МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____Ю. П. Кондратенко
«____»_____2022 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

ДОСЛІДЖЕННЯ ВПЛИВУ АРХІТЕКТУР ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ НА ЕФЕКТИВНІСТЬ СЕГМЕНТАЦІЇ ОБ'ЄКТІВ

Спеціальність 122 «Комп'ютерні науки»

122 – МКР – 601.21610161

Студент _____ М. А. Костиря

«14» лютого 2022 р.

Консультант _____ Ю. П. Кондратенко
д-р техн. наук, проф.

«14» лютого 2022 р.

Миколаїв – 2022

Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Освітньо-кваліфікаційний рівень **магістр**
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри інтелектуальних інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
«__» _____2022 р.

ЗАВДАННЯ
на магістерську кваліфікаційну роботу

Костиря Михайлу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи: Дослідження впливу архітектур згорткових нейронних мереж на ефективність сегментації об'єктів .

Керівник роботи Кондратенко Ю.П., д-р техн. наук, професор .
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затв. наказом Ректора ЧНУ ім. Петра Могили від «___» _____20__р. № 228____

2. Строк подання студентом роботи 14.02.2022 р.

3. Вхідні (початкові) дані до роботи: набір даних з зображеннями міста і сегментаційними масками .

Очікуваний результат роботи: система, повертаюча сегментаційну маску на виході. .

4. Зміст пояснювальної записки (перелік питань, які потрібно розглянути): аналіз сучасного стану задачі комп'ютерного зору та сегментації зображень; дослідження можливих технологій та підходів для вирішення поставленої задачі; розробка програмної реалізації з використанням обраних методів лоя сегментації об'єктів; аналіз отриманих результатів.

5. Перелік графічних матеріалів: презентація _____
_____.

6. Завдання до спеціальної частини: Оцінка умов праці та забезпечення безпеки персоналу ТОВ “Посмішка” в умовах надзвичайної ситуації _____.

7. Консультанти:

Розділ	**Прізвище, ініціали та посада консультанта**	**Підпис**
Методичний	Кондратенко Ю.П., д-р техн. наук, професор	
З охорони праці	Щербак Ю.Г., к.т.н., доцент	

Керівник роботи _____ д-р техн. наук, проф., КондратенкоЮ.П.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____
(підпис)

Завдання прийнято до виконання _____Костиря М.А.
(прізвище та ініціали)

_____
(підпис)

Дата видачі завдання «__20__» _____жовтня_____ 2021 р.

КАЛЕНДАРНИЙ ПЛАН
Виконання магістерської кваліфікаційної роботи

Тема: Дослідження впливу архітектур згорткових нейронних мереж на ефективність сегментації об’єктів

№	Найменування роботи	Початок	Закінчення	Примітки
1	Визначення керівника і теми МКР. Подання заяви на затвердження теми МКР	01.09.2021	10.10.2021	
2	Отримання завдання на виконання МКР	19.10.2021	22.10.2021	
3	Складання календарного плану на період виконання МКР	23.10.2021	26.10.2021	
4	Огляд літератури за темою дослідження	27.10.2021	10.11.2021	
5	Проходження переддипломної практики, збір та аналіз матеріалів до МКР	22.11.2021	11.12.2021	
6	Аналіз предметної області та розробка технічного завдання. Моделювання результатів	16.12.2021	12.01.2022	
7	Опис фахової частини МКР, зокрема дослідження публікацій, огляд існуючих архітектур згорткових нейронних мереж, реалізація обраних технологій з аналізом отриманих результатів	13.01.2022	25.01.2022	
8	Розробка спеціальної частини з охорони праці та методичної частини	26.01.2022	30.01.2022	
9	Попередній захист МКР на засіданні комісії кафедри	31.01.2022	31.01.2022	
10	Корегування роботи за результатами попереднього захисту	01.02.2022	03.02.2022	
11	Остаточне оформлення пояснювальної записки та слайдів доповіді для захисту	04.02.2022	06.02.2022	
12	Подання МКР рецензенту	09.02.2022	10.02.2022	
13	Рецензування МКР	11.02.2022	12.02.2022	
14	Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	14.02.2022	15.02.2022	
15	Захист МКР перед екзаменаційною комісією (ЕК)	21.02.2022	22.02.2022	

Розробив студент _____Костиря М.А._____ _____
(прізвище та ініціали) (*підпис*)

Керівник роботи _____д.т.н., професор Кондратенко Ю.П._____ _____
(*наук. ступінь, вчене звання, прізвище та ініціали)* *(підпис)*

«23» жовтня 2021 р.

АНОТАЦІЯ

до магістерської кваліфікаційної роботи
студента групи 601 ЧНУ ім. Петра Могили

Костирі Михайла Андрійовича

на тему: **“ДОСЛІДЖЕННЯ ВПЛИВУ АРХІТЕКТУР ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ НА ЕФЕКТИВНІСТЬ СЕГМЕНТАЦІЇ ОБ’ЄКТІВ”**

Актуальність даного дослідження обумовлена тим, що сегментація здобула широкого використання у різних сферах, приклад в медицині. Одним із сценаріїв використання таких систем сегментації є виділення окремих клітин на зображенні з подальшою класифікацією на ракові та здорові.

Об’єктом дослідження процес семантичної сегментації зображення.

Предметом дослідження є структури, моделі та архітектури згорткових нейронних мереж.

Метою роботи є дослідження та порівняльний аналіз впливу архітектур згорткових нейронних мереж на ефективність сегментації об’єктів.

В результаті виконання роботи було розроблено декілька систем, що надають можливість обробляти фото, класифікуючи об’єкти на піксельному рівні. Цей процес називається семантичною сегментацією зображення. Ідея декількох програмних реалізацій полягає у подальшому порівнянні різних моделей та підходів.

Дана робота складається з фахової частини, спеціальної частини з охорони праці та методичної частини. Пояснювальна записка магістерської кваліфікаційної роботи складається зі вступу, трьох розділів та висновків. У першому розділі розкрито важливість систем комп’ютерного зору у сучасному світі. У другому розділі описуються існуючі технології та алгоритми, також виконано порівняння методів з метою виявлення недоліків. У третьому розділі описано проектування та програмну реалізацію розробленої системи. Порівняно результати в залежності від обраної архітектури. Загальний обсяг роботи – 77 сторінок. Магістерська кваліфікаційна робота містить один додаток, 27 рисунків, 21 таблицю і посилання на 45 літературних джерел.

Ключові слова: комп’ютерний зір, штучний інтелект, семантична сегментація, нейронна мережа.

ABSTRACT

to the master's qualification work by the student of the group 601 of Petro Mohyla Black Sea National University

Mykhailo Kostyria

“STUDYING THE IMPACT OF ARCHITECTURE ON THE FINAL RESULTS OF CONVOLUTIONAL NEURAL NETWORKS”

A relevance of this study lies in the popularity of segmentation in different areas, for example in medical problems. One of the scenarios for usage of such segmentation systems is separation of cells on the image, following with classification.

An object of research is the process of semantic segmentation.

A subject of the research are the structures, models and architectures of convolutional neural networks.

A purpose of the study is studying and the resulting analysis of the impact of architecture on the final results of convolutional neural networks.

As a result of the work, a few systems that allow computations on photos were developed. Such systems classify objects on pixel level, this process is called semantic segmentation. The idea of developing a few systems lies in the further analysis and studying.

This work consists of five sections. Each of them is devoted to: analysis of the subject area, mathematical models and methods used in the thesis, modeling and design of the system, labor protection and life safety, methodological part of the master's work.

The overall scope of the work is 77 pages. Thesis contains 1 application, 27 figures, 21 tables and 45 sources in it.

Key words: computer vision, artificial intelligence, semantic segmentation, neural network.

ВСТУП

Одним із найбільш потужних і актуальних типів штучного інтелекту є комп’ютерний зір. Ця сфера фокусується на реплікації частин візуальної системи людини, надаючи комп’ютерам можливість ідентифікувати та обробляти об’єкти на зображеннях та відео. До недавніх часів, комп’ютерний зір мав обмежені можливості.

Завдяки нещодавнім відкриттям в області штучного інтелекту і машинного навчання, сфера комп’ютерного зору здобула значного розвитку і навіть перевершила людину у деяких завданнях, що стосуються детекції та класифікації об’єктів. Одним із основних факторів, що сприяли стрімкому розвитку комп’ютерного зору, є значна кількість дати, що ми генеруємо на сьогоднішній день.

У рамках даної роботи було розроблено декілька систем, що надають комп’ютеру можливість обробляти фото, класифікуючи об’єкти на піксельному рівні. Цей процес називається семантичною сегментацією зображення. Сегментація здобула широкого використання у ряді різних сфер, як приклад можна навести медицину. Одним із сценаріїв використання таких систем є виділення окремих клітин на зображенні, з подальшою класифікацією на ракові, та здорові. Ідея декількох програмних реалізацій полягає у подальшому порівнянні різних моделей та підходів.

Як результат було сплановано, створено та протестовано систему для семантичної сегментації зображень, порівняно різні архітектури. Створено програмну реалізацію, перевірено її роботу у різних ситуаціях, виділено основні переваги та недоліки у порівнянні з іншими моделями.

Об’єктом дослідження є процес семантичної сегментації зображення.

Предметом дослідження є структури, моделі та архітектури згорткових нейронних мереж.

Метою роботи є дослідження та порівняльний аналіз впливу архітектур згорткових нейронних мереж на ефективність сегментації об’єктів

Для реалізації поставленої мети необхідно виконати наступні завдання:

– аналіз сучасного стану задачі комп’ютерного зору та сегментації зображень;

– дослідження можливих технологій та підходів для вирішення поставленої задачі;

– розробка програмної реалізація з використанням обраних методів для сегментації об’єктів;

– аналіз отриманих результатів.

Методи дослідження. Для вирішення поставлених задач застосовуються теорії математичного аналізу, теорії статистики, методи та засоби штучного інтелекту.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ЗАДАЧІ КОМП’ЮТЕРНОГО ЗОРУ ТА СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ

1.1 Основні поняття та визначення

Одним із найпотужніших і найпереконливіших видів штучного інтелекту (ШІ) є комп’ютерний зір, який ви майже напевно відчули різними способами, навіть не знаючи.

Комп’ютерний зір — це галузь інформатики, яка зосереджується на відтворенні складних частин системи людського зору та наданні комп’ютерам можливості ідентифікувати й обробляти об’єкти на зображеннях та відео так само, як це роблять люди [1]. Донедавна комп’ютерний зір працював лише в обмежених можливостях.

Завдяки досягненням у галузі штучного інтелекту та інноваціям у глибокому навчанні та нейронних мережах, ця галузь змогла зробити великі стрибки за останні роки та змогла перевершити людей у деяких завданнях, пов’язаних із виявленням та маркуванням об’єктів.

1.1.1 Глибоке навчання

Глибинне навчання – це галузь машинного навчання, що ґрунтується на наборі алгоритмів, які намагаються моделювати високорівневі абстракції в даних, застосовуючи глибинний граф із декількома обробними шарами, що побудовано з кількох лінійних або нелінійних перетворень [1].

Одним із рушійних факторів розвитку комп’ютерного зору є кількість даних, які ми генеруємо сьогодні, які потім використовуються для навчання та покращення комп’ютерного зору. Поряд із величезною кількістю візуальних даних (понад 3 мільярди зображень публікуються в Інтернеті щодня), тепер доступні обчислювальні потужності, необхідні для аналізу даних. Оскільки область комп’ютерного зору розширюється з новим обладнанням та алгоритмами, точність ідентифікації об’єктів зростає. Менш ніж за десятиліття сучасні системи

досягли точності 99 відсотків із 50 відсотків, що робить їх точнішими, ніж люди, у швидкому реагуванні на візуальні дані [2].

Ранні експерименти з комп’ютерним зором почалися в 1950-х роках, і вперше його почали використовувати в комерційних цілях для розрізнення набраного та рукописного тексту в 1970-х роках, сьогодні застосування комп’ютерного зору зросло в геометричній прогресії.

Одне з головних відкритих питань як у нейронауці, так і в машинному навчанні: як саме працює наш мозок і як ми можемо наблизити це за допомогою наших власних алгоритмів? Реальність така, що існує дуже мало робочих і всеосяжних теорій обчислень мозку; тому, незважаючи на те, що нейронні мережі повинні «імітувати роботу мозку», ніхто не впевнений, що це насправді так.

Той самий парадокс справедливий і для комп’ютерного зору — оскільки ми ще не вирішили, як мозок і очі обробляють зображення, важко сказати, наскільки добре алгоритми, які використовуються у виробництві, наближають наші власні внутрішні розумові процеси.

На певному рівні комп’ютерний зір пов’язаний з розпізнаванням образів. Таким чином, один із способів навчити комп’ютер розуміти візуальні дані – це передати йому зображення, тисячі, мільйони зображень, якщо можливо, позначені, а потім піддати їх різним програмним методам або алгоритмам, які дозволяють комп’ютеру вистежувати. візерунки у всіх елементах, які стосуються цих міток.

Так, наприклад, якщо ви «нагодуєте» комп’ютеру мільйони зображень котів, він обробить їх усіх алгоритмам, які дозволять їм аналізувати кольори на фотографії, форми, відстані між фігурами, де об’єкти межують один з одним і так далі, щоб визначити профіль того, що означає «кіт». Коли це буде завершено, комп’ютер (теоретично) зможе використати свій досвід, якщо йому дадуть інші зображення без міток, щоб знайти ті, які є котами.

Нижче наведена проста ілюстрація буфера зображення у відтінках сірого, в якому зберігається наше зображення Авраама Лінкольна (див. рис. 1.1). Ліворуч наше зображення Лінкольна; у центрі пікселі, позначені цифрами від 0 до 255, що

представляють їх яскравість; і праворуч, ці числа самі по собі. Яскравість кожного пікселя представлена одним 8-бітовим числом, діапазон якого становить від 0 (чорний) до 255 (білий).

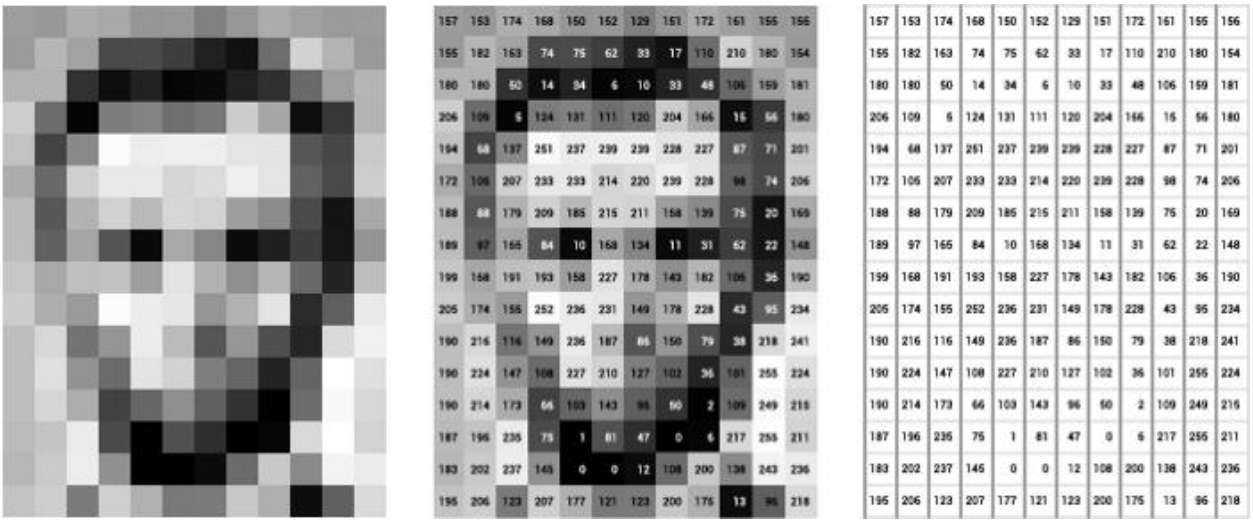


Рис. 1.1. Діаграма піксельних даних

Як що додаємо кольорове зображення, все починає становитися більш складним. Комп’ютери зазвичай зчитують колір як серію з 3 значень — червоного, зеленого та синього (RGB) — у тій самій шкалі 0–255.Тепер кожен піксель фактично має 3 значення, які комп’ютер зберігає на додаток до свого положення. Якби ми розфарбували президента Лінкольна, це призвело б до значень 12 x 16 x 3, або 576 чисел. Тож, для одного зображення потрібно багато пам’яті та багато пікселів для повторення алгоритму.

Глибоке навчання спирається на нейронні мережі, функцію загального призначення, яка може вирішити будь-яку проблему, яку можна представити на прикладах. Більшість сучасних програм комп’ютерного зору, таких як виявлення раку, безпілотні автомобілі та розпізнавання обличчя, використовують глибоке навчання.

1.1.2 Нейронні мережі

Нейронні мережі, також відомі як штучні нейронні мережі або змодельовані нейронні мережі, є підмножиною машинного навчання і лежать в основі алгоритмів глибокого навчання . Їх назва та структура натхненні людським мозком, імітуючи спосіб, яким біологічні нейрони сигналізують один одному.

Штучні нейронні мережі складаються з шарів вузлів, які містять вхідний шар, один або кілька прихованих шарів і вихідний шар. Кожен вузол або штучний нейрон з’єднується з іншим і має відповідну вагу та поріг. Якщо вихід будь-якого окремого вузла перевищує вказане порогове значення, цей вузол активується, надсилаючи дані на наступний рівень мережі. В іншому випадку дані не передаються на наступний рівень мережі.

Нейронні мережі покладаються на навчальні дані, щоб з часом вивчати та покращувати свою точність. Однак, як тільки ці алгоритми навчання будуть точно налаштовані на точність, вони стануть потужними інструментами в інформатиці та штучному інтелекті , що дозволить нам класифікувати та кластерувати дані з високою швидкістю. Завдання розпізнавання мовлення або зображення можуть займати хвилини та години, якщо порівнювати з ручною ідентифікацією експертів. Однією з найвідоміших нейронних мереж є пошуковий алгоритм Google.

Нейронні мережі можна розділити на різні типи, які використовуються для різних цілей. Найпоширеніші типи нейронних мереж: персептрон, нейронні мережі прямого зв’язку, згорткові нейронні мережі, рекурентні нейронні мережі.

Персептрон — це найстаріша нейронна мережа (див. рис. 1.2), створена Френком Розенблатом у 1958 році. Вона має один нейрон і є найпростішою формою нейронної мережі:

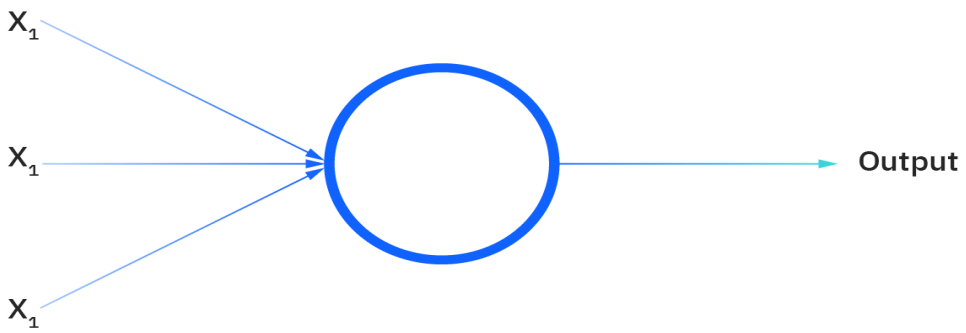


Рис. 1.2. Проста схема персептрона за допомогою ліній і синього кола

Нейронні мережі прямого зв’язку або багатошарові персептрони складаються з вхідного шару, прихованого шару або шарів і вихідного шару. Ці нейронні мережі складаються з сигмовидних нейронів, а не з персептронів, оскільки більшість проблем реального світу є нелінійними. Дані зазвичай подаються в ці моделі для їх навчання, і вони є основою для комп’ютерного зору, обробки природної мови та інших нейронних мереж.

Згорткові нейронні мережі подібні до мереж з прямим зв’язком, але вони зазвичай використовуються для розпізнавання зображень, образів та/або комп’ютерного зору. Ці мережі використовують принципи лінійної алгебри, зокрема множення матриці, для визначення шаблонів у зображенні.

Рекурентні нейронні мережі ідентифікуються за їх петлями зворотного зв'язку. Ці алгоритми навчання в основному використовуються під час використання даних часових рядів для прогнозування майбутніх результатів, таких як прогнози фондового ринку або прогнозування продажів.

Глибоке навчання та нейронні мережі, як правило, використовуються як взаємозамінні в розмові, що може заплутати. Як результат, варто зазначити, що «глибина» в глибокому навчанні означає лише глибину шарів нейронної мережі. Нейронну мережу, яка складається з більш ніж трьох шарів, які включають вхідні та вихідні дані, можна вважати алгоритмом глибокого навчання. Нейронна мережа, яка має лише два або три шари, є просто базовою нейронною мережею.

Глибоке навчання та глибокі нейронні мережі перейшли від концептуальної сфери до практичних застосувань завдяки доступності та прогресу в апаратних і хмарних обчислювальних ресурсах.

1.1.3 Семантична сегментація

Один із ключових розділів комп'ютерного зору – семантична сегментація, процес поділу цифрового зображення на ряд сегментів (об'єктів зображення). Сегментацію описують, як класифікацію зображень на піксельному рівні [2].

