних застосувань. Відповідний декодер у мережі декодера підвищує дискретизацію своєї вхідної карти (карт) ознак, використовуючи запам'ятовані індекси максимального об'єднання з відповідної карти (карт) функцій кодера.

Цей крок створює розріджену карту(и) об'єктів. Ці карти функцій потім згортаються за допомогою банку фільтрів декодера, який можна навчити, для отримання щільних карт функцій. Потім до кожної з цих карт застосовується етап пакетної нормалізації. Зауважте, що декодер, який відповідає першому кодеру (найближчому до вхідного зображення), виробляє багатоканальну карту функцій, хоча його вхід кодера має 3 канали (RGB). Це не схоже на інші декодери в мережі, які виробляють карти функцій з такою ж кількістю розмірів і каналів, що й їхні входи кодера. Представлення ознак великої

розмірності на виході кінцевого декодера подається в класифікатор soft-max, який можна навчити.

Цей soft-max класифікує кожен піксель окремо. Результатом класифікатора soft-max є зображення K каналу ймовірностей, де K — кількість класів. Прогнозована сегментація відповідає класу з максимальною ймовірністю для кожного пікселя. Інші архітектури такі як U-Net, мають схожу архітектуру з SegNet, але з деякими відмінностями. На відміну від SegNet, U-Net [24] (запропонований для спільноти медичних зображень) не повторно використовує індекси об'єднання, а натомість передає всю карту функцій (ціною більшого обсягу пам'яті) до відповідних декодерів і об'єднує їх у збільшену дискретизацію. (через деконволюцію) карти функцій декодера. В U-Net немає блоку conv5 і max-pool5, як в мережевій архітектурі VGG. SegNet, з іншого боку, використовує всі попередньо підготовлені вагові коефіцієнти згорткового шару з мережі VGG як попередньо підготовлені вагові коефіцієнти.

РОЗДІЛ 3. ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДІВ

Навчання виконувалося з використанням GoogleTensorFlow/Keras на платформі GoogleColab, що надає безкоштовно обчислювальні потужності. Оцінка точності проводилася на тестовій вибірці всім представлених класів. Крім цього було проведено оцінку швидкості збіжності нейронної мережі та точність роботи у процесі збіжності на навчальній вибірці [25]. Системний аналіз, управління та обробка інформації. Точність навчання мереж Під час підготовки остаточних мереж використовувалося 400 епох, оскільки під час роботи це число епох було визначено як оптимальне.

3.1. Обґрунтування вибору ПЗ

За останні роки архітектура глибоких мереж досягла значного прогресу, оскільки Keras і TensorFlow домінують з різними попередньо підготовленими моделями, які вже включені в бібліотеки, серед яких VGG16, VGG1, ResNet50, Inception V3, Xception, MobileNet. Мережі VGG і AlexNet 2012 слідує типовій моделі класичних згорткових мереж. MobileNet — це спрощена архітектура Xception, оптимізована для мобільних додатків. Наступні архітектури; ResNet, Inception і Xception стали точкою відліку для наступних досліджень штучного зору та навчання через його універсальність Deep.

Є багато факторів, які пояснюють революцію глибокого навчання, серед цих факторів виділяється; доступність наборів величезних даних і якості, паралельні обчислення (GPU) мають ефективну активацію для зворотного поширення, нові архітектури, нові методи регуляризації, які дозволяють навчати більш розгалужені мережі з меншою небезпекою перевищення, надійні оптимізатори та програмні платформи з великими спільнотами, такими як TensorFlow, Theano , Keras, CNTK, PyTorch, Chainer і Mxnet.

Усе це дозволило Expotecnología 2019 «Дослідження, інновації та розробки в інженерії» IOP Conf. Серія: Materials Science and Engineering 844

(2020) 012024 IOP Publishing doi:10.1088/1757-899X/844/1/012024 5 простіше розв'язувати задачі. Сьогодні мова програмування Python має велике значення в машинному навчанні порівняно з іншими мовами через підтримку структури глибокого навчання. У цю структуру входять TensorFlow, бібліотека програмного забезпечення з відкритим кодом для машинного навчання, яка дозволяє розгортати обчислення на центральному або графічному процесорі, розроблена Google, використовуючи дані потоку графіків, PyTorch використовує мову Python і має підтримку Facebook, Theano є Бібліотека Python, яка підтримує математичні вирази, що включають натягувачі, CNTK – це набір інструментів, розроблений Microsoft, відкритий для коду глибокого навчання, Keras – це бібліотека нейронних мереж високого рівня, створена Френсісом Шолле, членом Brain google обладнання, яке дозволяє вам вибирати, чи створені моделі виконуватимуться в Theano, TensorFlow або CNTK.

Keras і TensorFlow можуть будувати моделі трьома різними способами; використовуючи послідовну модель, функціональний API і попередньо навчені моделі. Раніше ми говорили про різні архітектури (MobileNet, Inception, ResNet, серед інших), тепер ми обговорюємо моделі для розпізнавання об'єктів і Keras TensorFlow; Швидше-CCN R, R-FCN, SSD і YOLO. Ці моделі класифікуються на основі детекторів у регіоні (швидші R-CNN, R-FCN, FPN) і одиночних детекторів (SSD і YOLO), починаючи з різних шляхів, але зараз вони виглядають дуже схожими, борючись за звання швидшого та точнішого детектора .

Існують різні показники, які можуть покращити алгоритми виявлення об'єктів на основі більш точного позиціонування, більшої швидкості та більш точної класифікації;

AI Tools	Функціональність	Підтримувані ОС / Мови / Платформа	Найкраща характеристика	Ціна

IBM Watson	Система відповідей на запитання.	на SUSE Linux Enterprise Server 11 OS Apache Hadoop.	Він багато дізнається з невеликих даних.	Безкоштовно
Платформа вмісту ДНК	Машинне навчання / Комп'ютерне бачення.	Підходить як для хмарних, так і для локальних моделей розгортання.	Машинне навчання без нагляду. Навчання щодо ваших даних.	Одноразова плата.
Google Cloud Machine Learning Engine	Машинне навчання	Консоль GCP	Моделі поїздів за вашими даними. Розгорніть його. Ви можете керувати цим.	За годину за навчальну одиницю витрати: США: 0,49 дол Європа: 0,54 дол Азіатсько-Тихоокеанський регіон: 0,54 дол
Студія машинного навчання Azure	Машинне навчання	На основі браузера	Модель буде розгорнуто як веб-сервіс.	Безкоштовно
TensorFlow	Машинне навчання	Робочі столи, Кластери, Мобільний, Крайові пристрої, процесори, Графічні та TPU.	Це для всіх, від початківців до експертів.	Безкоштовно
H2O AI	Машинне навчання	Поширюється в пам'яті Програмування Мови: R & Python.	Функція AutoML включена.	Безкоштовно
Кортана	Віртуальний помічник	ОС Windows, iOS, Android та Xbox. Підтримувані мови: англійська, португальська,	Він може виконувати стільки завдань, від встановлення нагадувань до увімкнення світла.	Безкоштовно