Більш конкретно, метою семантичної сегментації зображення є позначення кожного пікселя зображення відповідним класом того, що представляється. Оскільки ми прогнозуємо для кожного пікселя зображення, це завдання зазвичай називають щільним передбаченням.

Важливо зазначити, що ми не відокремлюємо екземпляри одного класу; ми піклуємося лише про категорію кожного пікселя. Іншими словами, якщо у вхідному зображенні є два об’єкти однієї категорії, карта сегментації не розрізняє їх як окремі об’єкти. Існує інший клас моделей, відомий як моделі сегментації екземплярів , які розрізняють окремі об’єкти одного класу. Моделі сегментації корисні для виконання різноманітних завдань, зокрема:

– автономні транспортні засоби. Нам потрібно оснастити автомобілі необхідним сприйняттям, щоб зрозуміти навколишнє середовище, щоб самокеровані автомобілі могли безпечно інтегруватися в наші існуючі дороги;

– обладнання для діагностики медичних зображень може розширити аналіз, проведений радіологами, значно скорочуючи час, необхідний для проведення діагностичних тестів;

– розпізнавання рукописного тексту: семантична сегментація використовується для вилучення слів і рядків із рукописних документів;

– портретний режим Google: є багато випадків використання, коли абсолютно необхідно відокремити передній план від фону. Наприклад, у

портретному режимі Google ми бачимо, що фон розмитий, а передній план залишається незмінним, щоб створити прохолодний ефект;

– віртуальний макіяж: нанесення віртуальної помади тепер можливо за допомогою сегментації зображення;

– віртуальна примірка: віртуальна примірка одягу – цікава функція, яка була доступна в магазинах за допомогою спеціалізованого обладнання, яке створює 3D-модель. Але за допомогою глибокого навчання та сегментації зображень те саме можна отримати, використовуючи лише двовимірне зображення;

– візуальний пошук зображень: ідея сегментації одягу також використовується в алгоритмах пошуку зображень в електронній комерції. Наприклад, Pinterest/Amazon дозволяє завантажувати будь-яке зображення та отримувати схожі товари, виконуючи пошук зображень на основі сегментації частини тканини.

Коли ми накладаємо один канал нашої цілі (або передбачення), ми називаємо це маскою , яка висвітлює області зображення, де присутній певний клас.

У звичайній старій задачі класифікації зображень ми просто зацікавлені в отриманні міток усіх об’єктів, які присутні на зображенні. У виявленні об’єктів ми йдемо ще на один крок і намагаємося разом з тим, що всі об’єкти присутні на зображенні, місце розташування об’єктів за допомогою обмежувальних рамок. Сегментація зображення виводить його на новий рівень, намагаючись точно з’ясувати точні межі об’єктів на зображенні.

Ми знаємо, що зображення — це не що інше, як набір пікселів. Сегментація зображення — це процес класифікації кожного пікселя зображення, що належить до певного класу, і, отже, його можна розглядати як проблему класифікації пікселя. Існує два типи методів сегментації:

– семантична сегментація: семантична сегментація — це процес класифікації кожного пікселя, що належить до певної мітки. Він не відрізняється в

різних екземплярах одного і того ж об’єкта. Наприклад, якщо на зображенні є 2 кішки, семантична сегментація надає однакову мітку для всіх пікселів обох кішок;

– сегментація екземплярів: сегментація екземплярів відрізняється від семантичної в тому сенсі, що вона надає унікальну мітку кожному екземпляру конкретного об’єкта в зображенні. Як видно на зображенні вище, всім 3 собакам присвоєні різні кольори, тобто різні мітки. При семантичній сегментації всім їм було б призначено однаковий колір.

1.2 Останні дослідження та публікації

До появи глибокого навчання для вирішення проблеми сегментації зображень використовувалися класичні методи машинного навчання, такі як SVM, Random Forest, K-means Clustering. Але, як і в більшості випадків, пов’язаних із зображенням, глибоке навчання працювало всебічно краще, ніж існуючі методики, і тепер стало нормою при роботі із семантичною сегментацією.

1.2.1 Повністю згортка мережа (CNN)

Загальна архітектура CNN складається з кількох згорткових і об’єднаних шарів, за якими слідують кілька повністю пов’язаних шарів на кінці (див. рис. 1.3). У статті Fully Convolutional Network, опублікованій у 2014 році, стверджується, що остаточний повністю пов’язаний шар можна розглядати як згортку 1x1, яка охоплює весь регіон.

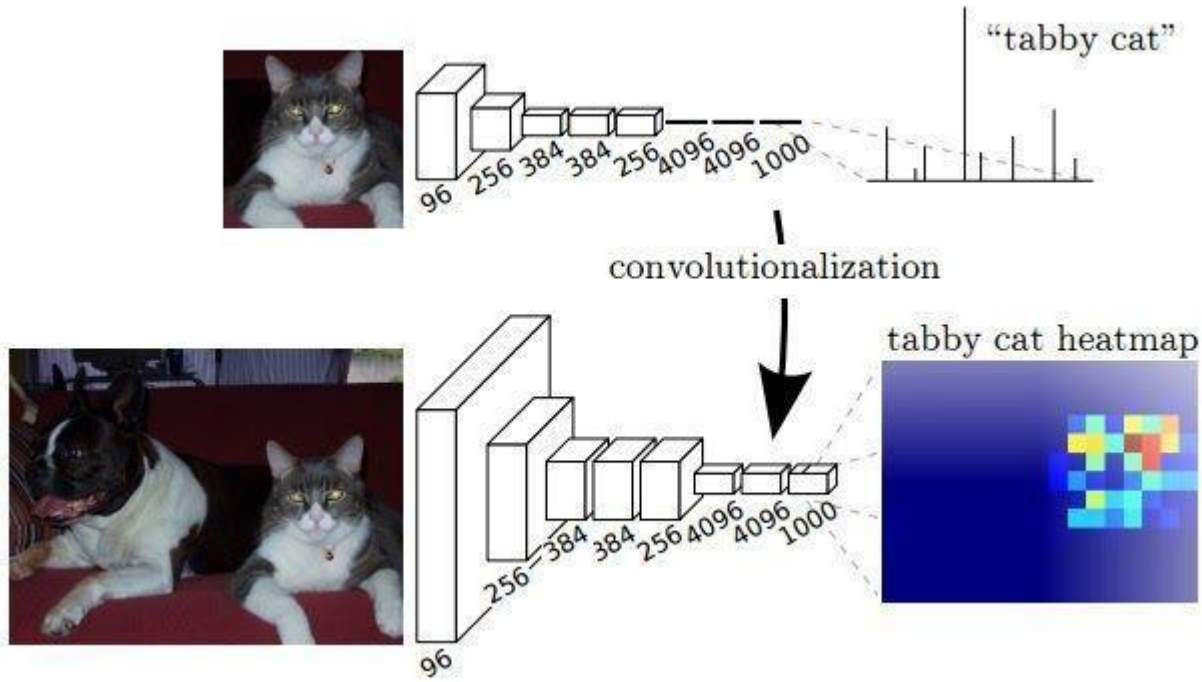


Рис. 1.3. Повністю згортка мережа

Таким чином, кінцеві щільні шари можна замінити шаром згортки, досягаючи того ж результату. Але тепер перевага цього полягає в тому, що розмір введення більше не потрібно фіксувати. При залученні щільних шарів розмір вхідних даних обмежений, і, отже, коли потрібно надати вхід іншого розміру, його потрібно змінити. Але замінивши щільний шар на згортку, це обмеження не існує.

Крім того, якщо в якості вхідних даних надається зображення більшого розміру, отриманий вихід буде картою об’єктів, а не просто виводом класу, як для зображення нормального розміру вхідного формату. Також спостережувана поведінка остаточної карти об’єктів представляє теплову карту необхідного класу, тобто положення об’єкта виділено на карті об’єктів. Оскільки вихідною картою ознак є теплова карта необхідного об’єкта, це дійсна інформація для нашого випадку використання сегментації.

Оскільки карта ознак, отримана на вихідному шарі, є нижчою вибіркою через набір виконаних згорток, ми хотіли б підвищити її вибірку за допомогою техніки інтерполяції. Дволінійна підвищена вибірка працює, але в статті

пропонується використовувати вивчену підвищувальну вибірку з деконволюцією, яка може навіть вивчати нелінійну підвищувальну вибірку.

Нижня частина мережі з дискретизацією називається кодером, а частина дискретизації вгору — декодером. Це шаблон, який ми побачимо в багатьох архітектурах, тобто зменшуючи розмір за допомогою кодера, а потім підвищуючи вибірку за допомогою декодера. В ідеальному світі ми б не хотіли зменшувати вибірку за допомогою об’єднання і зберігати однаковий розмір на всьому протязі, але це призвело б до величезної кількості параметрів і було б нездійсненним з точки зору обчислень (див. рис. 1.4).

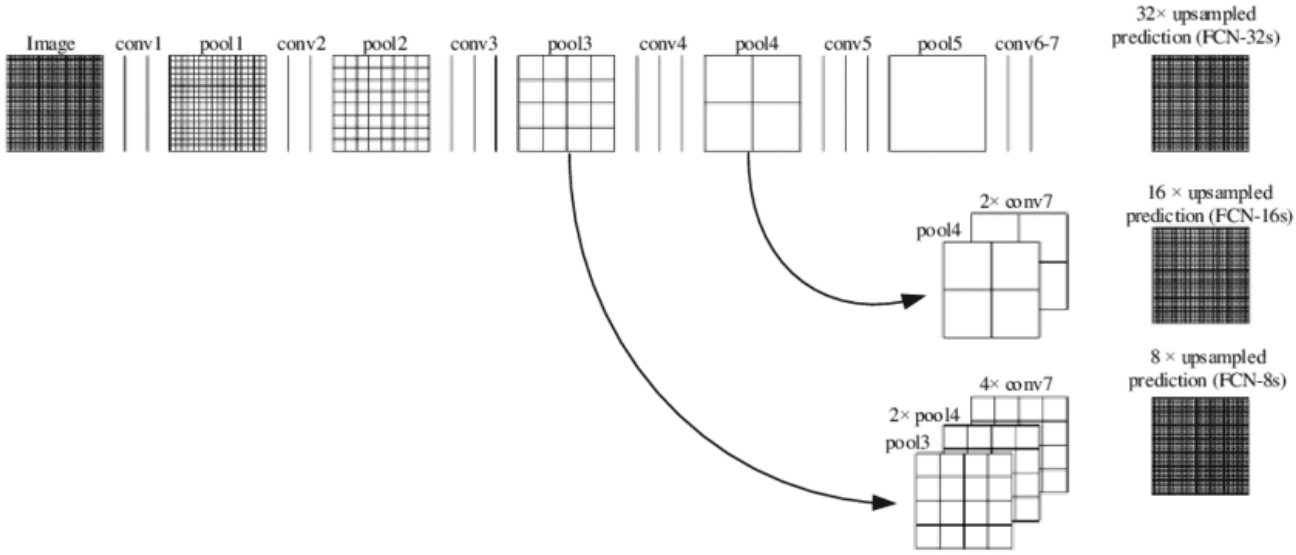


Рис. 1.4. Приклад кодеру нейронної мережі

Хоча отримані результати були гідними, спостережуваний результат був грубим і нерівним. Причиною цього є втрата інформації на кінцевому графічному шарі через зменшення дискретизації в 32 рази за допомогою шарів згортки. Тепер для мережі стає дуже важко виконати 32-кратне підвищення дискретизації, використовуючи цю невелику інформацію. Ця архітектура називається FCN-32.

Для вирішення цієї проблеми в роботі запропоновано 2 інші архітектури FCN-16, FCN-8. У FCN-16 інформація з попереднього рівня об’єднання використовується разом з остаточною картою об’єктів, і, отже, тепер завдання мережі полягає в тому, щоб навчитися дискретизації в 16 разів, що краще

порівняно з FCN-32. FCN-8 намагається зробити його ще кращим, включивши інформацію з ще одного попереднього рівня об’єднання.

1.2.2 U-net

U-net будується на основі повністю згорткової мережі зверху. Він був побудований для медичних цілей, щоб знайти пухлини в легенях або мозку. Він також складається з кодера, який понижує вибірку вхідного зображення на карту об’єктів, і декодера, який збільшує вибірку карти об’єктів для введення розміру зображення за допомогою вивчених шарів деконволюції (див. рис. 1.5).

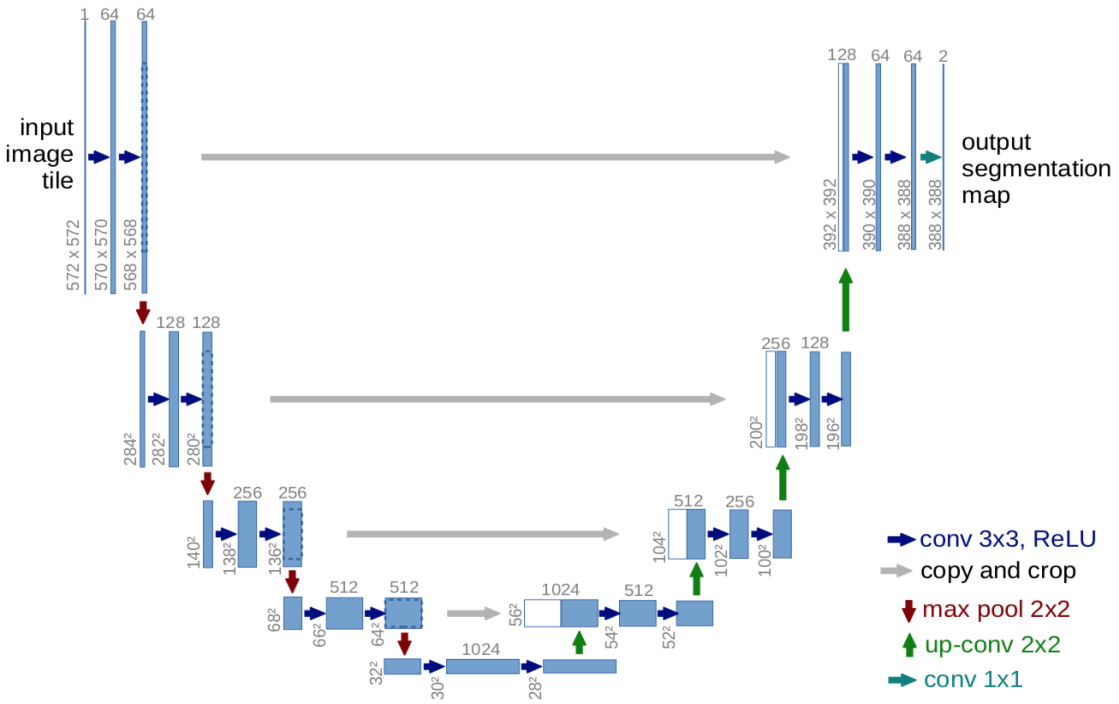


Рис. 1.5. Схема архітектури U-net

Основним внеском архітектури U-Net є ярлики. Вище в FCN ми бачили, що оскільки ми зменшили вибірку зображення як частину кодера, ми втратили багато інформації, яку не можна легко відновити в частині кодера. FCN намагається вирішити цю проблему, беручи інформацію з шарів об’єднання перед остаточним шаром.

U-Net пропонує новий підхід до вирішення цієї проблеми втрати інформації. Він пропонує надсилати інформацію до кожного рівня дискретизації

вгору в декодері з відповідного нижнього рівня дискретизації в кодері, як можна побачити на малюнку вище, таким чином фіксуючи більш точну інформацію, а також зберігаючи низький рівень обчислень. Оскільки шари на початку кодера мали б більше інформації, вони підсилили б операцію декодера з підвищенням вибірки, надаючи дрібні деталі, що відповідають вхідним зображенням, таким чином значно покращуючи результати. У статті також запропоновано використання нової функції втрат, яку ми обговоримо нижче.

1.2.3 DeepLab

Deeplab з групи дослідників з Google запропонував безліч методів для покращення існуючих результатів і отримання кращого результату при менших обчислювальних витратах. 3 основні вдосконалення, запропоновані в рамках дослідження:

– Atrous згортки;

– Atrous Spatial Pyramidal Pooling;

– використання умовних випадкових полів для покращення кінцевого результату.

Однією з основних проблем підходу FCN є надмірне скорочення розмірів через послідовні операції об’єднання. Через серію об’єднання вхідне зображення зменшується в 32х, а вибірка знову підвищується, щоб отримати результат сегментації. Зменшення дискретизації в 32 рази призводить до втрати інформації, що дуже важливо для отримання точного результату в задачі сегментації. Крім того, деконволюція до збільшення вибірки в 32х є дорогою операцією для обчислень і пам’яті, оскільки існують додаткові параметри, які беруть участь у формуванні вивченої вибірки.

У статті пропонується використання згортки Атроуса або згортки отворів або розширеної згортки, що допомагає отримати розуміння великого контексту з використанням тієї ж кількості параметрів.

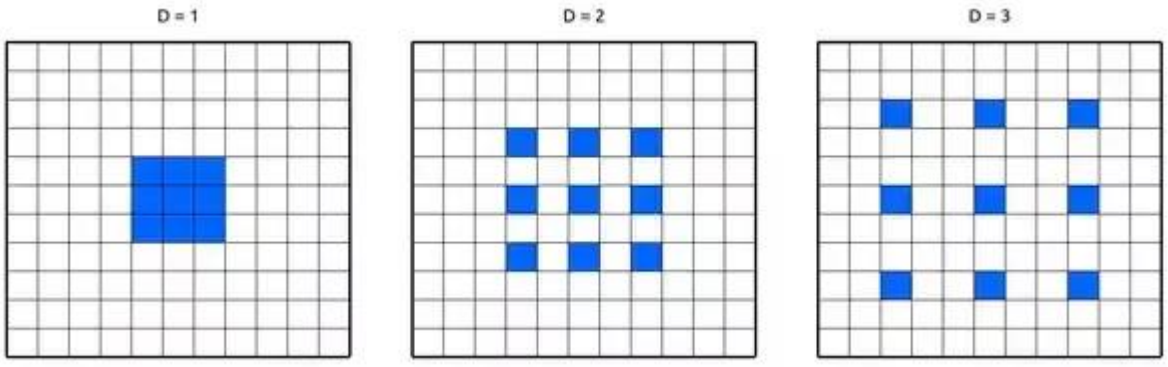


Рис. 1.6. Ілюстрація розширеної згортки

Розширена згортка працює шляхом збільшення розміру фільтра шляхом додавання нулів (так звані отвори), щоб заповнити проміжок між параметрами (див. рис. 1.6). Кількість дірок/нулів, заповнених між параметрами фільтра, називається швидкістю розширення терміну. Коли швидкість дорівнює 1, це не що інше, як звичайна згортка. Коли швидкість дорівнює 2, один нуль вставляється між кожним іншим параметром, завдяки чому фільтр виглядає як згортка 5x5. Тепер він має можливість отримати контекст згортки 5x5, маючи параметри згортки 3x3. Аналогічно для швидкості 3 рецептивне поле переходить до 7x7.

У Deelab останні шари об’єднання замінюються, щоб мати крок 1 замість 2, таким чином знижуючи частоту дискретизації лише 8x. Потім застосовується серія атрузійних звивин, щоб охопити більший контекст. Для навчання вихідна позначена маска зменшується в 8 разів для порівняння кожного пікселя. Для висновку, дволінійна підвищена вибірка використовується для отримання вихідних даних того ж розміру, що дає досить гідні результати при менших витратах на обчислення/пам’ять, оскільки дволінійна підвищена вибірка не потребує жодних параметрів, на відміну від деконволюційної вибірки.

1.2.4 SPP

Просторовий пірамідальний об’єднання – це концепція, введена в SPPNet для отримання багатомасштабної інформації з карти об’єктів. Перед введенням SPP надаються вхідні зображення з різною роздільною здатністю, а обчислені карти функцій використовуються разом для отримання багатомасштабної

інформації, але це вимагає більше обчислень і часу. Завдяки Spatial Pyramidal Pooling багатомасштабну інформацію можна отримати за допомогою одного вхідного зображення.

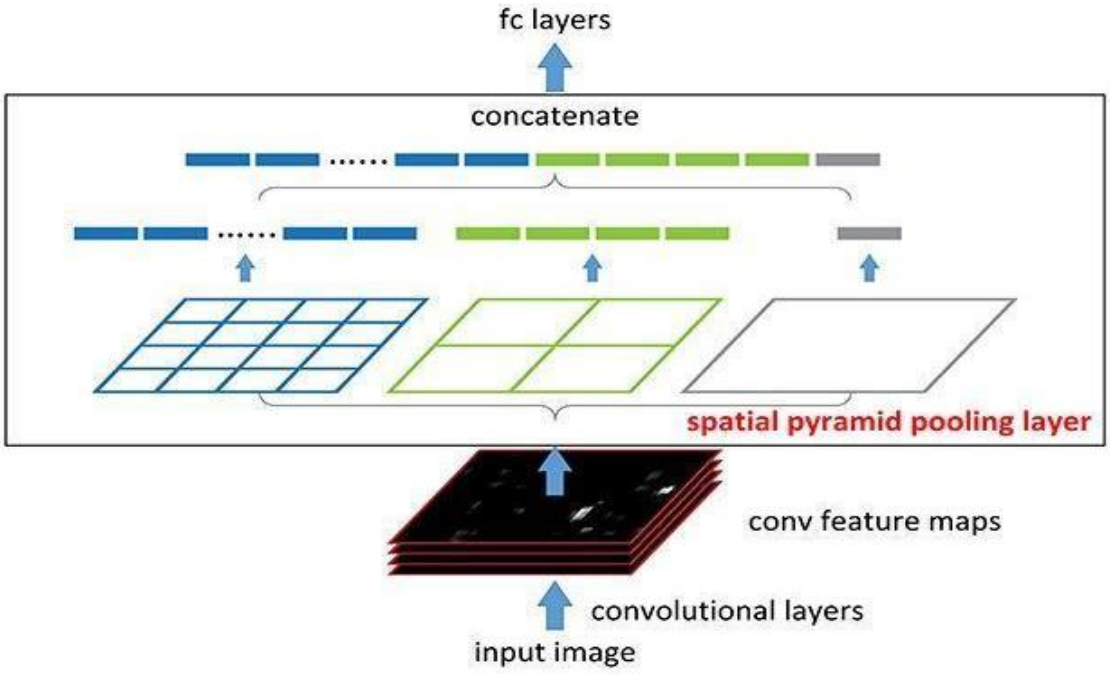


Рис. 1.7. Схема архітектури SPPNet

За допомогою модуля SPP мережа виробляє 3 виходи розмірів 1x1 (тобто GAP), 2x2 і 4x4 (див. рис. 1.7). Ці значення об’єднуються шляхом перетворення в 1d вектор, таким чином фіксуючи інформацію в різних масштабах. Ще однією перевагою використання SPP є можливість надання вхідних зображень будь-якого розміру.

1.2.5 ASPP

ASPP бере концепцію злиття інформації з різних масштабів і застосовує її до звивин Атроуса. Вхід згортається з різною швидкістю розширення, а виходи з них зливаються разом.

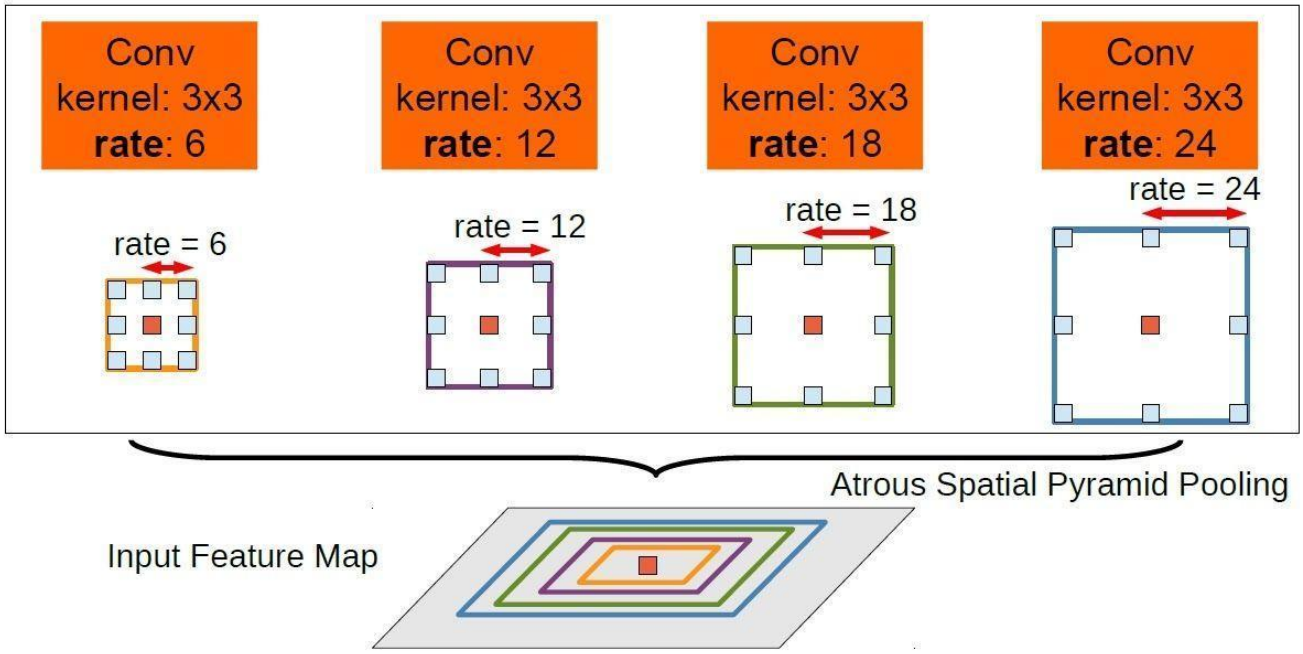


Рис. 1.8. Схема архітектури ASPP

Як видно (див. рис. 1.8), вхід згорнуто за допомогою фільтрів 3x3 зі швидкостями розширення 6, 12, 18 і 24, а виходи об’єднані разом, оскільки вони однакового розміру. Вихід згортки 1x1 також додається до виходу з плавким перемиканням. Щоб також надати глобальну інформацію, вихід GAP також додається до вищевказаного після збільшення вибірки. Об’єднаний вихід 3x3 різноманітних розширених виходів, 1x1 і вихід GAP пропускається через згортку 1x1, щоб отримати необхідну кількість каналів.

Оскільки необхідне зображення для сегментації може бути будь-якого розміру на вході, багатомасштабна інформація від ASPP допомагає покращити результати.

1.2.6 CRF

Підвищення продуктивності за допомогою CRF. Пул — це операція, яка допомагає зменшити кількість параметрів у нейронній мережі, але вона також приносить властивість інваріантності разом із нею. Інваріантність — це якість нейронної мережі, на яку не впливають незначні трансляції у вхідних даних.

Через цю властивість, отриману при об’єднанні, вихід сегментації, отриманий нейронною мережею, є грубим, а межі не визначені конкретно.

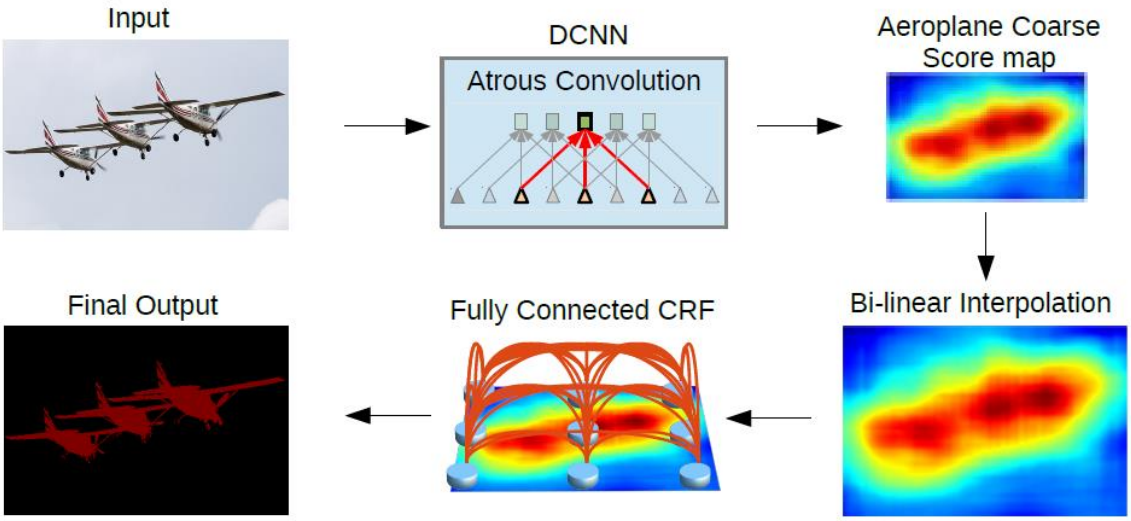
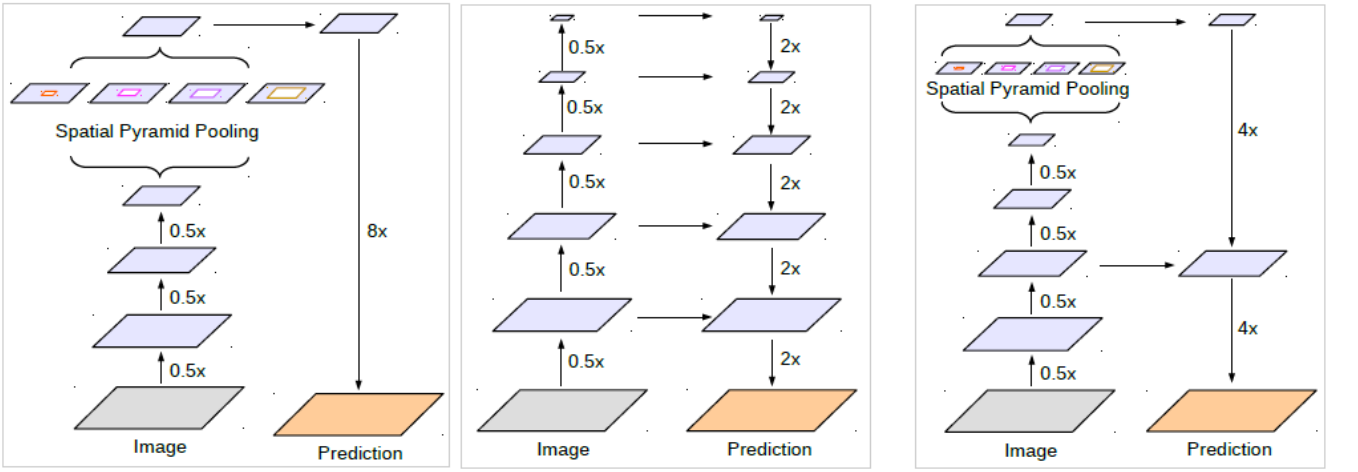


Рис. 1.9. Схема CRF

Для вирішення цього питання пропонується використовувати графічну модель CRF (див. рис. 1.9). Умовне випадкове поле виконує етап постобробки та намагається покращити отримані результати, щоб визначити межі форми. Він працює, класифікуючи піксель на основі не тільки його мітки, але й на основі інших міток пікселів. Як видно з наведеного вище малюнка, груба межа, створена нейронною мережею, стає більш витонченою після проходження через CRF.

1.2.7 Deelab-v3

Deelab-v3 ввів пакетну нормалізацію та запропонував швидкість розширення, помножену на (1,2,4) всередині кожного шару в блоці Resnet. Також у рамках цієї статті було запропоновано додати функції рівня зображення до модуля ASPP, який обговорювався вище в обговоренні ASPP (див. рис. 1.10).



(a) Spatial Pyramid Pooling (b) Encoder-Decoder (c) Encoder-Decoder with Atrous Conv

Рис. 1.10. Комбінація кодер-декодер з SPP

Deelab-v3+ запропонував мати декодер замість звичайної дволінійної дискретизації 16x. Декодер отримує підказку від декодера, який використовується в таких архітектурах, як U-Net, які беруть інформацію з шарів кодера для покращення результатів. Вихід кодера збільшується в 4 рази за допомогою дволінійної дискретизації і об’єднується з функціями кодера, який знову дискретується в 4 рази після виконання згортки 3x3. Цей підхід дає кращі результати, ніж пряма вибірка в 16 разів. Також пропонується використовувати модифіковану архітектуру Xception замість Resnet як частину кодера, а згортки з роздільними по глибині тепер використовуються поверх згорток Atrous, щоб зменшити кількість обчислень.

1.2.8 GCN

Семантична сегментація передбачає одночасне виконання двох завдань:

– класифікація;

– локалізація.

Мережі класифікації створені для того, щоб бути інваріантними щодо трансляції та обертання, таким чином не надаючи значення інформації про місцезнаходження, тоді як локалізація передбачає отримання точних деталей щодо розташування (див. рис. 1.11).

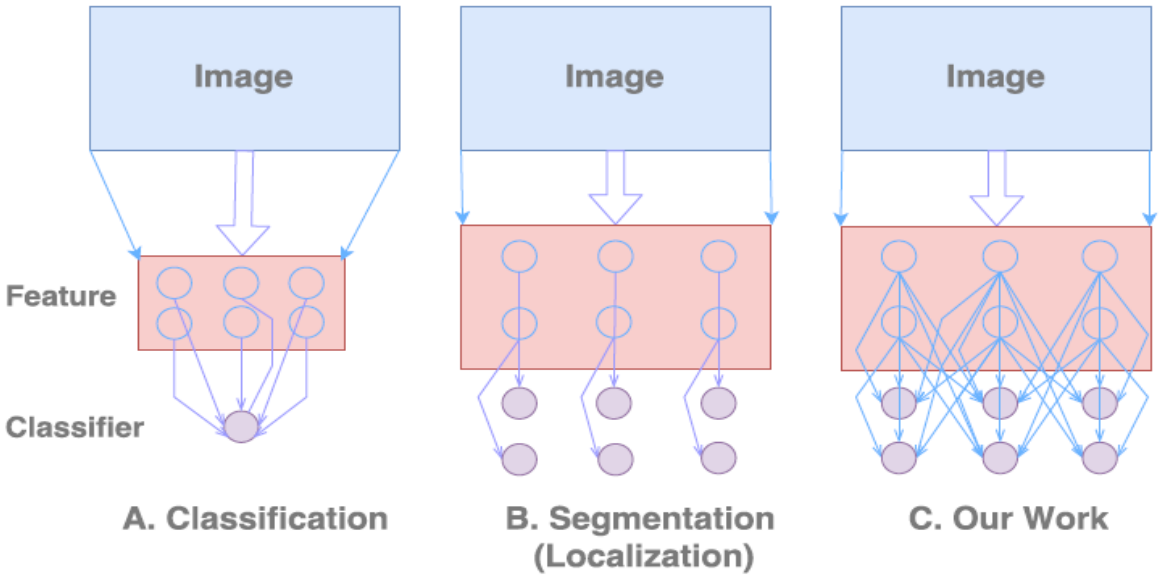


Рис. 1.11. Схема архітектури GCN

Таким чином, за своєю суттю ці два завдання є суперечливими. Більшість алгоритмів сегментації надають більше значення локалізації, тобто другому на малюнку вище, і таким чином втрачають з поля зору глобальний контекст. У цій роботі автор пропонує спосіб надати важливості і завданням класифікації, водночас не втрачаючи інформації про локалізацію.

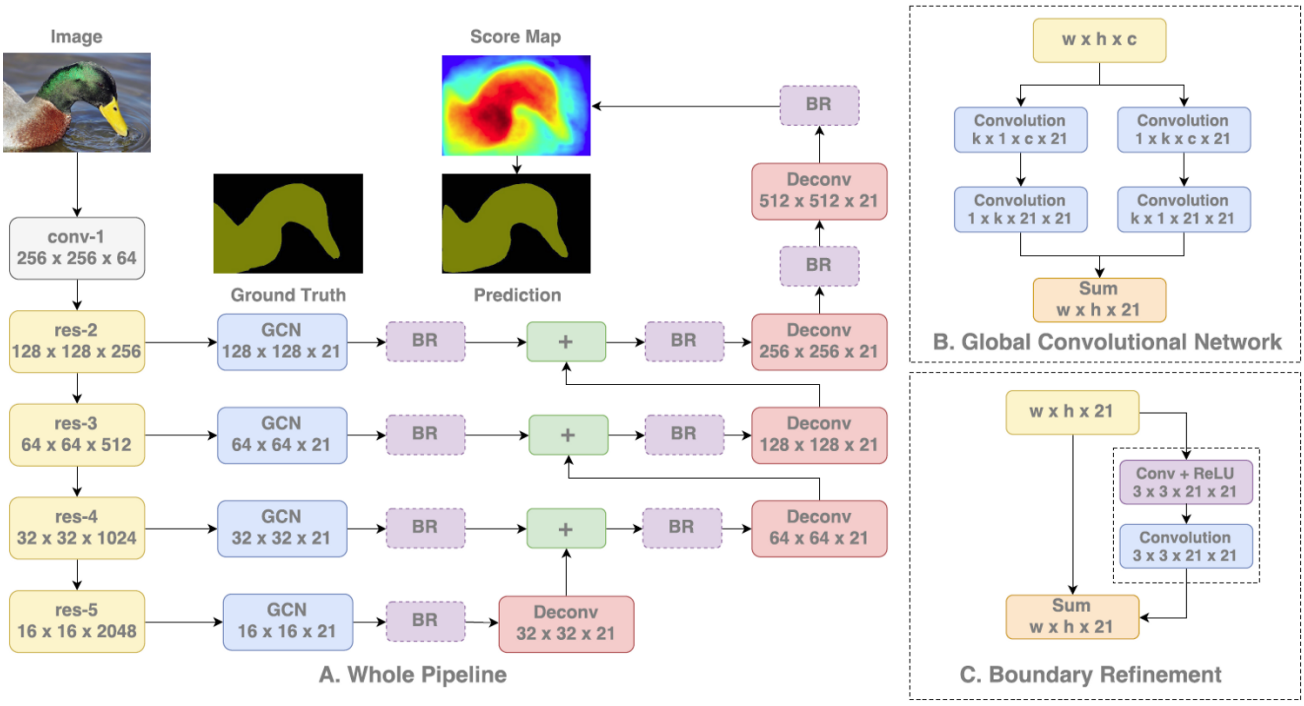


Рис. 1.12. Обробка Global Convolution Network

Автор пропонує досягти цього, використовуючи великі ядра як частину мережі, що забезпечує щільні з'єднання, а отже, і більше інформації. Це досягається за допомогою блоку GCN (див. рис. 1.12). Блок GCN можна розглядати як фільтр згортки akxk, де k може бути числом, більшим за 3. Щоб зменшити кількість параметрів, фільтр akxk додатково розбивається на блоки 1 xk і kx 1, kx1 і 1xk, які потім підсумовуються. Таким чином, збільшення значення k охоплює більший контекст.

Крім того, автор пропонує блок Boundary Refinement, подібний до залишкового блоку, який можна побачити в Resnet, що складається з ярлика та залишкового з’єднання, які підсумовуються для отримання результату. Помічено, що наявність блоку Boundary Refinement призвела до покращення результатів на межі сегментації. Результати показали, що блок GCN покращив точність класифікації пікселів ближче до центру об’єкта, що вказує на покращення, викликане захопленням контексту дальнього діапазону, тоді як блок Boundary Refinement допоміг покращити точність пікселів ближче до кордону.

1.2.9 KSAC

Перегляньте більше ніж один раз – використання KSAC для семантичної сегментації.

Сімейство Deelab використовує ASPP, щоб мати кілька сприйнятливих полів, які збирають інформацію з різними показниками атрозної згортки. Незважаючи на те, що ASPP був значно корисним для покращення сегментації результатів, існують деякі притаманні проблеми, викликані архітектурою. У ASPP немає інформації, спільної для різних паралельних шарів, що впливає на потужність узагальнення ядер у кожному шарі. Крім того, оскільки кожен шар обслуговує різні набори навчальних вибірок (менші об’єкти до меншої швидкості атрімації, а більші об’єкти до більшої швидкості токсичності), кількість даних для кожного паралельного шару буде меншою, що вплине на загальну узагальненість. Також кількість параметрів у мережі збільшується лінійно з кількістю параметрів і, таким чином, може призвести до переобладнання.

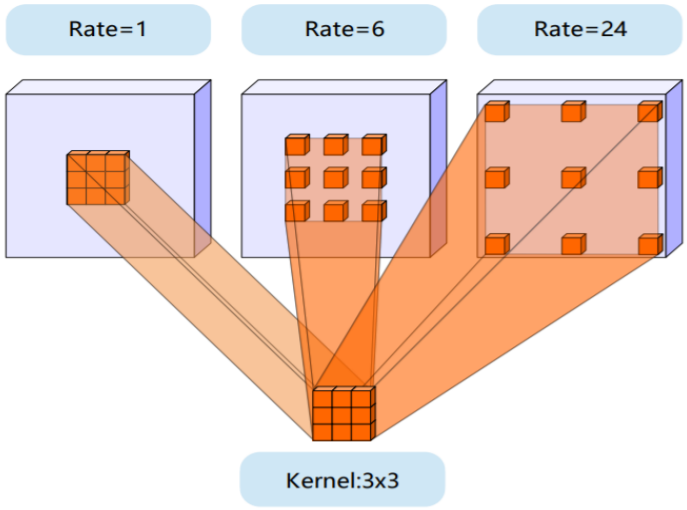


Рис. 1.13. Ілюстрація розширеної згортки

Щоб вирішити всі ці проблеми, автор пропонує нову мережеву структуру під назвою Kernel-Sharing Atrous Convolution (KSAC). Як видно на малюнку (див. рис. 1.13), замість того, щоб мати різне ядро для кожного паралельного рівня ASPP, єдине ядро використовується спільно, таким чином покращуючи здатність

мережі до узагальнення. Використовуючи KSAC замість ASPP, зберігається 62% параметрів при використанні коефіцієнтів дилатації 6,12 і 18.

Ще одна перевага використання структури KSAC полягає в тому, що кількість параметрів не залежить від кількості використовуваних швидкостей розширення. Таким чином, ми можемо додати якомога більше ставок, не збільшуючи розмір моделі. ASPP дає найкращі результати з показниками 6,12,18, але точність знижується з 6,12,18,24, що вказує на можливе переобладнання. Але точність KSAC все ще значно покращується, що свідчить про розширені можливості узагальнення.