		французька, німецька, італійська, іспанська, китайська та японська.		
Salesforce Ейнштейна	CRM-система	На основі хмари.	Не потрібно керувати моделями та підготовкою даних.	Зв'яжіться з ними для отримання детальної інформації про ціни
Infosys Nia	Машинне навчання Чат-бот.	Підтримувані пристрої: Windows, Mac та Інтернет.	Він забезпечує три компоненти, тобто платформу даних, платформу знань та платформу автоматизації.	Зв'яжіться з ними для отримання детальної інформації про ціни.
Amazon Alexa	Віртуальний помічник	ОС: Fire OS, iOS та Android. Мови: англійська, французька, німецька, японська, італійська та іспанська.	Його можна підключити до таких пристроїв, як камера, світло та розважальні системи.	Безкоштовно з деякими пристроями або послугами Amazon.
Google Assistant	Віртуальний помічник	ОС: Android, iOS та KaiOS. Мови: англійська, хінді, індонезійська, французька, німецька, італійська, японська, корейська, португальська, іспанська, голландська та шведська.	Підтримує двосторонню розмову.	Безкоштовно

Таблиця 2 Порівняння функціональності

3.1.1. Google Collaboratory

Google Collaboratory або Colab. Це ще один хмарний сервіс від Google Research. Це IDE, яка дозволяє будь-якому користувачеві писати вихідний код у своєму редакторі та запускати його з браузера. Зокрема, він підтримує мову програмування Python і орієнтований на завдання машинного навчання, аналіз даних, освітні проекти тощо.

Базуючись на Jupyter Notebook, ця служба розміщується абсолютно безкоштовно з обліковим записом Gmail і не потребує налаштування, і вам не потрібно завантажувати чи встановлювати Jupyter. Він запропонує вам обчислювальні ресурси для редагування та тестування вашого коду, як-от GPGPU його серверів тощо. Очевидно, що як щось безкоштовне, Google Collaboratory не має та не гарантує необмежених ресурсів, але вони різняться залежно від використання, яке ставиться до системи.

Важливо зауважити, що коли отримуєте доступ до Colab за допомогою свого облікового запису, ви отримуєте віртуальну машину, де ви можете запускати свій код ізольовано від інших користувачів і ресурсів. Тож ви можете відновити віртуальну машину до початкового стану, якщо у вас виникнуть проблеми. Це також означає, що якщо ви запустите якийсь код у своїй віртуальній машині та закриєте браузер, машини буде видалено після періоду бездіяльності, щоб звільнити ресурси. Однак ви матимете свої блокноти в GDrive, якщо ви їх зберегли, або ви можете завантажити їх локально (формат Jupyter з відкритим кодом .ipynb).

Отримавши новий доступ до Google Collaboratory, знайдете інтуїтивно зрозуміле та просте у використанні середовище. Також містить покажчик із документацією та довідкою, а також кілька прикладів, які допоможуть вам почати, змінити коди, які ви вже створили, і розпочати тестування.

Colaboratory, або скорочено «Colab», є продуктом Google Research. Colab дозволяє будь-кому писати та виконувати довільний код Python через

браузер і особливо добре підходить для машинного навчання, аналізу даних і освіти.

Одною з переваг є те, що Colab можна використовувати безкоштовно. Ресурси Colab не гарантовані та необмежені, а обмеження використання іноді коливаються. Це необхідно для того, щоб Colab могла надавати ресурси безкоштовно. Щоб отримати додаткові відомості, перегляньте обмеження ресурсів

Ресурси в Colab мають пріоритет для інтерактивних випадків використання. Ми забороняємо дії, пов'язані з масовими обчисленнями, дії, які негативно впливають на інших, а також дії, пов'язані з обходом нашої політики. У середовищах виконання Colab заборонено:

- розміщення файлів, обслуговування медіа або інші пропозиції веб-служб, не пов'язані з інтерактивними обчисленнями з Colab
- завантажувати торренти або брати участь у одноранговому обміні файлами
- за допомогою віддаленого робочого столу або SSH
- підключення до віддалених проксі
- майнінг криптовалют
- запуск атак на відмову в обслуговуванні
- злом пароля
- використання кількох облікових записів для обходу обмежень доступу або використання ресурсів

3.1.2. TensorFlow

TensorFlow — це безкоштовна бібліотека програмного забезпечення з відкритим кодом для машинного навчання та штучного інтелекту. Він може використовуватися в ряді завдань, але зосереджений на навчанні та висновках глибоких нейронних мереж.

TensorFlow розроблено командою Google Brain для внутрішнього використання Google у дослідженнях і виробництві

TensorFlow можна використовувати в багатьох мовах програмування, включаючи Python, JavaScript, C++ і Java. Ця гнучкість дає змогу використовувати різноманітні програми в різних секторах.

TensorFlow є основною платформою та бібліотекою для машинного навчання. API TensorFlow використовують Keras, щоб дозволити користувачам створювати власні моделі машинного навчання. Окрім побудови та навчання моделі, TensorFlow також може допомогти завантажити дані для навчання моделі та розгорнути її за допомогою TensorFlow Serving.

Numpy є однією з найпопулярніших бібліотек даних Python, а TensorFlow пропонує інтеграцію та сумісність зі своїми структурами даних.[26] Numpy NDarrays, власний тип даних бібліотеки, автоматично перетворюються на тензори TensorFlow в операціях TF; те ж саме вірно і навпаки.[27] Це дозволяє двом бібліотекам працювати в унісон, не вимагаючи від користувача писати явні перетворення даних. Більше того, інтеграція поширюється на оптимізацію пам'яті завдяки тому, що тензори TF спільним використанням представлень пам'яті, що лежить в основі Numpy NDarrays, коли це можливо.[28]

3.1.3. Keras

Keras — це бібліотека програмного забезпечення з відкритим кодом, яка надає інтерфейс Python для штучних нейронних мереж. Keras діє як інтерфейс для бібліотеки TensorFlow.

До версії 2.3 Keras підтримував кілька серверних програм, зокрема TensorFlow, Microsoft Cognitive Toolkit, Theano та PlaidML. Починаючи з версії 2.4, підтримується лише TensorFlow. Розроблений для швидкого експериментування з глибокими нейронними мережами, він зосереджений на зручності, модульності та розширюваності. Його було розроблено в рамках

дослідницької роботи проекту ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), а його основним автором і супроводжувачем є Франсуа Шолле, інженер Google. Шолле також є автором моделі глибокої нейронної мережі Xception.