Цю техніку спільного використання ядра також можна розглядати як розширення простору функцій, оскільки одне й те саме ядро застосовується на кількох швидкостях. Подібно до того, як збільшення введення дає кращі результати, розширення функцій, що виконується в мережі, повинно допомогти покращити можливості представлення мережі.

1.3 Огляд існуючих аналогів

1.3.1 SegNet

SegNet складається з стеку кодерів, за якими слідує відповідний стек декодерів, який подається на рівень класифікації soft-max. Декодери допомагають зіставляти карти об’єктів низької роздільної здатності на виході стека кодувальників у карти об’єктів повного розміру вхідного зображення. Це усуває важливий недолік останніх підходів глибокого навчання, в яких використовуються мережі, розроблені для категоризації об’єктів для піксельного маркування. У цих методів відсутній механізм для відображення карт об’єктів глибокого шару на вхідні розміри. Вони вдаються до спеціальних методів для підвищення вибірки функцій, напршляхом реплікації. Це призводить до шумних передбачень, а також обмежує кількість шарів об’єднання, щоб уникнути занадто великої дискретизації і, таким чином, зменшує просторовий контекст. SegNet

долає ці проблеми, навчаючись зіставляти вихідні дані кодера з мітками пікселів зображення.

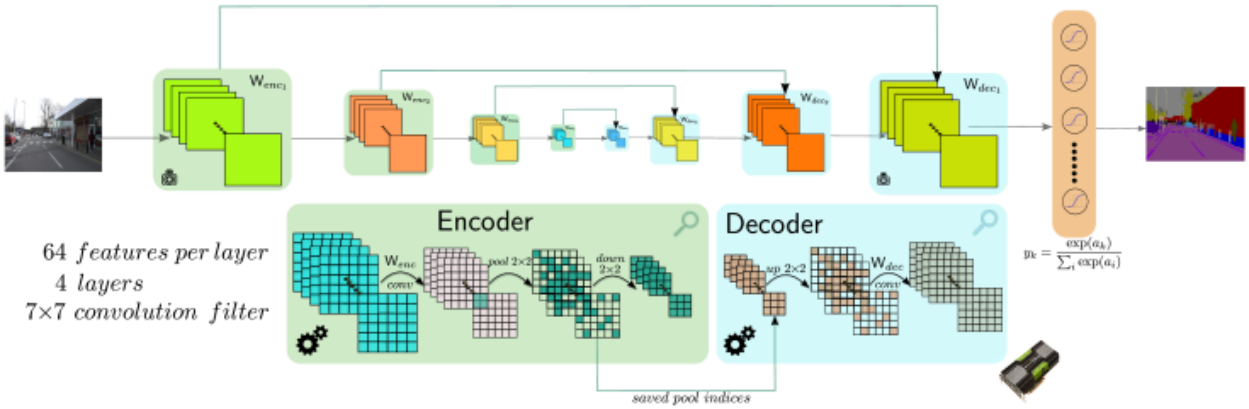


Рис. 1.14. SegNet 4 рівня

Чотирирівнева архітектура SegNet, використана в експериментах, (див. рис. 1.14). Кожен кодер виконує щільні згортки, нелінійність ReLU, максимальний пул без перекриття з 2×2 вікно і, нарешті, знижуючи вибірку. Кожен декодер підвищує дискретизацію свого входу за допомогою збережених об’єднаних індексів і згортає його за допомогою банку фільтрів, який можна навчати. У декодері не використовується нелінійність ReLU, на відміну від мережі деконволюції. Це полегшує оптимізацію фільтрів у кожній парі. Фільтри кодера і декодера також роз’єднані, щоб забезпечити додаткові ступені свободи для мінімізації цілі. Останній шар — це soft-max класифікатор (без терміну зміщення), який класифікує кожен піксель незалежно. Результатом soft-max є зображення каналу K, де K — кількість класів.

SegNet використовує «плоску» архітектуру, тобто кількість функцій у кожному шарі залишається незмінною (64 у цьому випадку), але з повним підключенням. Цей вибір мотивований двома причинами. По-перше, він уникає вибуху параметрів, на відміну від розширюваної мережі глибокого кодування з повнофункціональним підключенням (те саме для декодера). По-друге, час навчання залишається незмінним (у експериментах він трохи зменшується) для кожної додаткової/глибшої пари кодер-декодер, оскільки роздільна здатність

карти об’єктів менша, що робить згортки швидшими. Зважаємо, що декодер, що відповідає першому кодеру (найближче до вхідного зображення), створює багатоканальну карту ознак, хоча вхід кодера має 3 або 4 канали (RGB або RGBD) (див. рис. 1.14). Це високорозмірне представлення ознак подається в soft-max класифікатор. Це на відміну від інших декодерів, які створюють карти ознак того ж розміру, що і вхідні дані кодера. Фіксоване вікно об’єднання 2×2 з кроком без перекриття 2 використовується пікселів. Цей невеликий розмір зберігає тонкі структури в сцені. Крім того, постійний розмір ядра 7×7 над усіма шарами було вибрано, щоб забезпечити широкий контекст для плавного маркування, тобто піксель у карті особливостей найглибшого шару можна простежити назад до контекстного вікна у вхідному зображенні 106×106 пікселів. Компроміс тут полягає між розміром вікна контексту та збереженням тонких структур. Менші ядра зменшують контекст, а більші потенційно руйнують тонкі структури.

Вхідним сигналом для SegNet може бути будь-яке довільне багатоканальне зображення або карта(и), наприклад , RGB, RGBD, карта нормалей, глибина тощо. Виконуємо локальну нормалізацію контрасту (LCN) як крок попередньої обробки вхідних даних. Переваги цього кроку багато:

– для виправлення нерівномірного освітлення сцени, таким чином зменшуючи динамічний діапазон (збільшує контраст у затінених частинах;

– виділення країв, що спонукає мережу до вивчення форми категорії;

– покращує конвергенцію, оскільки декорує вхідні розміри [23]. LCN виконується незалежно для кожної модальності, тобто, RGB – це контраст, нормалізований як триканальний вхід, а глибина – як один канал для входів RGBD.

Це дозволяє уникнути виділення країв псевдо глибини через краї RGB і навпаки.

1.3.2 DeepLabV3+

DeepLabV3+ – це метод семантичної сегментації, який дає дуже багатообіцяючі результи та має на даний момент найкращий рейтинг серед

кількох методів, включаючи SegNet, PSP і FCN. DeelabV3 + використовує механізм Atrous Spatial Pyramid Pooling (ASPP), який використовує багато масштабну контекстну інформацію для покращення сегментації. Atrous (що означає діри) згортка має переваги перед стандартною згорткою, надаючи відповіді на всі зображення позиції, а кількість параметрів фільтра та кількість операцій залишаються незмінними. DeepLabV3 + має структуру мережі кодувальник-декодер. Кодерна частина його складається з набору процесів, які зменшують карти ознак і захоплюють семантичну інформацію та декодер частина з нього відновлює просторову інформацію і призводить до більш чіткої сегментації.

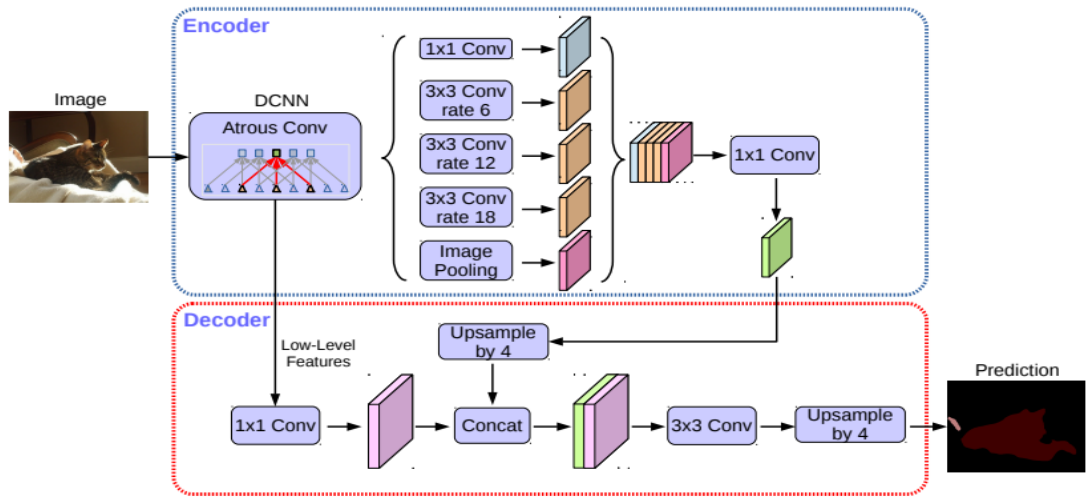


Рис. 1.15. Схема архітектури DeepLabV3+

Блок-схему DeepLabV3 + можна побачити на малюнку (див. рис. 1.15).

DeepLabv3+ — це останній алгоритм семантичної сегментації , запущений Google у 2018 році. Алгоритм розроблено на основі DeepLabv1-3. По-перше, Deelabv1 пропонує операцію згортки з дірками. У разі зменшення дискретизації, сприйнятливе поле мережі розширюється, і модель отримує більш щільні карти ознак. Щоб додатково покращити здатність сегментації мережі , мережа згодом приймає повністю підключену операцію CRF, що ще більше підвищує точність цільової сегментації , але Deelabv1 має погану здатність обробки для багатомасштабних об’єктів сегментації. Щоб вирішити цю проблему, Deelabv2

пропонує структуру ASPP (Atous Spatial Pyramid Pooling) на основі версії V1, яка використовує операцію розширеної згортки різних частот вибірки для паралельної вибірки вхідної карти об’єктів, тобто об’єкта. карта багатомасштабована для отримання інформації про контекст зображення. Deelabv2 також використовує повністю підключену операцію CRF у подальшій обробці для отримання більш точного сегментування зображень. Коли швидкість розширення ядра згортки 3×3 у структурі ASPP продовжує зростати, згортка 3×3 виродиться в згортку 1×1. Щоб компенсувати цей дефект та інтегрувати інформацію про глобальний контекст, Deelabv3 змінює структуру ASPP до трьох 3 Швидкість розширення операцій згортки ×3 становить {6, 12, 18} і операція глобального середнього об'єднання відповідно. Оскільки ASPP інтегрує функції на рівні зображення та містить інформацію про цільове розташування , версія V3 видаляє повністю підключені Операція CRF. Мережа Deelabv3+ додає структуру кодування-декодування на основі серії алгоритмів Deelabv3, що робить її одним із найкращих алгоритмів семантичної сегментації на цьому етапі. Мережа DeepLabv3+ використовується у багатьох загальнодоступних наборах даних., такі як PASCAL VOC2012, CitySpaces тощо, досягли ідеальних результатів. Видно, що мережа складається з кодера і декодера, а кодер розділений на розширену по глибині згортку нейронні мережі. І на рівні ASPP, декодер об’єднує низькорівневі функції та виконує відновлення карти характеристик, що детально описано нижче.

По-перше, повністю згортка глибокої нейронної мережі використовує шар згортки для вилучення особливостей об’єктів або сцен із зображення. Щоб зменшити складність обчислень глибокої мережі, шар об’єднання використовується для зменшення розмірності карти ознак після згортка, тобто процес зниження дискретизації, але багаторазові операції зниження дискретизації призведуть до занадто великої втрати цільової інформації про межі, що не підходить для завдань семантичної сегментації . DeepLabv3+ додає Atrous Convolution до мережі глибокого вилучення ознак і збільшує сприйнятливе поле мережі, забезпечуючи скорочення операцій зниження дискретизації та відсутність

збільшення параметрів мережі, так що карта характеристик не втрачає інформацію про особливості межі цільової сегментації, наскільки це можливо, покращуючи таким чином ефект розщеплення.

По-друге, для завдань сегментації з кількома об’єктами різні об’єкти на зображенні мають різний масштаб, а для сегментації кількох об’єктів використовується один і той же шар ознак , що не може гарантувати точність сегментації . Мережа DeepLabv3+ запозичує операцію Spatial Pyramid Pooling (SSP) в SSP-Net і покращує її в ASPP, щоб досягти можливості сегментації багатомасштабних об’єктів . ASPP виконує згортку 1×1 на вхідній карті об’єктів, згортку 3×3 зі швидкістю розширення 6, 12 і 18 і операцію глобального середнього об’єднання, потім об’єднує карти ознак і виконує згортку 1×1 для стиснення кількості каналів. Зрештою, ASPP може завершити вилучення та розрізнення інформації про цільові характеристики різного масштабу та добре реалізувати сегментацію багатомасштабних цілей

Нарешті, щоб повністю витягти інформацію про функції високого рівня цільового об’єкта в зображенні, мережа DeepLabv3+ виконує необхідну операцію зниження дискретизації на вхідному зображенні. Щоб компенсувати втрачену інформацію про межі під час операції зменшення дискретизації, DeepLabv3+ приймає структуру кодування-декодування. Низькорівневі ознаки об'єднуються для відновлення інформації про межі цільової частини, а метод лінійної інтерполяції , що, нарешті, покращує точність сегментації мереж.

Механізм самоуваги використовувався в різних областях глибокого навчання останніми роками, і він має хороші показники в обробці зображень, розпізнаванні мовлення та обробці природної мови . Він може моделювати довготривалу залежність в обробці зображень і встановлювати зв’язок між двома пікселями на певній відстані в зображенні. Наприклад, при запровадженні механізму самоуваги в генерацію зображень та оцінку мереж GAN і виявляється, що використання механізму уваги в функціях середнього або високого рівня значно спричиняє ефект створення зображення в мережі GAN на основі самосвідомості Механізм уваги пропонує нелокальні операції в просторово-

часовому вимірі і досягає добрих результатів на зображеннях і відео; запроваджується механізм самоуваги в завдання семантичної сегментації та розроблено мережеву модель DANet, яка доводить, що механізм самоуваги також може бути застосовний у задачі семантичної сегментації.

1.4 Постановка задачі

Для вирішення задачі семантичної сегментації обирається одна з моделей, які краще за все підходять під специфіку даних та обмежень по часу та наявним ресурсам. Слід звертати увагу на такі фактори, як наявна кількість зображень для тренування, розмір зображень, що на них зображено.

Бажаним є результат, розміри якого дорівнюють розмірам вхідних даних. Через це постає проблема обчислювальної складності системи, адже прогін всього зображення через кожний шар мережі є дуже затратним. Через це, використовується техніка зменшення розмірів дати при поглибленні до центру мережі. Таким чином, поверхневі шари концентрують увагу на деталях низького рівню, в той час як більш глибокі - загальні деталі високого рівню. Не дивлячись на наявність систем, що залишають вихідні дані зменшеного розміру - існує ряд технік для його збільшення. Така структура називається енкодер-декодер і її дуже широко використовують у сфері сегментації зображень. Архітектуру такої мережі часто представляють у вигляді двох гілок, що представляють енкодер і декодер, відповідно. У той час як можно просто збільшити результат лінійно, наївними методами, ряд систем пропонує більш продвинуті декодери, що використовують інформацію з відповідних рівнів енкодера. Прикладом такої системи є DeepLabv3+, що включає до обчислень контекстуальну інформацію з енкодера.

Зазвичай, моделі для вирішення задачі сегментації, базуються на CNN мережі. Такі стандартні моделі, як ResNet, VGG або MobileNet, обираються у якості бази у більшості випадків [25]. Зовнішні слої базової мережі використовуються у якості енкодеру, а залишок мережі будується на цій основі. Для більшості сегментаційних моделей, можна використовувати будь яку базову мережу.

Через те що різні моделі мають свої обмеження та переваги, постає питання, у якому випадку краще використовувати ту чи іншу архітектуру? Порівняння декількох моделей на прикладі дата сету із сценами міста є основним завданням цієї роботи.

Так, наприклад, PSPNet (The Pyramid Scene Parsing Network) оптимізована під вивчення глобального контексту сцени. Цю модель варто використовувати, наприклад, працюючи із зображеннями кімнати, або вулиці із різними за розміром об’єктами. За розміри вхідних даних варто брати приблизно 500x500.

Для зображень із сфери медицини набрала популярності UNet. Завдяки з’єднанням із пропусками, UNet не втрачає деталі. Також можна використовувати для зображень кімнат, вулиць з малими об’єктами.

Для простих дата сетів із значною кількістю зображень UNet і PSPNet не є оптимальними варінтами, адже спрямовані на більш складні задачі. У такому випадку краще використовувати звичайні FCM або SegNet. Рекомендується поекспериментувати з декількома сегментаційними моделями з різними розмірами вхідних даних, перш ніж робити кінцевий висновок.

Результат роботи системи для сегментації зображень є надзвичайно актуальним у наш час. Ця сфера дуже широка і використовується у значній кількості напрямів. Одним із таких напрямів є створення візуальної репрезентації медичних зображень для подальшої обробки. За рахунок процесу сегментації, на зображенні можна виділити конкретні клітини, або області, що мають потенційний інтерес. Таким чином можна попередити рак, за рахунок класифікації ракових клітин. Майбутнє класифікації у сфері медицини лежить за комп'ютерами і використанням нейронних мереж.

Ще один напрям активного використання сегментації - автономні автомобілі. Процес керування за відсутності водія потребує повного розуміння дорожньої ситуації зі сторони комп'ютера. Потенціальна загроза для людського життя і можливість створення аварійних ситуацій значно збільшують потреби щодо якості, та надійності такої системи. Такий автомобіль має чітко розрізняти об'єкти інших авто, дороги, пішоходів, дорожніх знаків, та спеціальних поміток.

Більше того, швидкість обчислень має бути дуже високою, адже затримки можуть викликати аварійні ситуації. У місцях заторів, кількість автомобілів на зображенні може бути дуже високою, тому система має бути готовою до критичних випадків.

Сфера додаткової реальності також використовує результати сегментації зображень. Одним із відомих прикладів є гра Pokémon GO, що визначає ряд об'єктів як на кімнатній, та і на вуличній сценах, для подальших дій. Завдяки цьому, ігрових персонажів можна побачити, наприклад, на столі, чи у шафі, а не просто літаючими по екрану.

Розпізнавання обличчя також базується на процесі сегментації. Замість того, щоб обробляти обличчя як єдиний об'єкт, системи розпізнавання ділять його на піделементи, такі як рот, ніс, чи очі. Це знайшло застосування не тільки у сфері охорони та безпеки, але і у повсякденному житті. Прикладом може бути розпізнавання обличчя через камеру телефону для покращення якості фото, шляхом автоматичних налаштувань фокусу. Навіть системи розпізнавання ока використовують сегментацію.

Система, що здатна сегментувати зображення, є основним елементом для подальшого використання у інших напрямках. Це фундамент, на основі якого створюються подальші системи, що обробляють результати сегментації. Широка сфера застосування і актуальність на сьогоднішній день - причини такої створення системи, що описана у цій роботі.

Об’єктом дослідження є процес семантичної сегментації зображення.

Предметом дослідження є структури, моделі та архітектури згорткових нейронних мереж.

Метою роботи є дослідження та порівняльний аналіз впливу архітектур згорткових нейронних мереж на ефективність сегментації об’єктів

Для реалізації поставленої мети необхідно виконати наступні завдання:

– аналіз сучасного стану задачі комп’ютерного зору та сегментації зображень;

– дослідження можливих технологій та підходів для вирішення поставленої задачі;

– розробка програмної реалізація з використанням обраних методів для сегментації об’єктів;

– аналіз отриманих результатів.

2 ТЕХНОЛОГІЇ ТА ПІДХОДИ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Огляд існуючих підходів

Перш за все, варто розглянути кожну запропоновану модель окремо, виділити обмеження, та рекомендовану сферу застосування.

2.1.1 UNet

Архітектура UNet була створена для розв’язання задачі сегментації біомедичних зображень [26]. В основі лежить ідея використання двох шляхів, шлях контракції (також відомий як енкодер, використовується для вивчення контексту зображення), та шлях симметричного збільшення зображення (декодер, використовується для точної локалізації за допомогою транспонованих конволюцій). Енкодер представляє собою традиційний стак згорткових та max pooling шарів. UNet мережа є повністю згортковою (Fully Convolutional Network, FCN), тобто складається лише зі згорткових шарів і здатна приймати зображення будь-якого розміру. Схему архітектури проілюстровано на рис. 2.1.

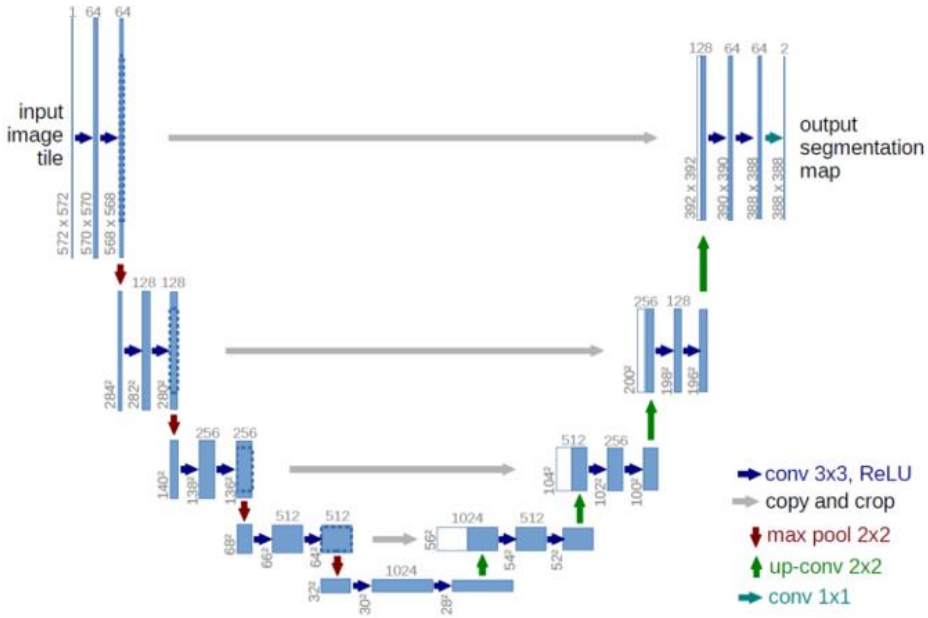


Рис. 2.1. Архітектура UNet

2.1.2 LinkNet

Більшість архітектур для сегментації використовують стратегію енкодер-декодер [27]. У якості прикладів можна навести Pyramid Architecture Network, PSPNet, та UNet. Результатом використання енкодеру є значне зменшення розміру зображення. Значною проблемою у сфері сегментації є подальше збільшення карти зображення до початкових розмірів, зберігаючи при цьому категоризацію пікселів (роль декодеру).

У LinkNet вхідні дані кожного шару енкодеру зберігаються для подальшого використання у якості виходу відповідного шару декодеру. Завдяки цьому просторова інформація зберігається і використовується декодером для більш точного збільшення зображення. Цей процес проілюстровано на рис. 2.2.

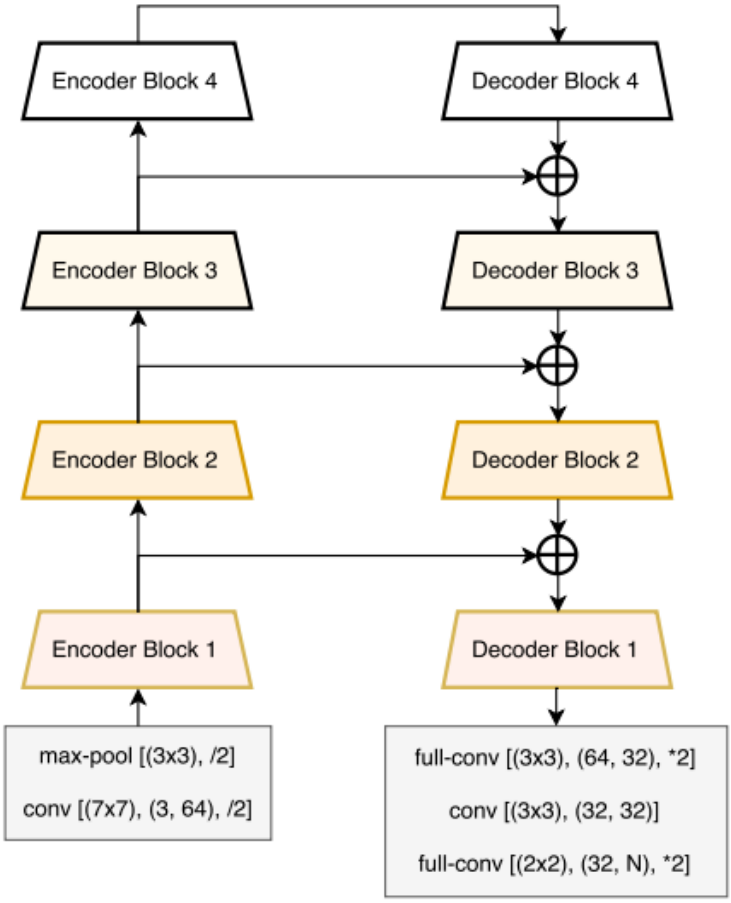


Рис. 2.2. Ілюстрація архітектури LinkNet

На перший погляд може бути важко побачити різницю від UNet, вона полягає, у використанні res-блоків, замість стандартної згорткової структури UNet.

2.1.3 FPN

Модель для сегментації FPN використовує структуру енкодер-декодер [28]. Регулярна згорткова мережа з повністю зв’язаним шаром використовується в ролі енкодеру. Декодер збільшує карту до початкових розмірів, використовується Feature Pyramid Network. Ілюстрацію даної архітектури можна побачити на рис. 2.3.

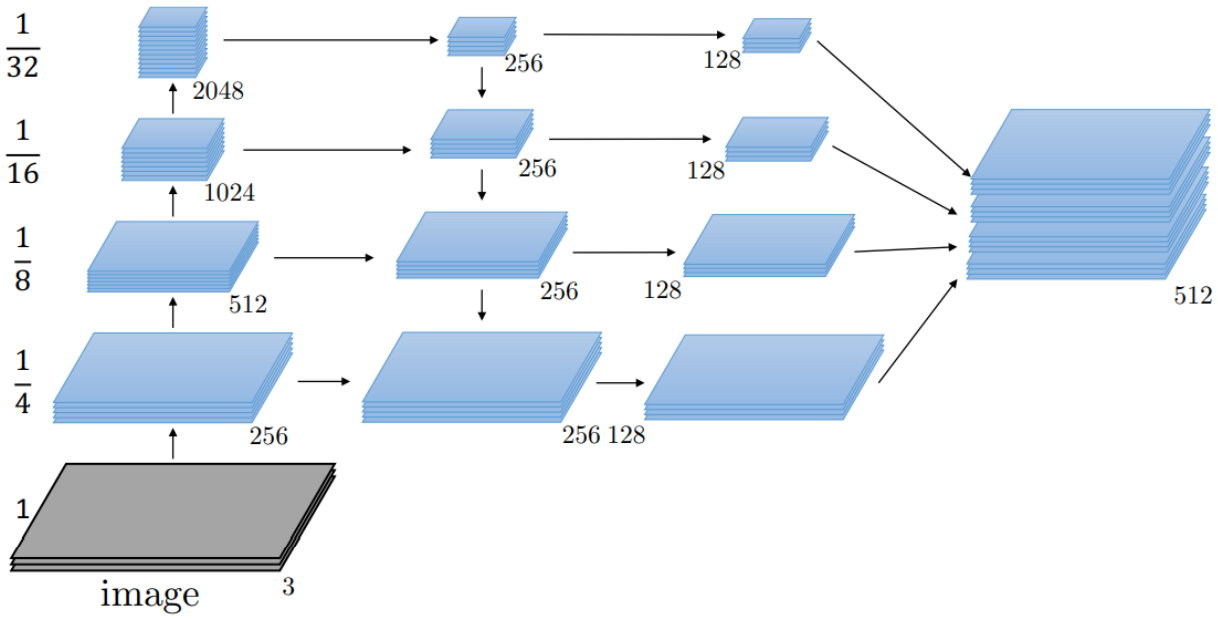


Рис. 2.3. Комбінація карт різних розмірів, FPN

Декодер комбінує карту малих розмірностей, семантично сильні риси високих розмірностей, семантично слабкі риси через прохід зверху-вниз і зв’язки з пропусками.

2.1.4 PSPNet

Архітектура PSPNet використовує глобальний контекст зображення для локальних передбачень, таким чином показуючи кращі результати на таких дата сетах як PASCAL VOC 2012 та cityscapes. Необхідність у створенні моделі була зумовлена тим, що FCN класифікатори не здатні до використання контексту всього зображення.

Головною частиною архітектури є Pyramid Pooling Module, дуже ефективний для отримання глобальної інформації зображення. Загальну архітектуру проілюстровано на рис. 2.4.

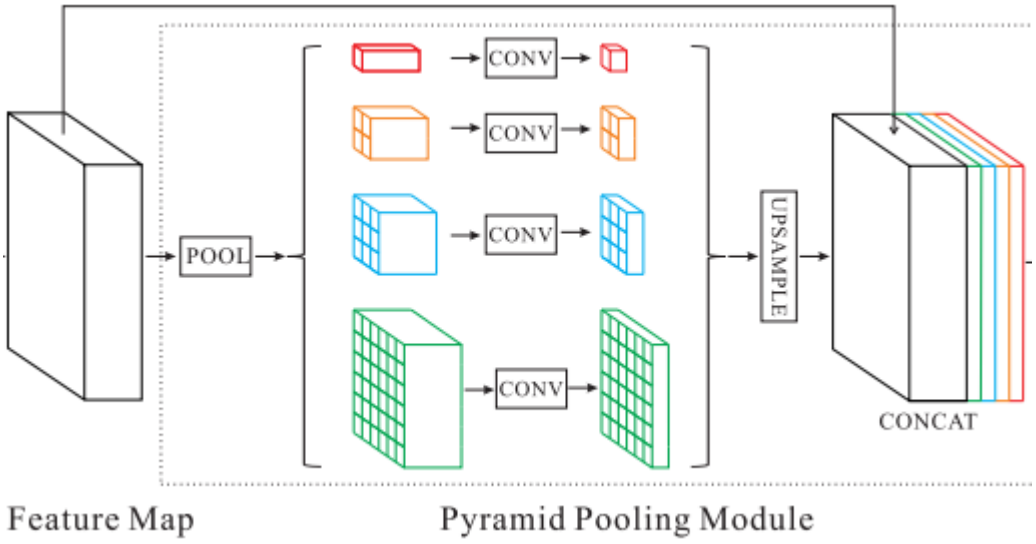


Рис. 2.4. Ілюстрація Pyramid Pooling Module

Спочатку проводиться пул карти рис до різних розмірів, результат передається до згортокового шару, після чого карти збільшуються до початкових розмірів зображення. Ця дата далі передається до декодеру, зберігаючи глобальний контекст, завдяки використанню різних розмірностей. Так, на схемі вище, різними кольорами зображені різні розмірності, у даному випадку 6, 3, 2 та 1, використовуючи зелений, блакитний, помаранчевий та червоний відповідно. Це розмірності, до яких зменшується карта рис, після чого результати згортаються

фільтрами 1x1 для зменшення глибини карти. Далі всі риси збільшуються до розміру карти та конкатенуються.

2.2 Детальний аналіз згорткових нейронних мереж

Згорткові нейронні мережі дуже схожі на звичайні нейронні мережі з попередньої глави: вони складаються з нейронів, які мають ваги та зміщення, які можна вивчати. Кожен нейрон отримує деякі вхідні дані, виконує точковий добуток і, за бажанням, слідує за ним з нелінійністю. Уся мережа все ще виражає єдину диференційовану функцію оцінки: від необроблених пікселів зображення на одному кінці до оцінок класу на іншому. І вони все ще мають функцію втрат (наприклад, SVM/Softmax) на останньому (повністю підключеному) шарі, і всі поради/трюки, які ми розробили для вивчення звичайних нейронних мереж, все ще застосовуються.

Архітектура ConvNet робить явне припущення, що вхідні дані є зображеннями, що дозволяє нам кодувати певні властивості в архітектурі. Тоді вони роблять функцію прямого доступу більш ефективною для реалізації та значно зменшують кількість параметрів у мережі.

Звичайні нейронні мережі отримують вхідні дані (один вектор) і перетворюють його через серію прихованих шарів . Кожен прихований шар складається з набору нейронів, де кожен нейрон повністю пов’язаний з усіма нейронами в попередньому шарі, і де нейрони в одному шарі функціонують повністю незалежно і не мають спільних зв’язків. Останній повністю підключений шар називається «вихідним шаром», і в налаштуваннях класифікації він представляє бали класу. Звичайні нейронні мережі погано масштабуються до повних зображень . До прикладу, зображення мають розмір 32x32x3 (32 ширини, 32 висоти, 3 колірних каналу), тому один повністю підключений нейрон у першому прихованому шарі звичайної нейронної мережі мав би 32*32*3 = 3072 ваги. Ця сума все ще здається керованою, але очевидно, що ця повністю пов’язана структура не масштабується до більших зображень. Наприклад, зображення більш респектабельного розміру, наприклад 200x200x3, призведе до нейронів, які мають

200*200*3 = 120 000 ваг. Очевидно, що це повне підключення є марнотратним, і величезна кількість параметрів швидко призведе до високих потреб в більш потужному обладнанні.

3D об'єми нейронів . Згорткові нейронні мережі використовують той факт, що вхідні дані складаються з зображень, і вони обмежують архітектуру більш розумним чином. Зокрема, на відміну від звичайної нейронної мережі, шари ConvNet мають нейрони, розташовані в трьох вимірах: ширина, висота, глибина. (глибина тут відноситься до третього виміру об’єму активації, а не до глибини повної нейронної мережі, яка може посилатися на загальну кількість шарів у мережі.) Наприклад, вхідні зображення є вхідним об’ємом активації, а обсяг має розміри 32x32x3 (ширина, висота, глибина відповідно). Нейрони в шарі будуть з’єднані лише з невеликою частиною шару перед ним, а не з усіма нейронами повністю пов’язаними способом. Більше того, кінцевий вихідний шар мав би розміри 1x1x10, тому що до кінця архітектури ConvNet ми зменшимо повне зображення в єдиний вектор оцінок класу, розташованих уздовж виміру глибини.

На рис.2.5 надано відмінності між стандартною трьохшаровою нейронною та згортковою нейронними мережами. Зліва: звичайна 3-шарова нейронна мережа. Праворуч: ConvNet розташовує свої нейрони в трьох вимірах (ширина, висота, глибина), як показано в одному з шарів. Кожен шар ConvNet перетворює 3D вхідний об’єм у 3D вихідний об’єм активації нейронів. У цьому прикладі червоний вхідний шар містить зображення, тому його ширина та висота будуть розмірами зображення, а глибина — 3 (червоний, зелений, синій канали).

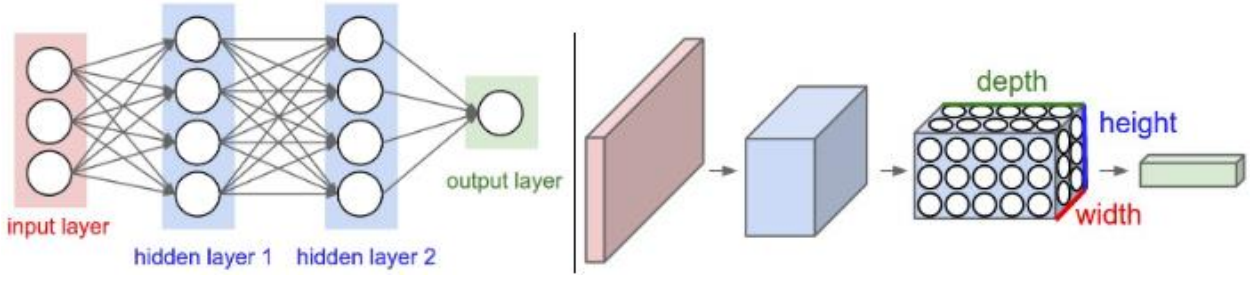


Рис. 2.5. Порівняння моделей звичайної нейронної мережі та згорткової

Метою операції згортки є вилучення з вхідного зображення високорівневих функцій , таких як краї. ConvNets не обов’язково обмежуються лише одним згортковим шаром. Зазвичай перший ConvLayer відповідає за захоплення функцій низького рівня, таких як краї, колір, орієнтація градієнта тощо. Завдяки доданим шарам архітектура також адаптується до функцій високого рівня, що дає нам повне розуміння мережі. зображень у наборі даних, як і ми.

2.2.1 Згортковий шар

Є два типи результатів операції — один, у якому розмірність згорнутого елемента зменшується порівняно з вхідним, а в іншому розмірність або збільшується, або залишається незмінною. Це робиться шляхом застосування дійсних відступів у випадку першого або Same Padding у випадку останнього. Тож заповнення: зображення 5x5x1 доповнюється нулями, щоб створити зображення 6x6x1. Коли ми збільшуємо зображення 5x5x1 до зображення 6x6x1, а потім застосовуємо до нього ядро 3x3x1, ми виявляємо, що згорнута матриця виявляється розміром 5x5x1. Звідси назва — Same Padding. З іншого боку, якщо ми виконаємо ту ж операцію без заповнення, нам буде представлена матриця, яка має розміри самого ядра (3x3x1) — Valid Padding.

2.2.2 Об’єднувальний шар

Подібно до шару згортки, шар об’єднання відповідає за зменшення просторового розміру згорнутого елемента. Це зменшує обчислювальну потужність, необхідну для обробки даних за рахунок зменшення розмірності. Крім того, він корисний для виділення домінуючих ознак , які є обертальними та позиційними інваріантами, таким чином підтримуючи процес ефективного навчання моделі.

Існує два типи об’єднання: максимальне об’єднання (Max Pooling) та середнє об’єднання (Average Pooling) (див. рис. 2.6). Max Pooling повертає максимальне значення з частини зображення, охопленої ядром. З іншого боку,

Average Pooling повертає середнє значення всіх значень з частини зображення, охопленої ядром.

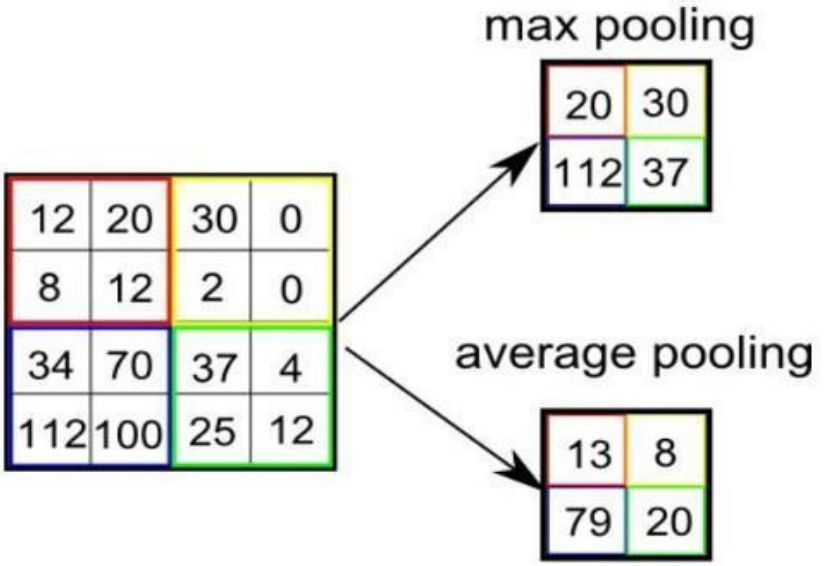


Рис. 2.6. Об’єднувальний шар

Max Pooling також виконує роль зниження шуму. Він повністю відкидає шумні активації, а також виконує видалення шумів разом із зменшенням розмірності. З іншого боку, Average Pooling просто виконує зменшення розмірності як механізм придушення шуму. Отже, ми можемо сказати, що Max Pooling працює набагато краще, ніж Average Pooling. Згортковий шар і шар об’єднання разом утворюють і-й шар згорткової нейронної мережі. Залежно від складності зображень кількість таких шарів може бути збільшена для ще більшого захоплення низькорівневих деталей, але ціною більшої обчислювальної потужності.

Продовжуючи, вирівняють кінцевий результат і передають його звичайній нейронній мережі для цілей класифікації.

2.2.3 Класифікація

Додавання повністю підключеного шару є (зазвичай) дешевим способом вивчення нелінійних комбінацій високорівневих функцій, представлених результатом згорткового шару (див. рис. 2.7). Повністю підключений шар вивчає, можливо, нелінійну функцію в цьому просторі.

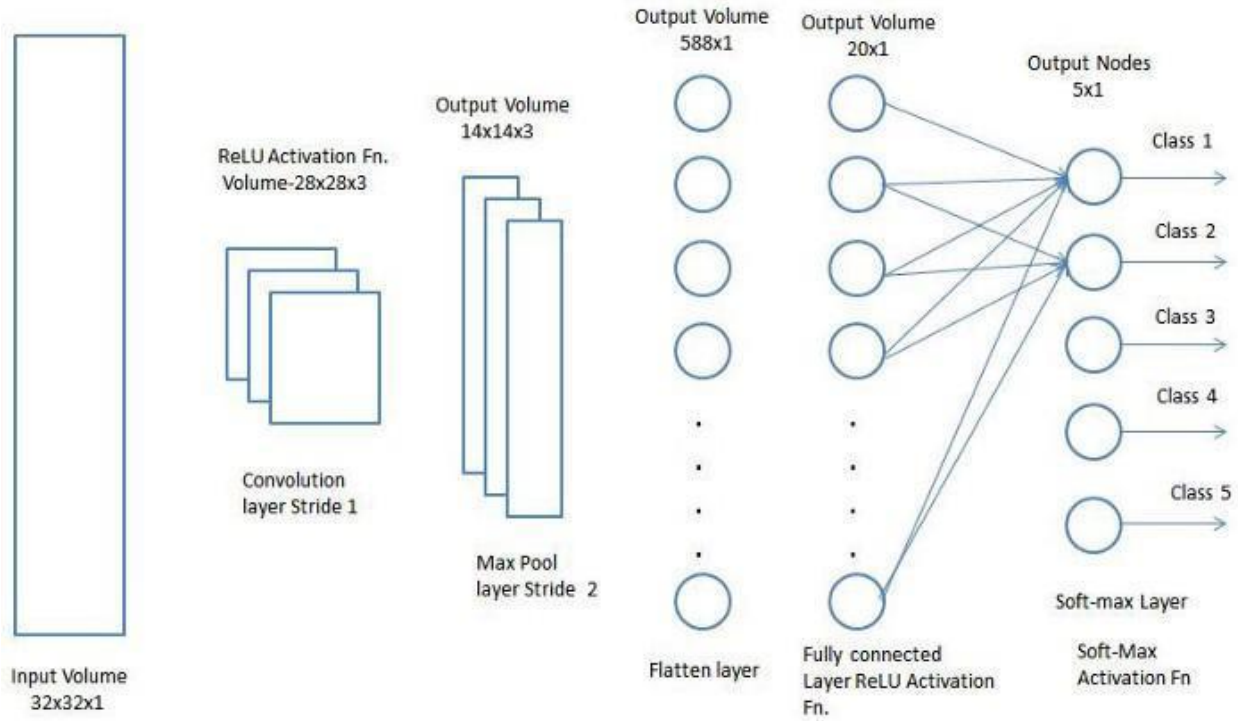


Рис. 2.7. Класифікація шарів

Тепер, коли вхідне зображення перетворено у відповідну форму для багаторівневого персептрона, зрівнюємо зображення у вектор-стовпець. Зведений вихід подається в нейронну мережу з прямим зв’язком і зворотне поширення застосовується до кожної ітерації навчання. Протягом ряду епох модель здатна розрізняти домінуючі та певні низькорівневі ознаки зображень і класифікувати їх за допомогою техніки Softmax Classification.