Keras містить численні реалізації часто використовуваних будівельних блоків нейронної мережі, таких як шари, цілі, функції активації, оптимізатори та безліч інструментів, які полегшують роботу з зображеннями та текстовими даними, щоб спростити кодування, необхідне для написання глибокого коду нейронної мережі. Код розміщено на GitHub, а форуми підтримки спільноти містять сторінку проблем GitHub і канал Slack.

Окрім стандартних нейронних мереж, Keras підтримує згорточні та рекурентні нейронні мережі. Він підтримує інші загальні службові рівні, такі як відключення, пакетна нормалізація та об'єднання.

Keras дозволяє користувачам створювати глибокі моделі на смартфонах (iOS і Android), в Інтернеті або на віртуальній машині Java. Це також дозволяє використовувати розподілене навчання моделей глибокого навчання на кластерах графічних процесорів (GPU) і тензорних процесорів (TPU).

3.1.4. NumPy

NumPy — це бібліотека для мови програмування Python, яка підтримує великі багатовимірні масиви та матриці, а також з великою колекцією математичних функцій високого рівня для роботи з цими масивами. Попередник NumPy, Numeric, спочатку був створений Джимом Хугуніним за участю кількох інших розробників. У 2005 році Тревіс Оліфант створив NumPy, включивши функції конкуруючого Numarray у Numeric із великими змінами. NumPy є програмним забезпеченням з відкритим вихідним кодом і має багато учасників. NumPy – це проект, який фінансується NumFOCUS.

NumPy націлений на еталонну реалізацію Python CPython, яка є неоптимізованим інтерпретатором байт-коду. Математичні алгоритми,

написані для цієї версії Python, часто працюють набагато повільніше, ніж скомпільовані еквіваленти через відсутність оптимізації компілятора. NumPy частково вирішує проблему повільності, надаючи багатовимірні масиви, функції та оператори, які ефективно працюють з масивами; їх використання вимагає переписування коду, переважно внутрішніх циклів, за допомогою NumPy.

Використання NumPy у Python надає функціональність, порівнянну з MATLAB, оскільки вони обидва інтерпретуються, і вони обидва дозволяють користувачеві писати швидкі програми, якщо більшість операцій працюють із масивами чи матрицями, а не зі скалярами. Для порівняння, MATLAB може похвалитися великою кількістю додаткових наборів інструментів, зокрема Simulink, тоді як NumPy внутрішньо інтегрований з Python, більш сучасною та повною мовою програмування. Крім того, доступні додаткові пакети Python; SciPy — це бібліотека, яка додає більше функціональних можливостей, подібних до MATLAB, а Matplotlib — це пакет для побудови графічних зображень, який забезпечує функціональні можливості побудови, подібні до MATLAB. Внутрішньо і MATLAB, і NumPy покладаються на BLAS і LAPACK для ефективних обчислень лінійної алгебри.

Прив'язки Python широко використовуваної бібліотеки комп'ютерного зору OpenCV використовують масиви NumPy для зберігання та роботи з даними. Оскільки зображення з декількома каналами просто представлені як тривимірні масиви, індексація, нарізка або маскування за допомогою інших масивів є дуже ефективними способами доступу до конкретних пікселів зображення. Масив NumPy як універсальна структура даних у OpenCV для зображень, вилучених точок функцій, ядер фільтрів та багато іншого значно спрощує робочий процес програмування та налагодження.

3.2. ImageNet

Набір даних ImageNet містить 14 197 122 анотованих зображення відповідно до ієрархії WordNet. З 2010 року набір даних використовується в

ImageNet Large Scale Visual Recognition Challenge (ILSVRC), еталону класифікації зображень і виявлення об'єктів. Публічно оприлюднений набір даних містить набір навчальних зображень, анотованих вручну. Також опубліковано набір тестових зображень, анотації вручну не надаються. Анотації ILSVRC належать до однієї з двох категорій: анотація на рівні зображення бінарної мітки для наявності чи відсутності класу об'єктів у зображенні, наприклад, «на цьому зображенні є машини», але «немає тигрів», а анотація на рівні об'єкта щільної обмежувальної рамки та мітки класу навколо екземпляра об'єкта на зображенні, наприклад, “є викрутка, центрована в позиції (20,25) із шириною 50 пікселів і висотою 30 пікселів. Проект ImageNet не володіє авторськими правами на зображення, тому надаються лише мініатюри та URL-адреси зображень.

3.3.Метрики оцінки ефективності семантичної обробки

Оціночні метрики дозволяють отримати числовий показник якості обраного методу, що допоможе при порівнянні різних методів між собою. Для різних завдань, пов'язаних із даними, існують різні показники. Серед них для задачі сегментації зображення можна виділити піксельну точність (pixel accuracy), середню точність (mean accuracy), mean Intersection-Over-Union (mIoU), частотно-зважену intersection-over- шляхом комбінування (frequency- зважений IoU) і середній показник F1 (середній показник f1).

3.3.1. Точність пікселів(pixel accuracy)

Альтернативним показником для оцінки семантичної сегментації є просто звіт про відсоток пікселів у зображенні, які були правильно класифіковані. Точність пікселів зазвичай повідомляється для кожного класу окремо, а також глобально для всіх класів.

Розглядаючи точність пікселів для кожного класу, ми, по суті, оцінюємо двійкову маску; справжній позитивний результат представляє піксель, який правильно передбачено належати до даного класу (відповідно

до цільової маски), тоді як істинний негативний представляє піксель, який правильно визначено як такий, що не належить до даного класу.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Ця метрика іноді може надавати оманливі результати, коли представлення класу в зображенні невелике, оскільки вимірювання буде необ'єктивним, головним чином повідомляючи про те, наскільки добре ви ідентифікуєте негативний випадок (тобто, коли клас відсутній).

3.3.2. Перетин через Союз (Intersection over Union (IoU))

Метрика Intersection over Union (IoU), також відома як індекс Жаккара, по суті, є методом кількісного визначення відсотка перекриття між цільовою маскою та результатом нашого прогнозу. Ця метрика тісно пов'язана з коефіцієнтом Дайса, який часто використовується як функція втрат під час навчання.

Простіше кажучи, показник IoU вимірює кількість спільних пікселів між цільовою та прогнозованою масками, поділену на загальну кількість пікселів, присутніх в обох масках.

$$IoU = \frac{\text{target} \cap \text{prediction}}{\text{target} \cup \text{prediction}}$$

Як наочний приклад, припустімо, що нам доручено обчислити оцінку IoU наступного прогнозу, враховуючи маску основної правди, позначену міткою.