Існують різноманітні архітектури CNN, які були ключовими у побудові алгоритмів, які забезпечують і забезпечують AI в цілому в осяжному майбутньому. Деякі з них перераховані нижче:

– LeNet;

- AlexNet;
- VGGNet;
- GoogleLeNet;
- ResNet;
- ZFNetFully Connected Layer.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ З ВИКОРИСТАННЯМ ОБРАНИХ МЕТОДІВ ДЛЯ СЕГМЕНТАЦІЇ ОБ’ЄКТІВ

3.1 Python

Python - мова програмування високого рівня. Великий вибір бібліотек є однією з головних причин, чому Python є найпопулярнішою мовою програмування, що використовується для штучного інтелекту (ШІ). Бібліотека — це модуль або група модулів, опублікованих різними джерелами, такими як PyPi , які містять попередньо написаний фрагмент коду, який дозволяє користувачам отримувати певні функції або виконувати різні дії. Бібліотеки Python надають елементи базового рівня, тому розробникам не доводиться кожен раз кодувати їх із самого початку.

Машинне навчання вимагає безперервної обробки даних, а бібліотеки Python дозволяють отримати доступ, обробляти та перетворювати дані. Ось деякі з найпоширеніших бібліотек, які можна використовувати для машинного навчання та ШІ:

– Scikit-learn для роботи з основними алгоритмами машинного навчання, такими як кластеризація, лінійні та логістичні регресії, регресія, класифікація та інші;

– Pandas для високорівневих структур даних і аналізу. Він дозволяє об’єднувати та фільтрувати дані, а також збирати їх із інших зовнішніх джерел, наприклад, Excel;

– Keras для глибокого навчання. Він дозволяє швидкі обчислення та створення прототипів, оскільки він використовує графічний процесор на додаток до ЦП комп’ютера;

– TensorFlow для роботи з глибоким навчанням шляхом налаштування, навчання та використання штучних нейронних мереж з масивними наборами даних;

– Matplotlib для створення двовимірних графіків, гістограм, діаграм та інших форм візуалізації;

– NLTK для роботи з обчислювальною лінгвістикою, розпізнавання природної мови та обробки;

– Scikit-зображення для обробки зображень;

– PyBrain для нейронних мереж, навчання без нагляду та навчання з підкріпленням;

– Caffe для глибокого навчання, що дозволяє перемикатися між процесором і графічним процесором і обробляти понад 60 мільйонів зображень на день за допомогою одного GPU NVIDIA K40;

– StatsModels для статистичних алгоритмів і дослідження даних.

У репозиторії PyPI можна знайти та порівняти більше бібліотек Python.

Працювати в індустрії машинного машинного навчання та штучного інтелекту означає мати справу з купою даних, які потрібно обробляти найбільш зручним та ефективним способом. Низький бар’єр для входу дозволяє більшій кількості спеціалістів з даних швидко опанувати Python і почати використовувати його для розробки ШІ, не витрачаючи зайвих зусиль на вивчення мови.

Мова програмування Python нагадує повсякденну англійську мову, і це полегшує процес навчання. Його простий синтаксис дозволяє комфортно працювати зі складними системами, забезпечуючи чіткі відносини між елементами системи, також є можливість вибрати або використовувати ООП або сценарії. Також немає необхідності перекомпілювати вихідний код, розробники можуть впровадити будь-які зміни та швидко побачити результати.

Гнучкість Python дозволяє розробникам вибирати стилі програмування, які їм цілком підходять, або навіть комбінувати ці стилі для вирішення різних типів проблем найефективнішим способом:

– імперативний стиль складається з команд, які описують, як комп’ютер повинен виконувати ці команди. За допомогою цього стилю ви визначаєте послідовність обчислень, які відбуваються як зміна стану програми;

– функціональний стиль також називають декларативним, оскільки він оголошує, які операції слід виконувати. Він не враховує стан програми, порівняно з імперативним стилем, він оголошує висловлювання у вигляді математичних рівнянь;

– об’єктно-орієнтований стиль ґрунтується на двох поняттях: клас і об’єкт, де подібні об’єкти утворюють класи. Цей стиль не повністю підтримується Python, оскільки він не може повністю виконати інкапсуляцію, але розробники все ще можуть використовувати цей стиль у обмеженій мірі;

– процедурний стиль є найпоширенішим серед новачків, оскільки виконує завдання в покроковому форматі. Він часто використовується для послідовності, ітерації, модульності та вибору.

Python не тільки зручний у використанні та легкий у освоєнні, але й дуже універсальний. Ми маємо на увазі, що Python для розробки машинного навчання може працювати на будь-якій платформі, включаючи Windows, MacOS, Linux, Unix та інші. Щоб перенести процес з однієї платформи на іншу, розробникам необхідно внести кілька невеликих змін і змінити деякі рядки коду, щоб створити виконувану форму коду для вибраної платформи.

Python пропонує багато функцій, які є корисними, зокрема, для ШІ та машинного навчання, і це робить його найкращою мовою для цих цілей. Не дивно, що різні галузі використовують Python для передбачень та інших завдань машинного навчання, такі як : подорожі; фінтех; перевезення; охорона здоров'я та інші.

3.2 Фреймворк TensorFlow

TensorFlow — це бібліотека програмного забезпечення з відкритим вихідним кодом для чисельних обчислень з використанням графіків потоків даних. Вузли в графі представляють математичні операції, а ребра графа представляють багатовимірні масиви даних (тензори), що передаються між ними. Гнучка архітектура дозволяє розгортати обчислення на одному або кількох центральних процесорів або графічних процесорів на настільному, серверному чи

мобільному пристрої за допомогою одного API. Спочатку TensorFlow був розроблений дослідниками та інженерами, які працювали в команді Google Brain в дослідницькій організації Google Machine Intelligence для цілей проведення машинного навчання та досліджень глибоких нейронних мереж, але система є достатньо загальною, щоб її можна було застосувати в багатьох інших областях. TensorFlow приймає дані у вигляді багатовимірних масивів вищих розмірів, які називаються тензорами. Багатовимірні масиви дуже зручні при обробці великих обсягів даних.

TensorFlow працює на основі графіків потоків даних, які мають вузли та ребра. Оскільки механізм виконання у вигляді графіків, набагато легше виконувати розподілений код TensorFlow на кластері комп’ютерів під час використання графічних процесорів.

TensorFlow підтримує обчислювальні пристрої як CPU, так і GPU.

Програми глибокого навчання дуже складні, а процес навчання вимагає багато обчислень. Це займає багато часу через великий розмір даних і включає кілька ітераційних процесів, математичні обчислення, множення матриці тощо. Якщо ви виконуєте ці дії на звичайному центральному процесорному блоці (ЦП), зазвичай це займе набагато більше часу.

Графічні процесори (GPU) популярні в контексті ігор, де потрібно, щоб екран і зображення мали високу роздільну здатність. Спочатку для цієї мети були розроблені графічні процесори. Однак вони також використовуються для розробки програм глибокого навчання.

Однією з основних переваг TensorFlow є те, що він підтримує графічні процесори, а також центральні процесори. Він також має швидший час компіляції, ніж інші бібліотеки глибокого навчання, такі як Keras і Torch.

На рис. 3.1 показана ієрархія наборів інструментів TensorFlow.

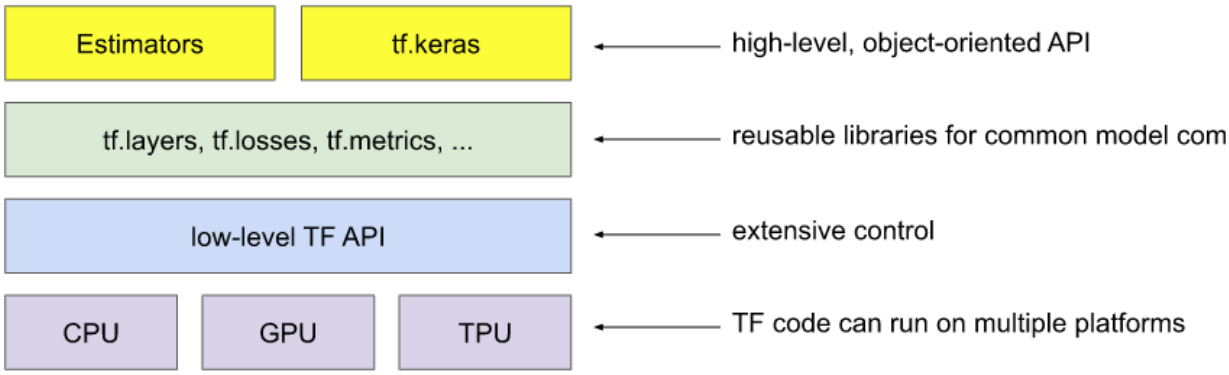


Рис. 3.1. Ієрархія інструментів TensorFlow

3.3 Інструменти анотації

Кожні кілька місяців на ринок виходить нова платформа даних для навчання, яка обіцяє нові інноваційні функції, швидше маркування або вищу точність. Легко заплутатися, намагаючись вибрати найкращий інструмент анотації зображень для конкретного випадку використання.

Оптимізація процесу анотації даних має вирішальне значення для забезпечення високої продуктивності та надійності моделі. Тому вибір правильного інструменту для проектів комп’ютерного зору є дуже важливий.

3.3.1 LabelMe

Інструмент анотації зображень, написаний на python. Підтримує анотацію зображення для багатокутника, прямокутника, кола, лінії та точки, а також анотацію прапора зображення для класифікації та очищення (див. рис. 3.2). З відкритим вихідним кодом і безкоштовно. Працює на Windows, Mac, Ubuntu або через Anaconda, Docker

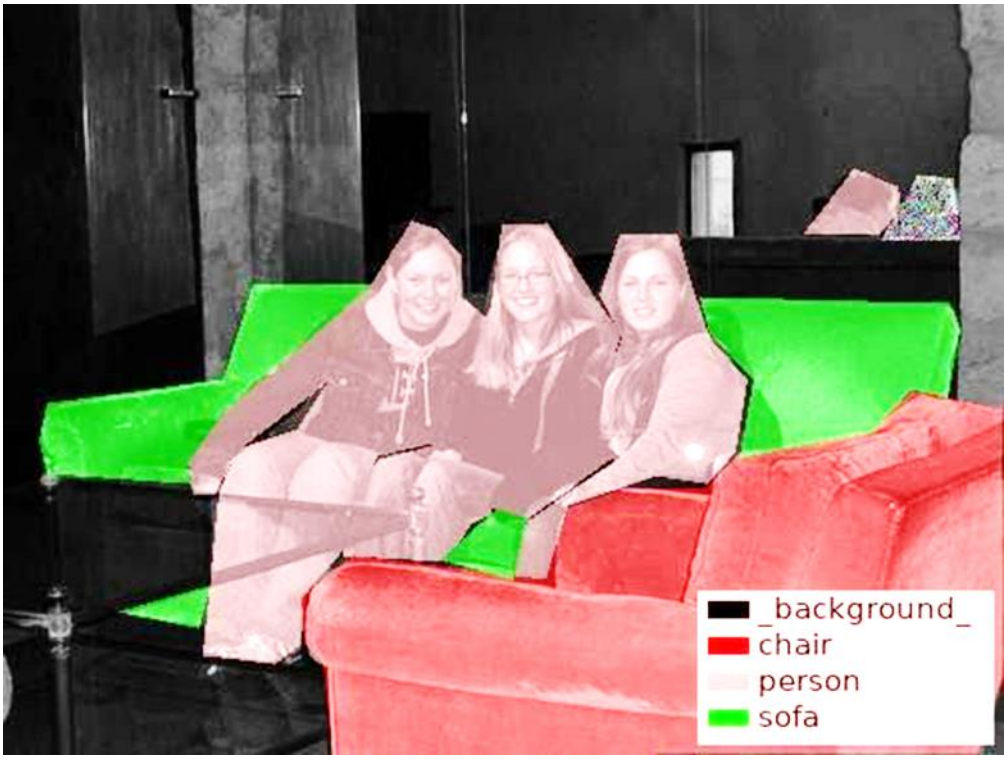


Рис. 3.2. Анотація LabelMe

3.3.2 CVAT

Computer Vision Annotation Tool (CVAT) – це безкоштовний інструмент із відкритим вихідним кодом для розмітки цифрових зображень та відео, розроблений Intel. Працює на Windows, Mac і Ubuntu. Основним його завданням є надання користувачеві зручних та ефективних засобів розмітки наборів даних. CVAT працює як універсальний сервіс, що підтримує різні типи та формати розмітки (див. рис. 3.3). Для кінцевих користувачів CVAT - це web-додаток, що працює у браузері. Він підтримує різні сценарії роботи і може бути використаний як для персональної, так командної роботи.

Рис. 3.3. Анотація CVAT

3.3.3 VIA

VGG Image Annotator — це безкоштовне, просте та самостійне програмне забезпечення для створення анотацій для зображень, аудіо та відео вручну. VIA працює у веб-браузері і не вимагає жодної установки чи налаштування. Повне програмне забезпечення VIA поміщається на одній автономній HTML-сторінці розміром менше 400 кілобайт, яка працює як автономна програма в більшості сучасних веб-браузерів. Підтримує багатокутні анотації та точки (див. рис. 3.4).

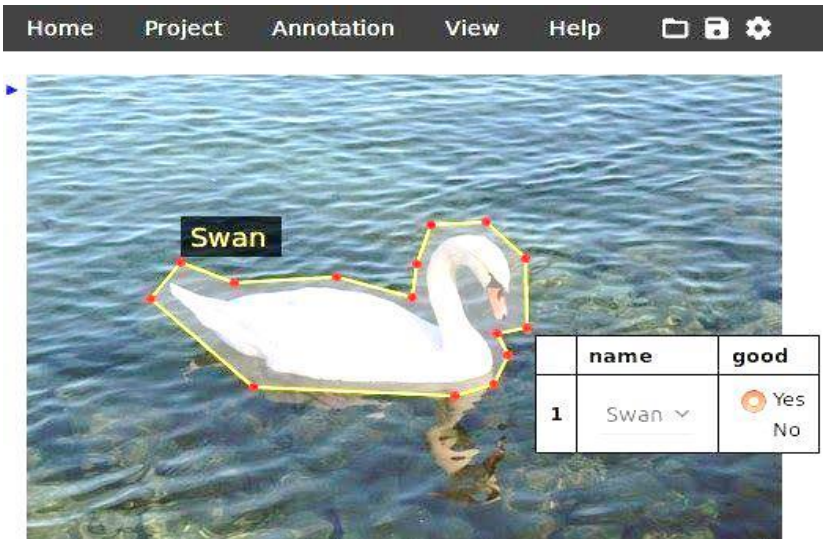


Рис. 3.4. Анотація VIA

3.3.4 Rectlabel

Платний інструмент анотації для MacOS. Може використовувати основні моделі машинного навчання для попереднього анотування зображень, підтримує багатокутники, кубічні без'є, лінії та точки (див. рис. 3.5).

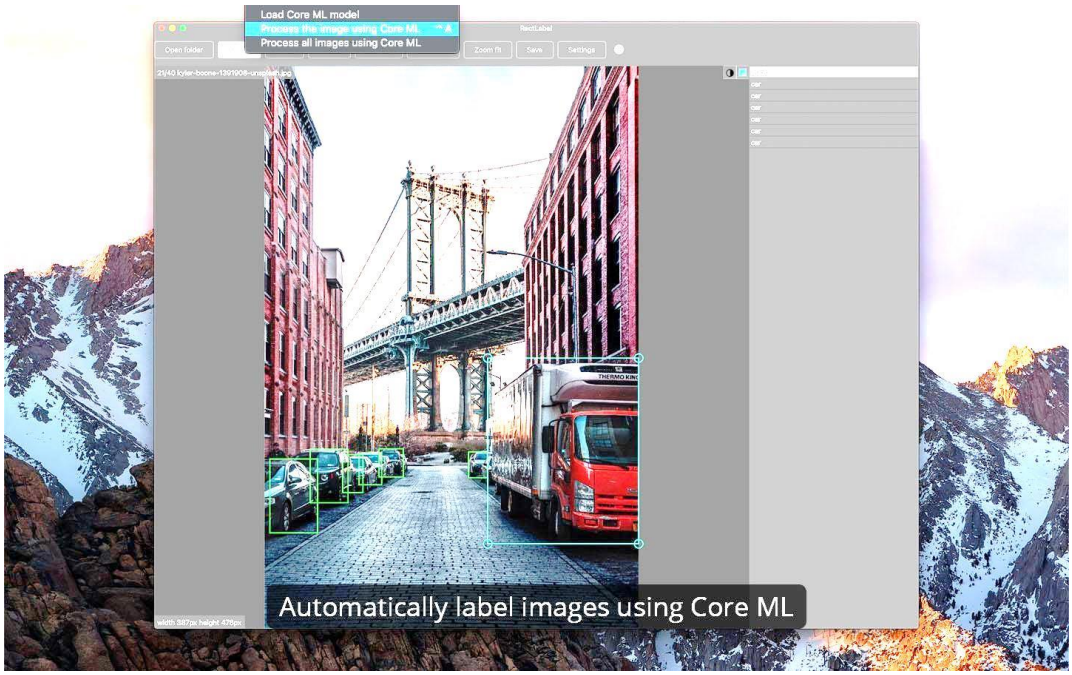


Рис. 3.5. Анотація Rectlabel

3.3.5 Labelbox

Платний інструмент анотації, підтримує інструмент перо для швидшого та точного створення анотацій (див. рис. 3.6). Налаштовується відповідно до точних вимог до структури даних (онтології). Векторна геометрія, класифікації, користувацькі атрибути, ієрархічні зв’язки та багато іншого доступні, щоб адаптувати необхідний варіант використання.

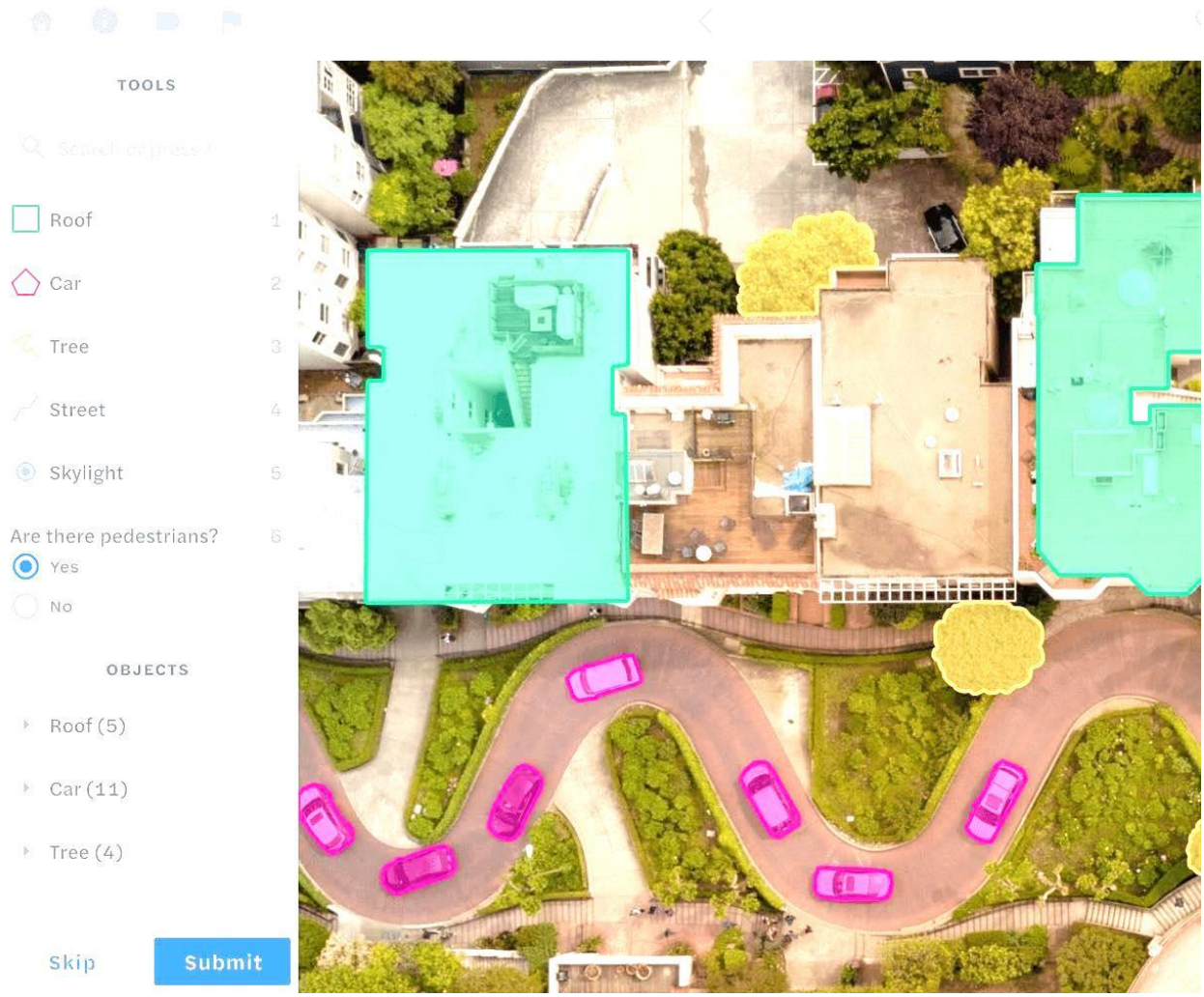


Рис. 3.6. Анотація Labelbox

3.4 Вибір наборів даних

Набори даних для машинного навчання (датасет) є важливою складовою, але їх отримання, підключення, та адаптація до системи є надто складні. Кожний датасет має свої складності та формати, тому кожному досліднику необхідно створювати свої програмні адаптації для завантаження та підготовки для роботи з різними датасетами.

3.4.1 Набір даних Pascal

Цей набір даних є розширенням набору даних Pascal VOC 2010 і виходить за рамки оригінального набору даних, надаючи анотації для всієї сцени та має понад 400 класів даних реального світу. На рис. 3.7 показано вхідна фотографія, а

на рис. 3.8 – вихідна з сегментацією зображення та обробленими мітками по різним об’єктам.

Рис. 3.7. Набір даних Pascal

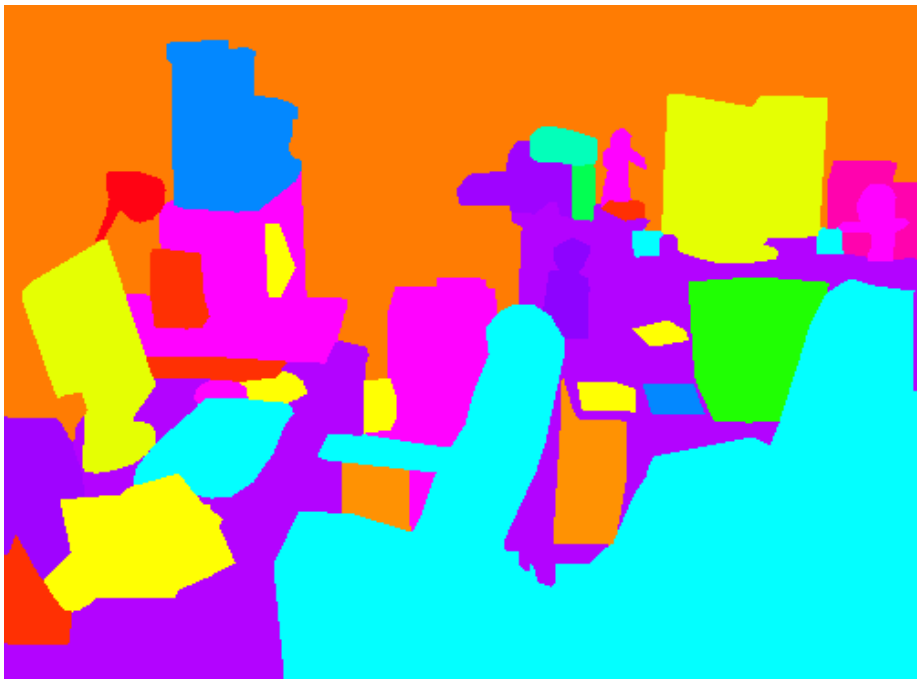


Рис. 3.8. Набір даних Pascal з сегментацією

3.4.2 Набір даних COCO

Набір даних COCO має 164 тисячі зображень вихідного набору даних COCO з анотаціями рівня пікселів і є звичайним набором даних для порівняння. Він охоплює 172 класи: 80 класів речей, 91 предметних класів і 1 клас "без міток" (див. рис. 3.9).

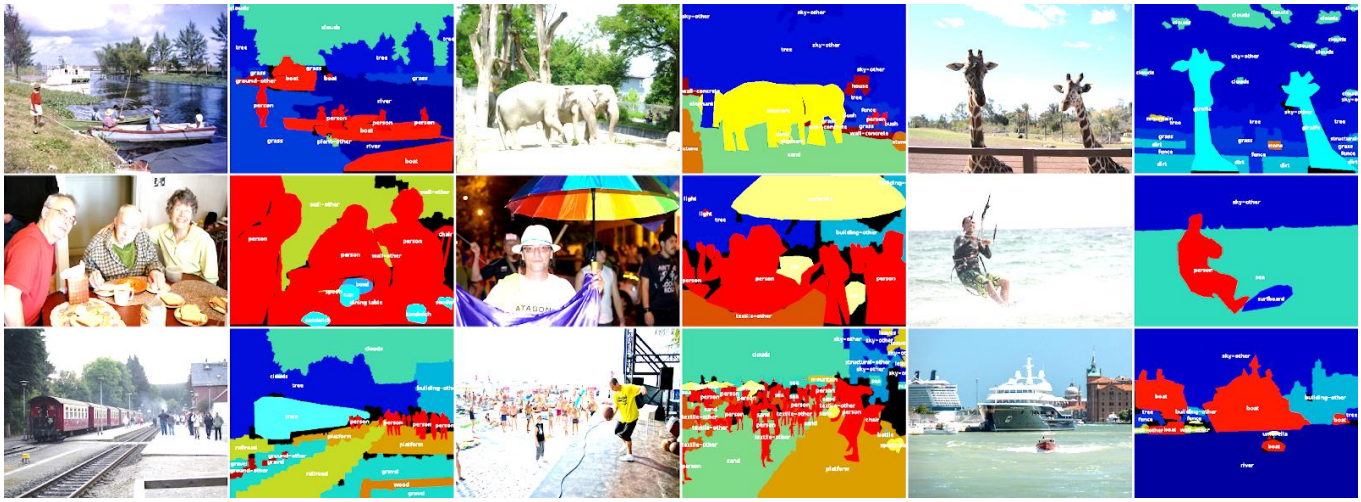


Рис. 3.9. Набір даних COCO

3.4.3 Набір даних міських пейзажів

Цей набір даних складається з істин сегментації для доріг, смуг, транспортних засобів та об’єктів на дорозі (див. рис. 3.10). Набір даних містить 30 класів і 50 міст, зібраних за різними екологічними та погодними умовами. Має також набір відеоданих з анотованими зображеннями, які можна використовувати для сегментації відео. Цей набір даних, можна використовувати для навчання самокерованих автомобілів.

Рис. 3.10. Набір даних міських пейзажів

3.4.4 Набір даних Lits

Набір даних був створений як частина завдання щодо виявлення пухлинних уражень за допомогою КТ печінки. Набір даних містить 130 КТ-сканування навчальних даних і 70 КТ-сканування даних тестування (див. рис. 3.11).

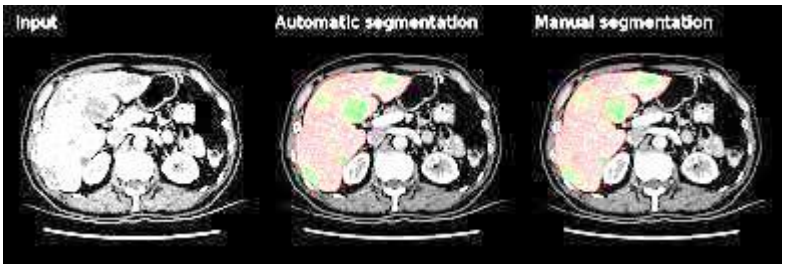


Рис. 3.11. Набір даних Lits

3.4.5 Набір даних CCP

Cloth Co-Parsing — це набір даних, створений як частина наукової роботи Clothing Co-Parsing за допомогою спільної сегментації та маркування зображень. Набір даних містить понад 1000 зображень з анотаціями на рівні пікселів, загалом 59 тегів (див. рис. 3.12).

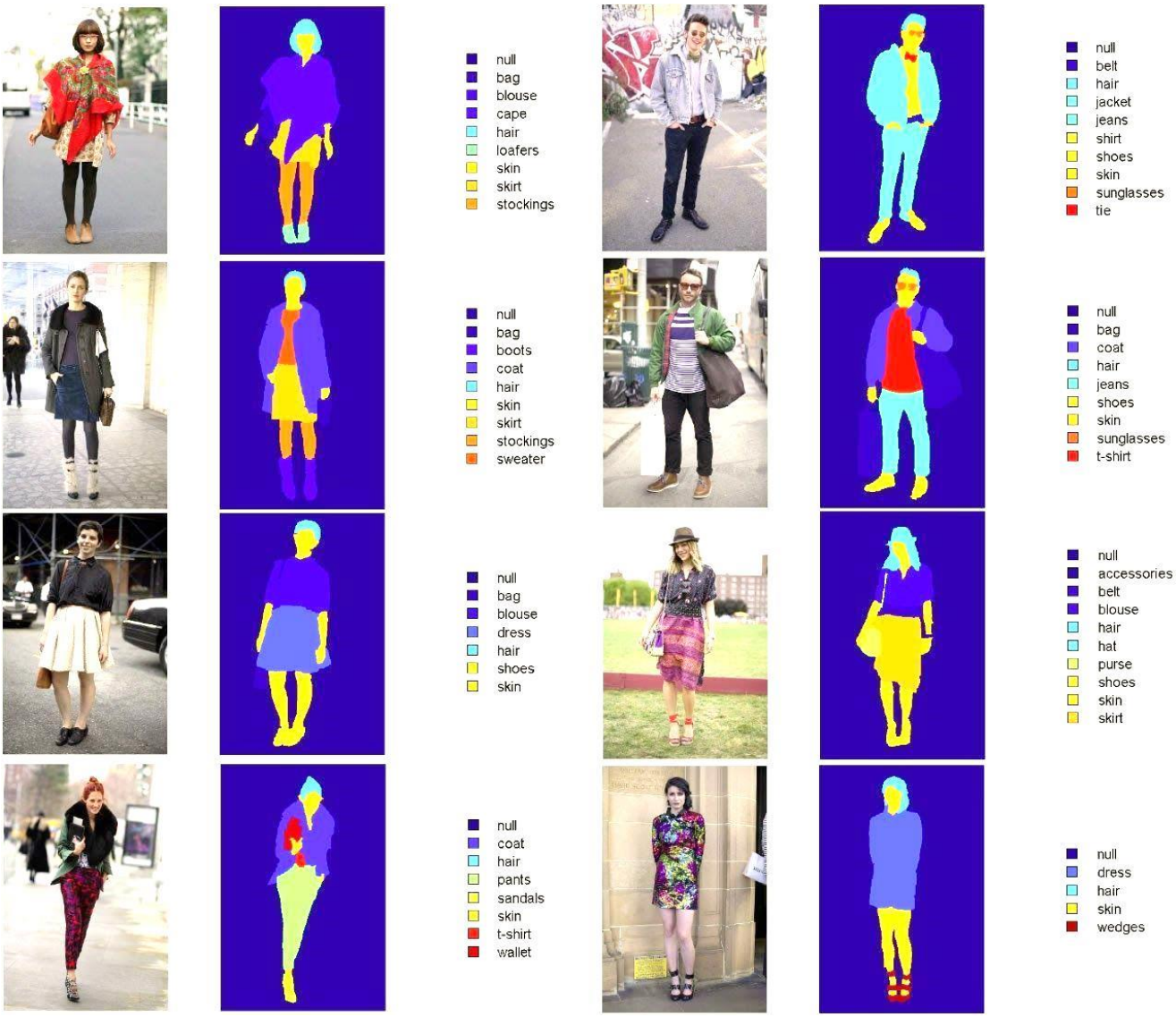


Рис. 3.12. Набір даних CPP

3.4.6 Набір даних Pratheepan

Набір даних, створений для завдання сегментації шкіри на основі зображень від Google, що містить 32 фотографії облич і 46 сімейних фотографій (див. рис. 3.13)

Рис. 3.13. Набір даних Pratheepan

3.4.7 Маркування повітряних зображень Inria

Набір даних карт аерофотосегментації, створених із зображень загального користування. Має охоплення 810 кв.км і має 2 класи будівельний і небудівельний. Викладено як начальне зображення (див. рис. 3.14), так і сегментоване (див. рис. 3.15).

Рис. 3.14. Набір даних Inria зображення

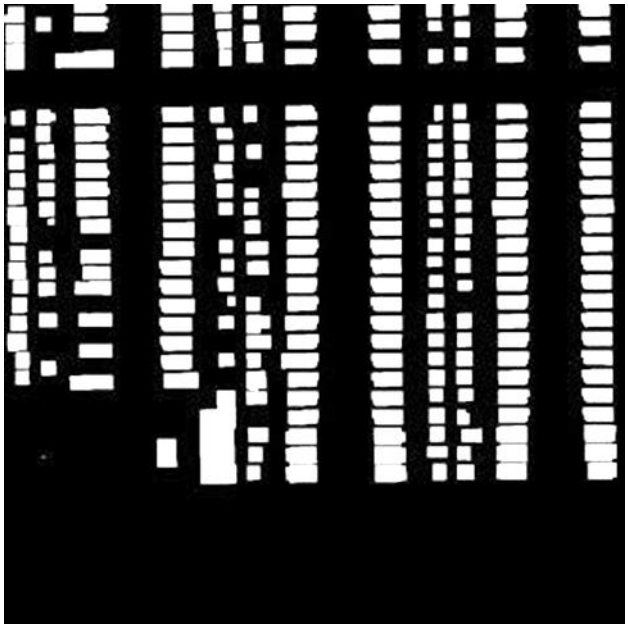


Рис. 3.15. Набір даних Inria з сегментацією зображення

3.4.8 S3DIS

Цей набір даних містить хмари точок із шести великомасштабних внутрішніх частин у 3 будівлях із понад 70 000 зображень (див. рис. 3.16).

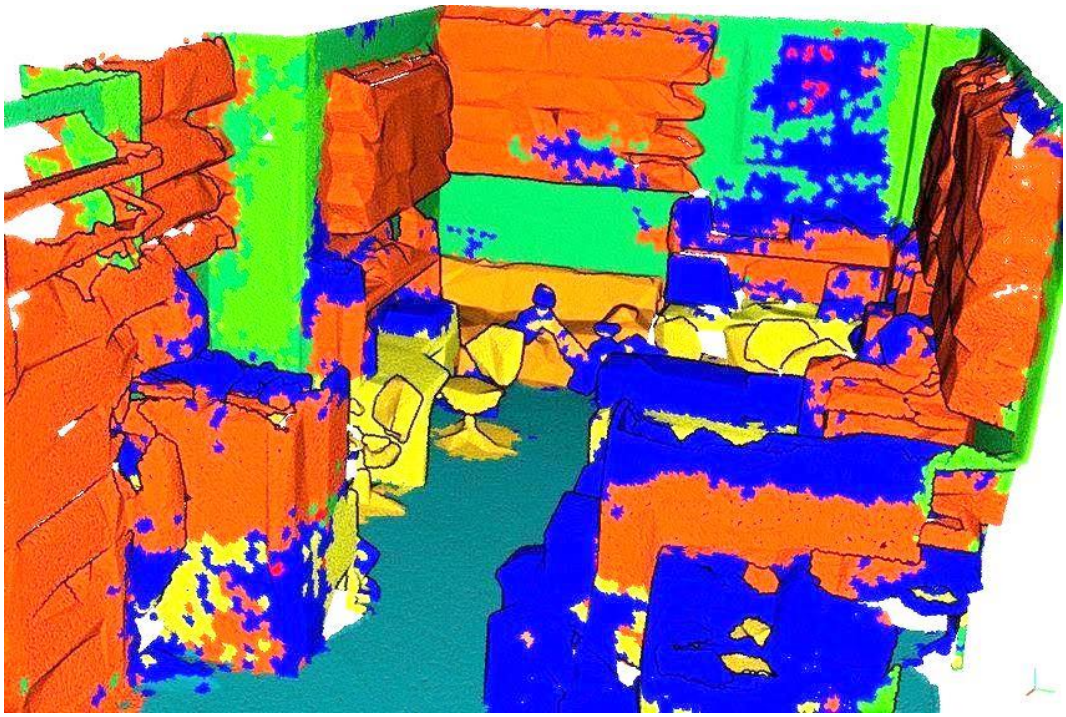


Рис. 3.16. Набір даних S3DIS

3.5 Модель системи

При роботі з будь-якою проблемою машинного навчання, завжди рекомендується провести певний час, намагаючись зрозуміти проблему. Це не тільки допомагає з пошуком необхідного забезпечення, але і сприяє мотивації розробника.

Для демонстраційних цілей буде використано датасет CamVid, та фреймворк TensorFlow. Датасет складається з зображень, відповідних підписів, та піксельних масок. Кожний піксель відноситься до однієї з 12-ти категорій: небо, будывля, стовб, дорога, тротуар, дерево, дорожій знак, забор, автівка, пішохід, велосипедист, відсутність підпису. На рис. 3.17 можна побачити приклад даних з датасету CamVid. Зліва показано початкове зображення, по центру - сегментаційну маску автівок, справа - сегментаційну маску неба.

Рис. 3.17. Демонстрація дати з CamVid Dataset

Дослідження буде проведено з такими популярними моделями як UNet, LinkNet та FPN [29]. У якості базової мережі використано efficientnetb3. Тренування проводиться на протязі 40 епох з темпом навчання 0.0001 і розміром пакету 4.

Для збільшення кількості дати і запобігання перетренування мережі проведено процес аугментації. Це потужна техніка, яка полягає у зміні тренувальних зображень для більш повного використання наявного матеріалу.

Враховуючи що CamVid достатньо малий дата сет, необхідно провести значну кількість аугментацій:

- горизонтальне відображення;
- перспективні трансформації;
- афінні трансформації;
- маніпуляції з яскравістю, контрастом, кольорами;
- розмиття зображення;
- гаусів шум;
- випадкове масштабування.

Для досягнення таких трансформацій буде використано швидку бібліотеку для аугментації Albumenations [30]. Одночасно з наданням широкого асортименту можливих аугментацій, Albumenations імплементує зрозумілий,але потужний аугментаційний інтерфейс для ряду різних задач комп’ютерного зору, включаючи класифікацію об’єктів, сегментацію та детекцію.

Вибір алгоритму оптимізації для моделі глибокого навчання може бути ключовою різницею між задовільними результатами за хвилини, години або дні. Алгоритм оптимізації Adam є доповненням до stochastic gradient descent, що здобуває все більше популярності для задач комп’ютерного зору за останні роки. Ключовою різницею є те, що темп навчання контролюється для кожної ваги окремо і змінюється у процесі тренування, в той час як stochastic gradient descent залишає його статичним. Автори описують Adam як комбінацію переваг двох інших доповнень - Adaptive Gradient Algorithm та Root Mean Square Propagation.

У якості активаційної функції використано softmax, що якраз підходить для мульти классової сегментації. У випадку бінарної сегментації можна використовувати sigmoid. У якості метрик для порівняння моделей взято IOU score та F-score. Дані подаються на вхід мережі тензорами 4x320x320x3, де 4 - розмір пакету, 320 - ширина та висота відповідно, а 3 - кількість каналів кольору.

Першою тренування пройде мережа з архітектурою UNet. Нагадаємо, що основою цієї моделі є використання структури енкодер-декодер, для отримання

глобального контексту та подальшого збільшення карти до початкових розмірів. На рис. 3.9 можна побачити результат передбачення мережею UNet на прикладі класів “автівки”, та “небо”.