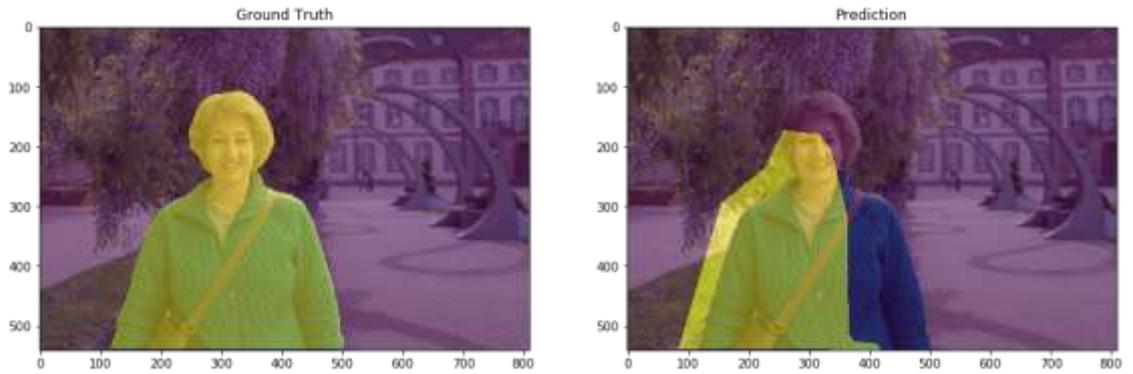


Рис. 3.1 Правильна сегментація та передбачення

Перетин ($A \cap B$) складається з пікселів, знайдених як у масці передбачення, так і в основній масці істинності, тоді як об'єднання ($A \cup B$) складається просто з усіх пікселів, знайдених або в прогнозованій, або в цільовій масці.

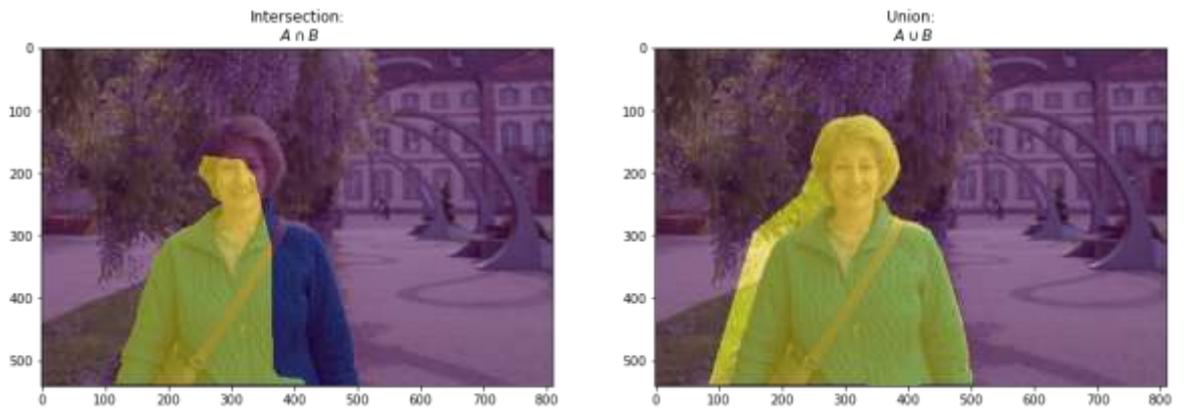


Рис. 3.2 Перетин та об'єднання

Показник IoU обчислюється для кожного класу окремо, а потім усереднюється для всіх класів, щоб отримати глобальний середній показник IoU нашого прогнозу семантичної сегментації.

3.3.3. Точність(Precision)

Точність – це те, наскільки точна модель, тобто це міра якості прогнозів, які ми робимо. Точність ефективно описує чистоту наших позитивних виявлень відносно основної правди.

Щоб обчислити його, нам потрібно мати доступ до істинних і прогнозованих позитивних результатів (як істинних, так і хибних):

$$Precision = \frac{TP}{TP + FP}$$

3.3.4. Recall

Тепер проблема з прогнозуванням лише одного пікселя з класу об'єктів, навіть якщо він правильний, полягає в тому, що ми втрачаємо більшість пікселів об'єктів, тобто запам'ятовування є дуже низьким.

Щоб обчислити його, нам потрібно мати доступ до справжніх позитивних результатів і всіх позитивних елементів (як правильно, так і неправильно передбачених):

Якщо точність можна порівняти з якістю моделі, то відкликання схоже на кількість, тобто скільки справжніх міток було знайдено.

Отже, на прикладі незбалансованого зображення, чим більше пікселів ми прогнозуємо як справжній клас, тим краще буде запам'ятовування.

Коефіцієнт Dice (оцінка F1)

Простіше кажучи, Коефіцієнт кубика дорівнює $2 * \text{Площа перекриття}$, поділена на загальну кількість пікселів в обох зображеннях.

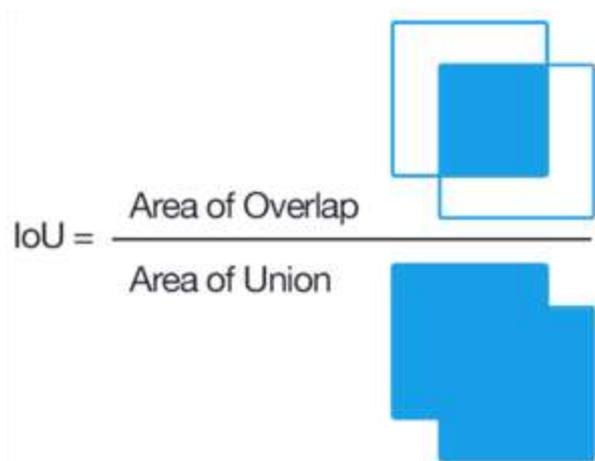


Рис.3.3 Ілюстрація коефіцієнта Dice. $2 \times \text{Overlap}$ / Загальна кількість пікселів

Реалізація Dice Coefficient у keras:

```
def dice_coef(y_true, y_pred, smooth=1):
    перетин = K.sum(y_true * y_pred, axis=[1,2,3])
    об'єднання = K.sum(y_true, axis=[1,2,3]) + K.sum(y_pred, axis=[1,2,3])
    dice = K.mean((2. * перетин + згладжування)/(об'єднання +
згладжування), вісь=0)
    повернути кубики
```

3.4.Ефективність обробки зображень з будівельними об'єктами із аерофотознімків

Inria Aerial Image Labeling розглядає основну тему дистанційного зондування: автоматичне попиксельне маркування аерофотознімків.

Особливості набору даних:

Покриття 810 км² (405 км² для навчання та 405 км² для тестування)

Кольоровий аерофотознімок з просторовою роздільною здатністю 0,3 м

Базові істинні дані для двох семантичних класів: будівельних і небудівних (публічно оприлюднені лише для навчальної підмножини)

Зображення охоплюють різні міські поселення, починаючи від густонаселених районів (наприклад, фінансовий район Сан-Франциско) і закінчуючи альпійськими містами (наприклад, Лієнц в австрійському Тіролі).

Замість поділу суміжних частин однакових зображень на навчальні та тестові підмножини, різні міста включені в кожен з підмножин. Наприклад, зображення над Чикаго входять до навчального набору (а не до тестового), а зображення над Сан-Франциско включені до тестового набору (а не до навчального). Кінцевою метою цього набору даних є оцінка узагальнюючої потужності методів: хоча зображення Чикаго можна використовувати для навчання, система повинна позначати аерофотознімки над іншими регіонами з різними умовами освітлення, міським ландшафтом і часом року.

Набір даних було створено шляхом об'єднання зображень із загальнодоступними зображеннями та слідами офіційних будівель, які є загальнодоступними.

Щоб перевірити продуктивність запропонованих методів для визначення будівельних об'єктів із аерофотознімків з високою роздільною здатністю дистанційного зондування, ми порівняли результати. Зокрема, ми порівняли підхід глибокого згорткового кодера-декодера, який називається моделлю Segnet, технікою FCN і глибокою згортковою моделлю Unet. Порівнюючи результати, отримані за допомогою моделей Segnet і Unet, із результатами запропонованої моделі SegUnet, можна побачити різницю в точності вилучення будівель. Однак підходи FCN і Segnet передбачають додаткові FN і FP, які зображені синім і зеленим кольорами відповідно. Таким чином, ці методи не можуть точно зберегти та отримати граничну інформацію, що призводить до виявлення FN та FP та створення карти сегментації з низькою роздільною здатністю. Модель Unet, яка використовує шари деконволюції та пропуск з'єднання, також може отримати та зберегти інформацію про межі з вищою точністю, ніж методи FCN та Segnet, таким чином отримуючи правильну карту сегментації

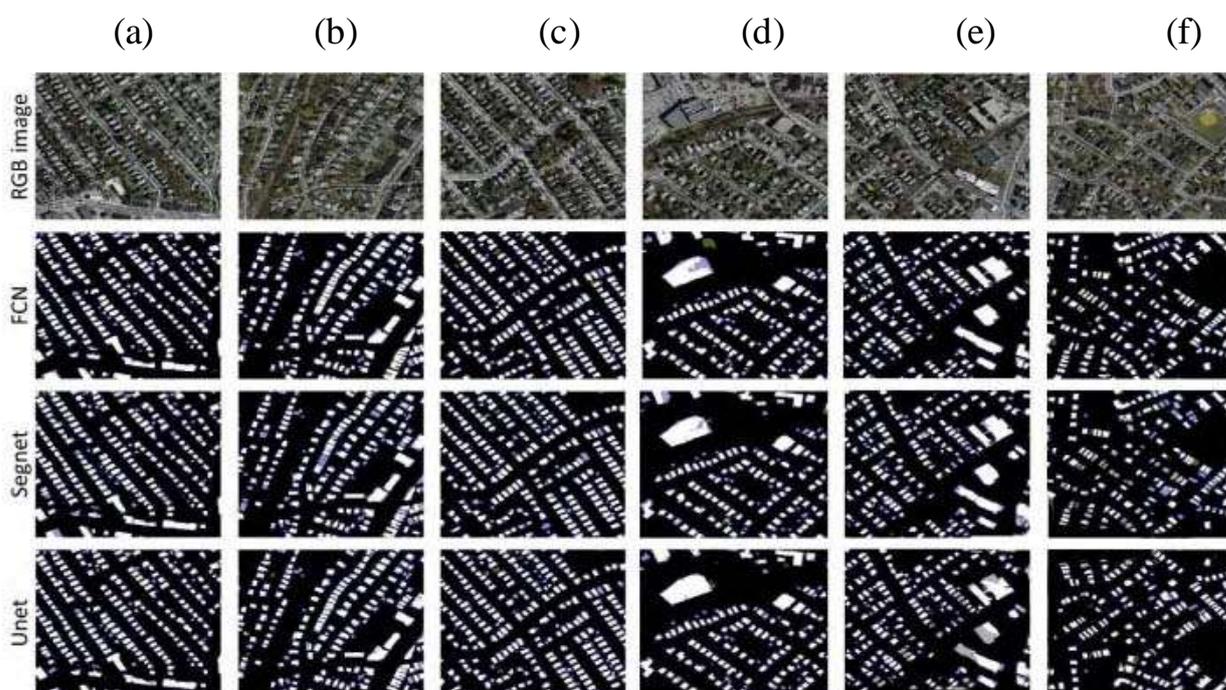


Рис.3.4 Результати проведення обробки зображення обраними методами

Щоб перевірити ефективність методів, кількісні результати методів продемонстровані у Таблиці 3. Перші рядки таблиці 3 представляють кількісну точність трьох основних показників, досягнуту порівняльним підходи для восьми зображень, тоді як останній рядок представляє середню точність показників. Як показано в таблиці, модель FCN може отримати вищу точність для коефіцієнта пригадування, ніж інші методи, оскільки модель передбачає багато FN. Навпаки, метод Unet може отримати вищу точність щодо точності та факторів OA, ніж методи FCN та Segnet. Крім того, метод Unet є найкращим підходом у вилученні побудови та може отримати правильну карту сегментації. На показано чіткі відмінності між підходами глибокого навчання для побудови сегментації об'єктів.

		FCN	Segnet	Unet
Зображення а	Recall	0.828	0.7852	0.8028
	precision	0.8492	0.8375	0.8704
	F1	0.8384	0.8105	0.8352
	Accuracy	0.92	0.908	0.9206
Зображення б	Recall	0.7908	0.7353	0.7525
	precision	0.8362	0.8371	0.8663
	F1	0.8128	0.7829	0.8054
	Accuracy	0.908	0.897	0.9081
Зображення с	Recall	0.7982	0.7238	0.7738
	precision	0.8541	0.865	0.8936
	F1	0.8252	0.7882	0.8294
	Accuracy	0.9179	0.9055	0.9227
Зображення д	Recall	0.8029	0.7848	0.7859
	precision	0.8497	0.8569	0.8694
	F1	0.8256	0.8192	0.8255
	Accuracy	0.914	0.9121	0.9157

Зображення e	Recall	0.8195	0.7838	0.8154
	precision	0.8534	0.8758	0.878
	F1	0.8361	0.8272	0.8456
	Accuracy	0.9311	0.9297	0.9361
Зображення f	Recall	0.846	0.7877	0.8428
	precision	0.8585	0.8691	0.8939
	F1	0.8522	0.8264	0.8676
	Accuracy	0.9332	0.9247	0.9415
Середнє	Recall	0.8233	0.7721	0.8079
	precision	0.8355	0.8450	0.8595
	F1	0.8288	0.8054	0.8318
	Accuracy	0.9229	0.9156	0.9259

Таблиця 3 Отримані результати дослідження

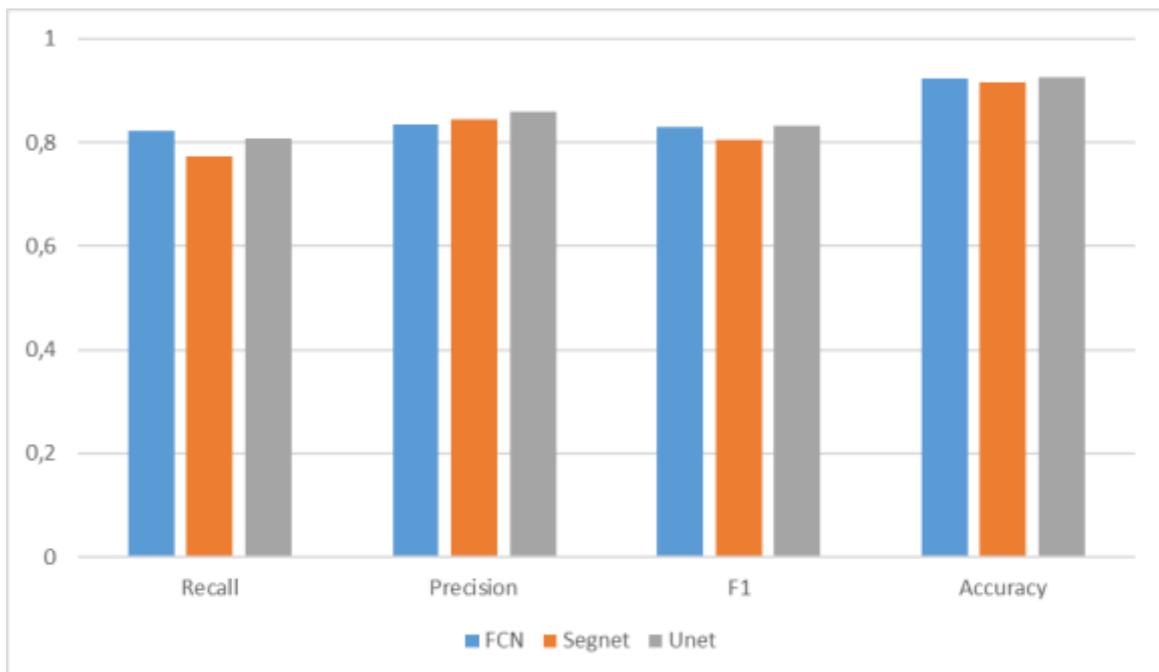


Рис. 3.5 Графік порівняння отриманих результатів

Було застосовано запропоновані методи до масиву будівельних даних штату Массачусетс. Після навчання та перевірки методів ми використали чотири показники точності, щоб оцінити ефективність під час вилучення будівель результатами інших методів глибокого навчання, таких як методів

Segnet, FCN і Unet, щоб продемонструвати їх ефективність. Результати показали, що серед запропонованих методів отримав найкращі кількісні показники та візуальні показники та перевершив інші DCNN у вилученні будівель із аерофотознімків високої роздільної здатності саме Unet.

3.5. Ефективність обробки зображень з водними об'єктами

Для досліджуваних ділянок отримано остаточні класифікаційні карти різними методами.

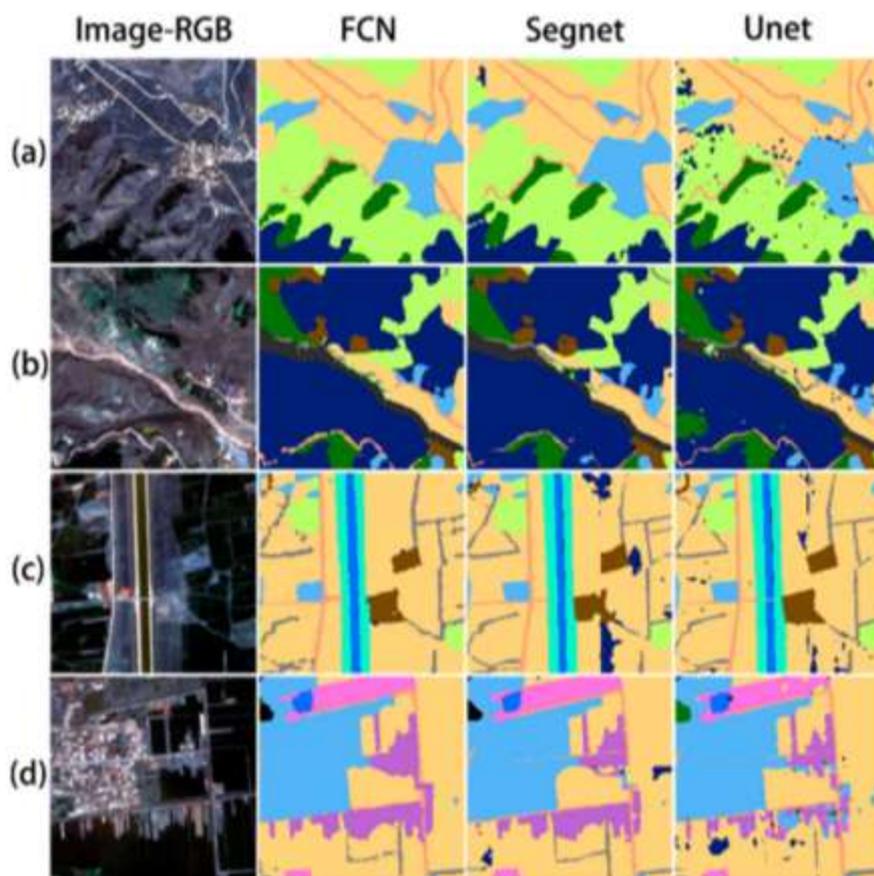


Рис. 3.6 Результати проведення обробки зображення обраними методами

На відміну від різних вхідних діапазонів, результати класифікації FCN у всіх місцях дослідження демонструють згладжені візуальні ефекти з найменшим спекл-шумом. Ми не змогли знайти очевидних відмінностей у результатах класифікації під час візуальної оцінки. На відміну від Segnet і FCN, виявили, що результати класифікації Unet містили кілька небажаних

шумів. Деякі пікселі в лінійному класі також не були чітко класифіковані. Що стосується результатів Segnet і FCN, усі проблеми неправильної класифікації, створені моделлю Unet, були вирішені з більшою точністю. Таким чином, неправильні класифікації, які з'являються, були відповідно переглянуті.

		FCN	Segnet	Unet
Зображення а	Recall	0.97	0.88	0.96
	precision	0.897	0.98	0.95
	F1	0.93	0.92	0.95
	Accuracy	0.93	89.9	87.9
Зображення б	Recall	0.93	0.91	0.85
	precision	0.91	0.84	0.85
	F1	0.92	0.88	0.85
	Accuracy	0.92	0.86	0.85
Зображення с	Recall	0.83	0.75	0.84
	precision	0.95	0.99	0.88
	F1	0.89	0.85	0.86
	Accuracy	0.92	0.85	0.85
Зображення д	Recall	0.91	0.91	0.90
	precision	0.94	0.96	0.98
	F1	0.93	0.93	0.94
	Accuracy	0.92	0.86	0.85
Середнє	Recall	0.912	0.862	0.887
	precision	0.922	0.942	0.915
	F1	0.917	0.895	0.975
	Accuracy	0.939	0.899	0.879

Таблиця 4 Отримані результати дослідження

З таблиці, розглядаючи метрики, ми виявили, що підхід FCN повідомив про найкращу класифікацію серед Segnet та Unet. У середньому по моделях ОА були на 0,06 вище в FCN, ніж у двох інших моделях. У підході FCN

Точніше, враховуючи точність індивідуального водних об'єктів в різних експериментах, показники Recall, Precision, F1, Accuracy працювали по-різному. Як показано в таблицівище, Метод FCN показав найкращу продуктивність.

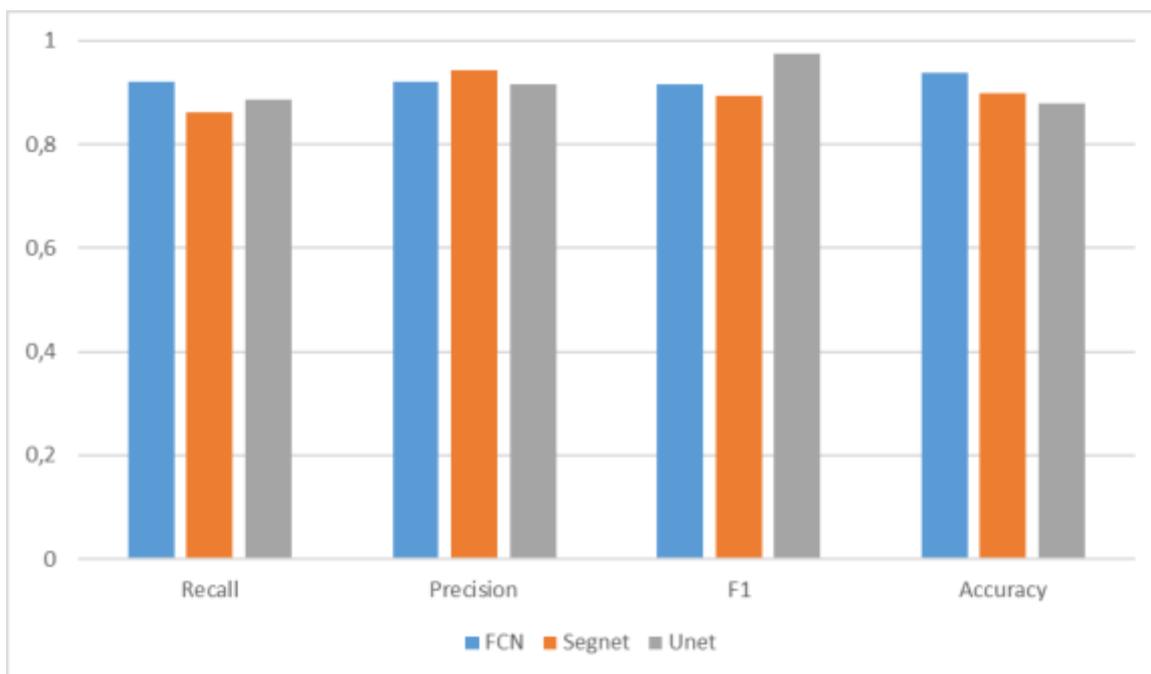


Рис. 3.7 Графік порівняння отриманих результатів

ВИСНОВОК

Вирішення питання семантичної обробки зображень є актуальною як ніколи, адже це важливо для подальшого розвитку детального розуміння об'єктів, присутніх на зображенні, а з розвитком технологій штучного інтелекту та відкриттям нових горизонтів, його актуальність ще більше зростає.

Для різних галузей промисловості за останні десятиліття було розроблено чимало методів за допомогою яких є можливість обробляти зображення. Для різних підходів існують свої методи які по тому чи іншому показнику відрізняються в своїй ефективності.

Для досягнення поставлених цілей в ході цієї роботи було вирішенні наступні завдання і зроблено наступні кроки:

1. Досліджено та проаналізовано літературу, що стосується сприйняття зображення комп'ютером.
2. Було проведено огляд загальних питань, пов'язаних з постановкою задачі та огляду основних проблем даного дослідження.
3. Розглянуто поширені методи семантичної обробки зображень на основі різних підходів
4. Порівняли результати даних отриманих при обробці зображень будівельних об'єктів із аерофотознімків та зображень з водними об'єктами

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Carrington D. How Many Photos Will Be Taken in 2023? *Tech today*. 2023. URL: <https://focus.mylio.com/tech-today/how-many-photos-will-be-taken-in-2023>
2. Робототехника К. Фу, Р. Гонсалес, К. Ли. М.: Мир, 1989.
3. Дуда Р.О., Харт П.Є. Розпізнавання образів та аналіз сцен. 1976. 511 ст.
4. Вапник В.М., Червоненкіс А.Я. Теорія розпізнавання образів. Стохастичні проблеми навчання. Наука, 1974. 415 ст.
5. Денисов, Д. А. Сегментація зображень на ЕВМ/ Д. А. Денисов, В. А. Нізовкин // Зарубіжна радіоелектроніка. 1985.
6. Linda G. Shapiro and George C. Stockman (2001): "Computer Vision", pp 279–325, New Jersey, Prentice-Hall, ISBN 0-13-030796-3
7. Barghout, Lauren, and Lawrence W. Lee. "Perceptual information processing system." Paravue Inc. U.S. Patent Application 10/618,543, filed July 11, 2003.
8. Nielsen, Frank; Nock, Richard (2003). "On region merging: The statistical soundness of fast sorting, with applications". 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. Vol. 2. IEEE. pp. II:19–26. doi:10.1109/CVPR.2003.1211447. ISBN 0-7695-1900-8.
9. Zachow, Stefan, Michael Zilske, and Hans-Christian Hege. "3D reconstruction of individual anatomy from medical image data: Segmentation and geometry processing." (2007).
10. K. Hara, V. Jagadeesh, R. Piramuthu. Fashion Apparel Detection: The Role of Deep Convolutional Neural Network and Pose-dependent Priors. URL: <http://arxiv.org/pdf/1411.5319v2.pdf> (Дата звернення: 11.05.2022).

11. Денисов Д.А. Комп'ютерні методи аналізу відеоінформації, монографія. Красноярськ, Краснояр. держ. техн. Університет. 1993, 192 с. [Denisov D.A. Комп'ютерні методи для аналізу інформації про відео, monograph. Красноярськ, Краснояр. Gos. tehn. University, 1993, 192 p. (in Ukrainian)]
12. Денисов Д.А., Дудкін А.К., П'яткін В.П. Структурні методи опису об'єктів зображень. ВЦ З АН СРСР. Новосибірськ, 1988. 35 с. [Denisov D.A., Dudkin K.A., Pyatkin V.P. Структурні методи опису зображення об'єктів, комп'ютерних центрів Siberian branch of Russian Academy of Sciences USSR. Novosibirsk, 1988 (in Ukrainian)]
13. Борисенко В.І., Златопольський А.А., Мучник І.Б. Сегментація зображень (стан проблеми). Автоматика і телемеханіка, 1987, 7, 3-56 [Борисенко В.І., Златопольська А.А., Мухнік І.Б. Image segmentation (state of the problem), Automation and remote control, 1987, 7, 3-56 (in Ukrainian)]
14. Бакут П.А., Колмогоров Г.С., Варновицький І.Е. Сегментація зображень: методи порогової обробки. Зарубіжна радіоелектроніка, 1987, 10, 54-61 [Bakut P. A., Kolmogorov G. S., Vornovitsky I. E. Image segmentation. thresholding методів. Foreign Radioelectronics, 1987, 10, 54-61 (in Ukrainian)]
15. Прет У. Цифрова обробка зображень, в 2 т. М., Світ, 1982. 790 с. [Pratt W. Digital image processing, in 2 vol. M., Mir, 1982. 790 p. (in Ukrainian)]
16. Яншин В.В. Аналіз та обробка зображень. принципи та алгоритми, М., Машинобудування, 1994. 112 с. [Yanshin V.V. Analysis and processing of images. principles and algorithms, M. Mashinostroenie, 1994, 112 p. (in Ukrainian)]
17. Коулмен Г.Б., Ендрюс Х.С. Сегментація зображень за допомогою автоматичної класифікації. ТИЕР. 1979, 5, 82-97. [Coleman G.B.,

- Andrews H.C. The image segmentation by automatic classification. ТИЕР. 1979, 5, 82-97 (in Ukrainian)]
18. Brian L. DeCost, Elizabeth A. Holm A computer vision approach for automated analysis and classification of microstructural image data. Computational Materials Science 110, 2015, 126-133.
 19. Khodaskar A., Ladhake S. Semantic Image Analysis for Intelligent Image Retrieval International Conference on Intelligent. Computing, Communication & Convergence Institute of Management and Technology, India 2014, 193-197.
 20. Banerjee S.K., Ghosh S., Datta S.K. Segmentation of dual phase steel micrograph. An automated approach. Measurement 2013, 2435-2440.
 21. Murphy A.G., Browne D.J., Mirihanage W.U., Mathiesen R.H. Combined in situ X-ray radiographic observations and post-solidification metallographic characterisation of eutectic transformations in Al-Cu alloy systems, Acta materialia 61, 2013, 4559-4571.
 22. Tolnai D., Requena G., Cloetens P., Lendvai J., Degischer H.P. Effect of solution heat treatment on the internal architecture and compressive strength of an AlMg4.7Si8 alloy. Materials Science & Engineering, A 585, 2013, 480-487.
 23. Попередня обробка зображень. Моделювання та розпізнавання 2D/3D образів: веб-сайт. URL: <https://bit.ly/3eOC54e> (дата звернення: 06.05.2022).
 24. Ратайчак, Р.; Кріспім-Джуніор, С.Ф.; Форе, Е.; Ферверс, Б.; Tougne, L. Автоматична реконструкція земельного покриву з історичних аерофотознімків: оцінка алгоритмів вилучення ознак і класифікації. IEEE Trans. Обробка зображення. 2019, 28, 3357–3371. [CrossRef] [PubMed]
 25. Chen, J.; Chen, J.; Liao, A.; Cao, X.; Chen, L.; Chen, X.; He, C.; Han, G.; Peng, S.; Lu, M.; et al. Global land cover mapping at 30 m resolution:

- A POK-based operational approach. *ISPRS J. Photogramm. Remote Sens.* 2015, 103, 7–27. [CrossRef]
26. Wulder, M.A.; Hermosilla, T.; Stinson, G.; Gougeon, F.A.; White, J.C.; Hill, D.A.; Smiley, B.P. Satellite-based time series land cover and change information to map forest area consistent with national and international reporting requirements. *For. Int. J. For. Res.* 2020, 93, 331–343. [CrossRef]
27. Griffiths, P.; van der Linden, S.; Kuemmerle, T.; Hostert, P. A Pixel-Based Landsat Compositing Algorithm for Large Area Land Cover Mapping. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 2013, 6, 2088–2101. [CrossRef]
28. Hauser, L.T.; Nguyen Vu, G.; Nguyen, B.A.; Dade, E.; Nguyen, H.M.; Nguyen, T.T.Q.; Le, T.Q.; Vu, L.H.; Tong, A.T.H.; Pham, H.V. Uncovering the spatio-temporal dynamics of land cover change and fragmentation of mangroves in the Ca Mau peninsula, Vietnam using multi-temporal SPOT satellite imagery (2004–2013). *Appl. Geogr.* 2017, 86, 197–207. [CrossRef]
29. Karakizi, C.; Karantzalos, K.; Vakalopoulou, M.; Antoniou, G. Detailed Land Cover Mapping from Multitemporal Landsat-8 Data of Different Cloud Cover. *Remote Sens.* 2018, 10, 1214. [CrossRef] Lobser, S.E.; Cohen, W.B. MODIS tasselled cap: Land cover characteristics expressed through transformed MODIS data. *Int. J. Remote Sens.* 2007, 28, 5079–5101. [CrossRef]