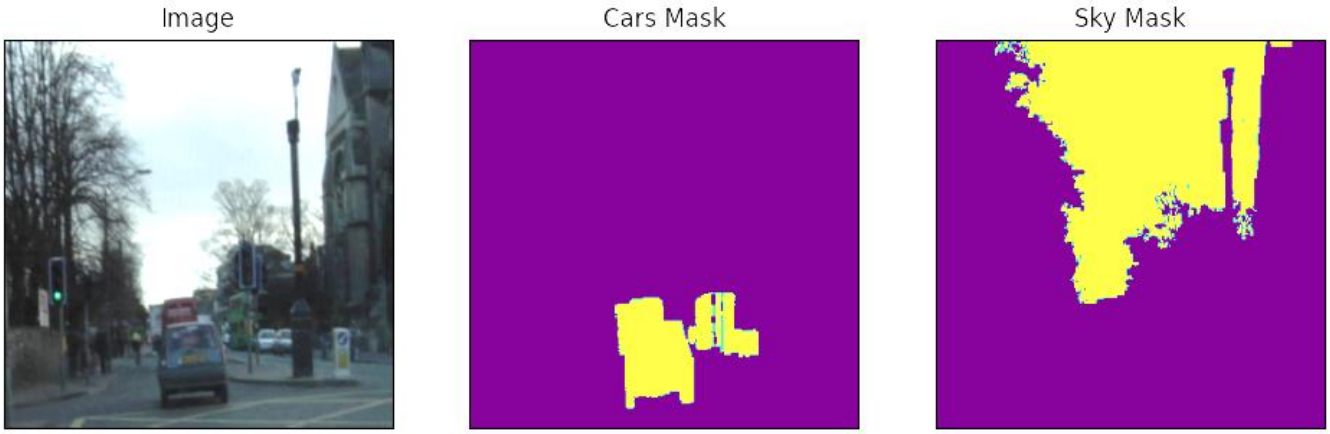
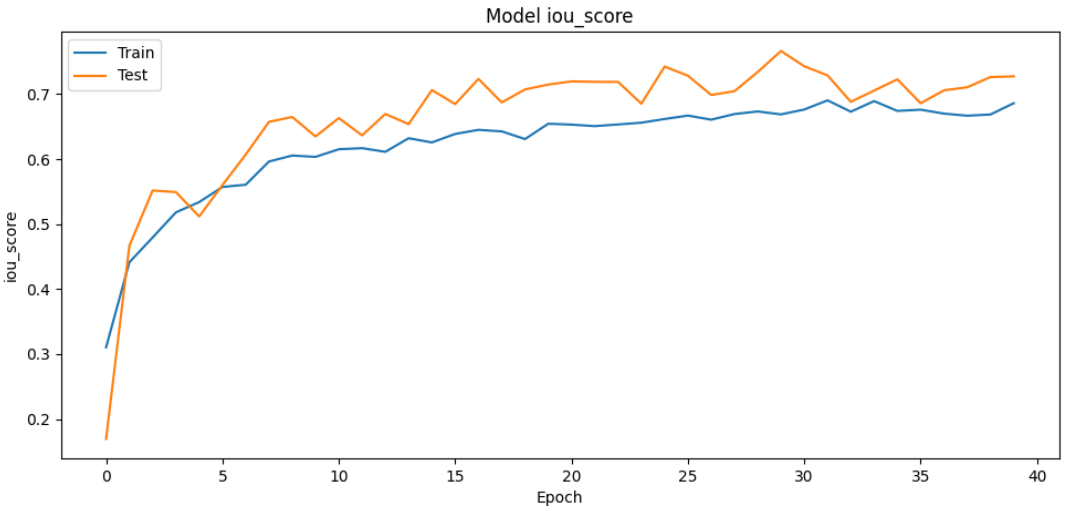


Рис. 3.18. Передбачення мережею UNet після 40 епох тренування

Результативні IOU score та загальна помилка моделі ілюстровані на рис. 3.18. Як можна бачити, валідаційна помилка не здобуває особливого прогрессу після епохи 15. Середнє значення IOU (тренувальне та валідаційне) після 40 епох сягає 0.7167, а середній F-score 0.7909.



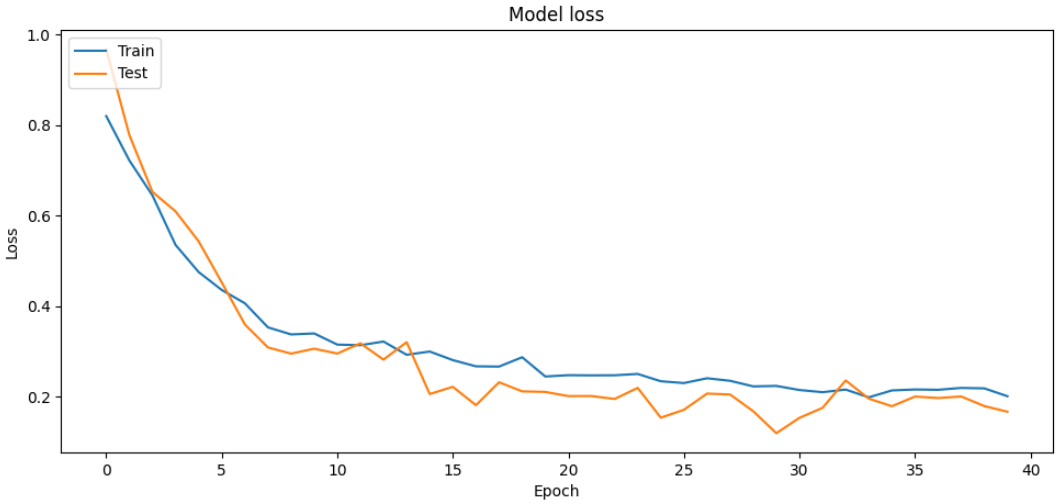


Рис. 3.19. IOU score та загальна помилка UNet

Наступну тренувальну ітерацію проведено з моделлю LinkNet. На рис. 3.20 зображене передбачення мережі після 40-ї епохи тренування.

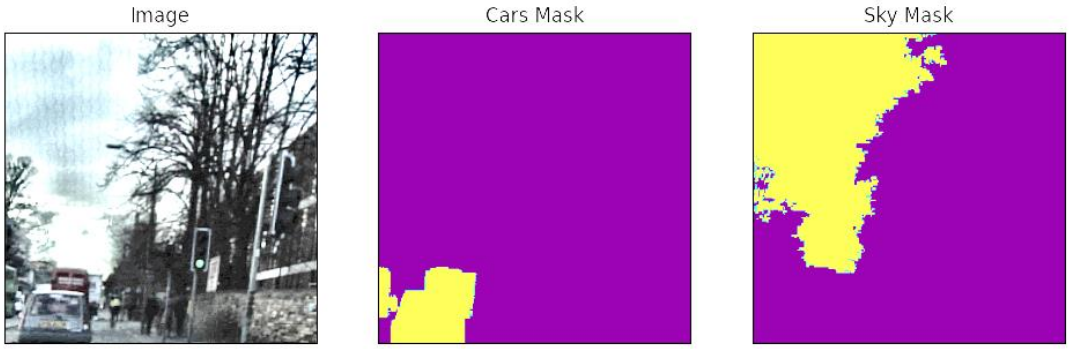


Рис. 3.20. Передбачення мережею LinkNet після 40 епох тренування

IOU score та помилку моделі можна побачити на рис. 3.21. Середнє значення IOU після 40 епох сягає 0.6812, а середній F-score 0.7565.

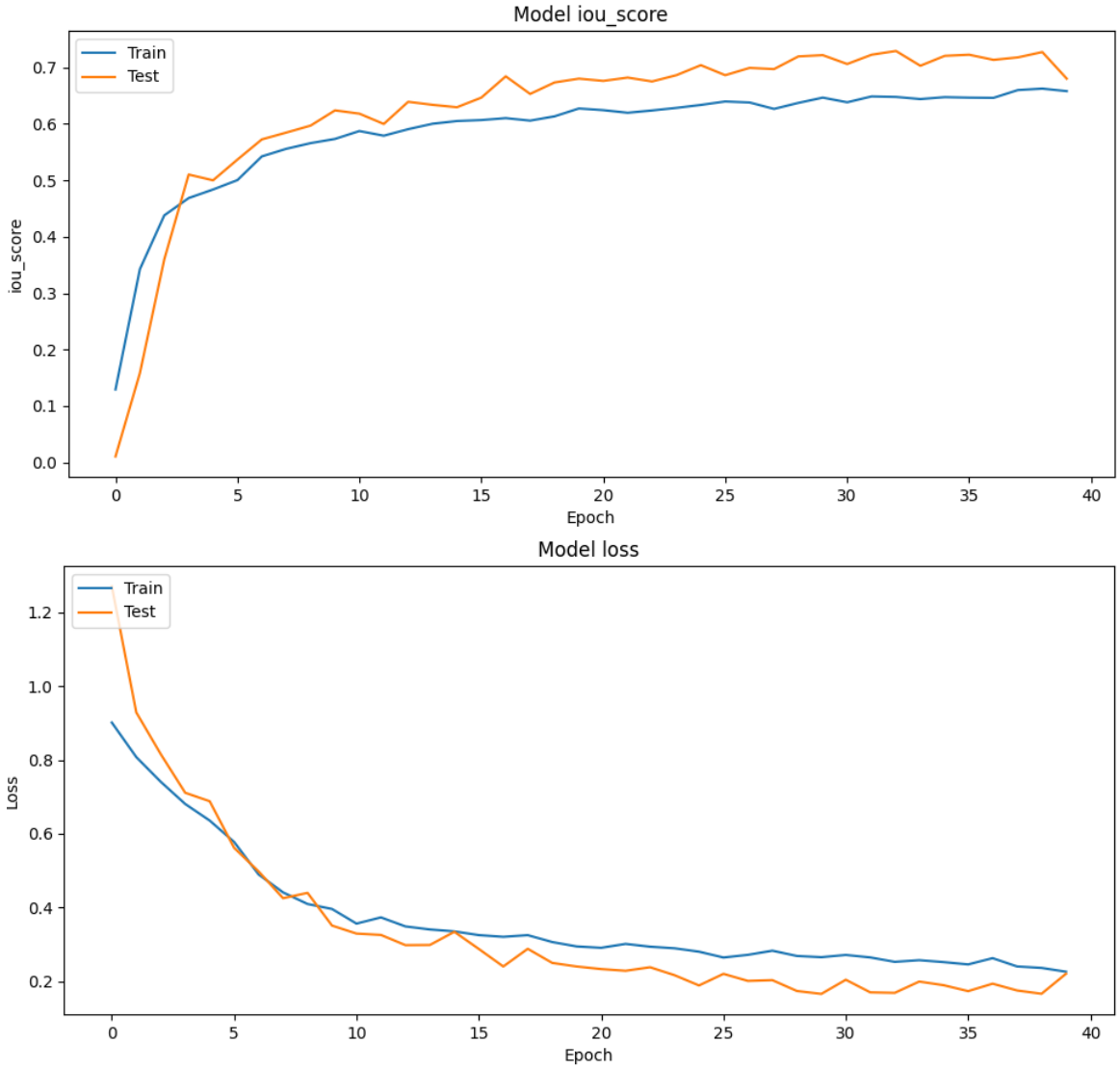


Рис. 3.21. IOU score та загальна помилка LinkNet

Наступний крок - тренування моделі FPN (Feature Pyramid Network). На рис. 3.22 можна зображене передбачення після 40-ї епохи тренування.

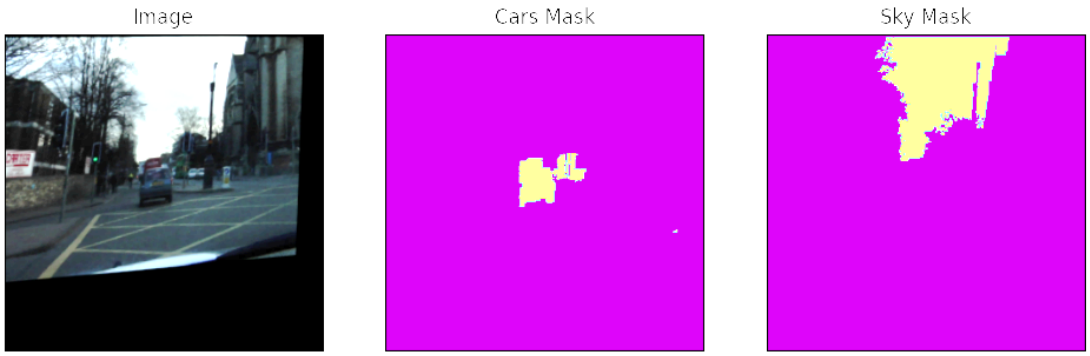


Рис. 3.22. Передбачення мережею FPN після 40 епох тренування

IOU score та помилку моделі можна побачити на рис. 3.23. Середнє значення IOU після 40 епох сягає 0.7073, а середній F-score 0.7819.

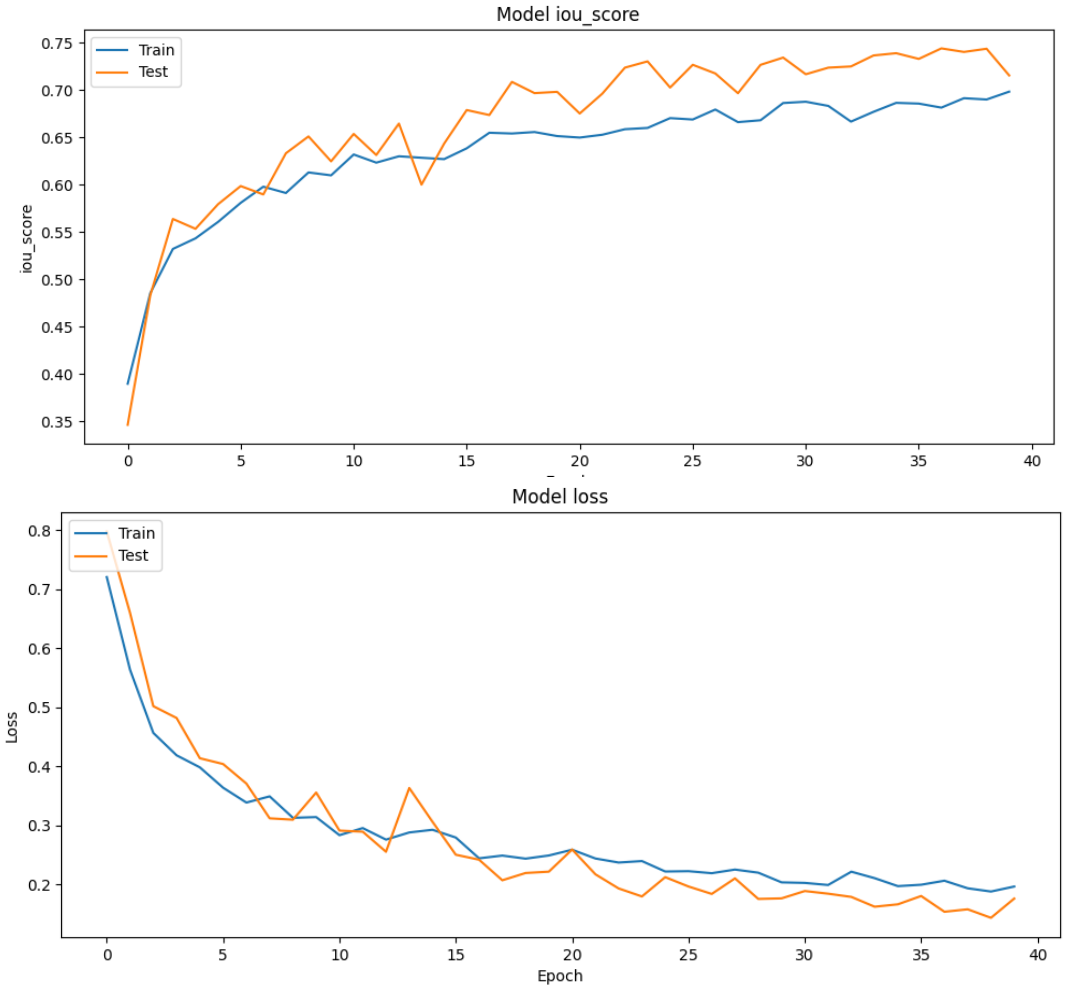


Рис. 3.23. IOU score та загальна помилка FPN

Нижче можна побачити порівняльну таблицю усіх вищеописаних моделей (див. Табл.3.1). До уваги взято помилки та оцінки після кожної десятої епохи.

Таблиця 3.1

Порівняльна таблиця

10 epoch	**Loss**	**IOU**	**F**	**Val Loss**	**Val IOU**	**Val F**
UNet	0.3394	0.6034	0.7055	0.3057*	0.6347*	0.7253*
LinkNet	0.3963	0.5734	0.6751	0.3513	0.6240	0.7097
FPN	0.3136*	0.6097*	0.7140*	0.3552	0.6245	0.7090
20 epoch	**Loss**	**IOU**	**F**	**Val Loss**	**Val IOU**	**Val F**
UNet	0.2442*	0.6543*	0.7542*	0.2101*	0.7146*	0.7850*
LinkNet	0.2948	0.6274	0.7282	0.2404	0.6803	0.7575
FPN	0.2486	0.6513	0.7542	0.2212	0.6980	0.7793
30 epoch	**Loss**	**IOU**	**F**	**Val Loss**	**Val IOU**	**Val F**
UNet	0.2234	0.6687	0.7662	0.1188*	0.7662*	0.8364*
LinkNet	0.2657	0.6467	0.7447	0.1662	0.7219	0.8035
FPN	0.2030*	0.6862*	0.7808*	0.1759	0.7342	0.8065
40 epoch	**Loss**	**IOU**	**F**	**Val Loss**	**Val IOU**	**Val F**
UNet	0.2006	0.6858	0.7820	0.1663*	0.7272*	0.8075*
LinkNet	0.2263	0.6582	0.7606	0.2208	0.6803	0.7598
FPN	0.1960*	0.6982*	0.7936*	0.1756	0.7153	0.7916

ВИСНОВКИ

У роботі було проведено аналіз джерел, що описують моделі вирішення окремих завдань комп’ютерного зору, зокрема для вирішення задач семантичної сегментації.

Проведено порівняльний аналіз використовуваних підходів і методів для вирішення завдання класифікації об’єктів на зображенні.

Було засвоєно та розібрано класифікацію зображення на піксельному рівні, як завдання комп’ютерного зору. Ця задача може бути сформульована як створення сегментаційної карти, де кожен піксель відповідає певному класу. У реальній системі пікселю може відповідати клас людини, дороги, тощо.

Для реалізації поставленої мети виконано наступні завдання:

– проаналізовано сучасний стан задачі комп’ютерного зору та сегментації зображень;

– досліджено можливі технологій та підходи для вирішення поставленої задачі, визначено їх основні недоліки та переваги;

– розроблено програмну реалізацію з використанням обраних методів для сегментації об’єктів;

– проаналізовано отримані результати.

Вивчення й вирішення проблем, пов'язаних із забезпеченням здорових та безпечних умов, у яких відбувається праця людини − одне з найбільш важливих завдань у розробці нових технологій і систем виробництва. Дослідження й виявлення можливих причин виробничих нещасних випадків, професійних захворювань, аварій, вибухів, пожеж, і розробка заходів і вимог, спрямованих на усунення цих причин дозволяють створити безпечні й сприятливі умови для праці людини. Комфортні й безпечні умови праці − один з основних факторів, який впливає на продуктивність і безпеку праці, здоров’я працівників.

Під час роботи над магістерською кваліфікаційною роботою не було виявлено жодних порушень з питань охорони праці. Технічний стан обладнання відповідав стандартам безпеки і нормам охорони праці, ніяких дефектів

обладнання під час виконання роботи не виявлено. Робоче місце було оснащене належним чином.

В результаті написання спеціальної частини з охорони праці було досягнуто поставленої мети, а саме створення безпечних і здорових умов праці на робочих місцях, в робочих зонах, у виробничих приміщеннях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Комп'ютерне бачення: Що це таке і чому це має значення? : веб-сайт. URL: https://www.sas.com/en_us/insights/analytics/computer-vision.html#:~:text=Computer%20vision%20is%20a%20field,to%20what%20they%20%E2%80%9Csee.%E2%80%9D (дата звернення: 01.01.2022).
2. Трофімов В.В. Інформаційні технології : підручник. Київ, 2015. 289 с.
3. Огляд семантичної сегментації зображення : веб-сайт. URL: https://www.jeremyjordan.me/semantic-segmentation/ (дата звернення: 01.01.2022).
4. Штучна нейронна мережа : веб-сайт. URL: https://en.wikipedia.org/wiki/Artificial_neural_network (дата звернення: 02.01.2022).
5. Конволюційна нейронна мережа : веб-сайт. URL: https://en.wikipedia.org/wiki/Convolutional_neural_network (дата звернення: 02.01.2022).
6. Тензор : веб-сайт. URL: https://en.wikipedia.org/wiki/Tensor (дата звернення: 02.01.2022).
7. Weakly- and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation : веб-сайт. URL: https://arxiv.org/pdf/1502.02734.pdf (дата звернення: 02.01.2022).
8. Fully Convolutional Networks for Semantic Segmentation : веб-сайт. URL: https://arxiv.org/pdf/1605.06211.pdf (дата звернення: 02.01.2022).
9. U-Net: Convolutional Networks for Biomedical Image Segmentation : веб-сайт. URL: https://arxiv.org/pdf/1505.04597.pdf (дата звернення: 03.01.2022).
10. Multi-scale context aggregation by dilated convolutions : веб-сайт. URL: https://arxiv.org/pdf/1511.07122.pdf (дата звернення: 03.01.2022).
11. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation : веб-сайт. URL: https://arxiv.org/pdf/1802.02611.pdf (дата звернення: 03.01.2022).

12. Improving Semantic Segmentation via Video Propagation and Label Relaxation : веб-сайт. URL: https://arxiv.org/pdf/1812.01593.pdf (дата звернення: 04.01.2022).

13. Gated-SCNN: Gated Shape CNNs for Semantic Segmentation : веб-сайт. URL: https://arxiv.org/pdf/1907.05740.pdf (дата звернення: 04.01.2022).

14. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation : веб-сайт. URL: https://arxiv.org/pdf/1511.00561.pdf (дата звернення: 04.01.2022).

15. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation : веб-сайт. URL: https://arxiv.org/pdf/1802.02611.pdf (дата звернення: 04.01.2022).

16. Вступ до сегментації зображень за допомогою кластеризації K-Means : веб-сайт. URL: https://www.kdnuggets.com/2019/08/introduction-image-segmentation-k-means-clustering.html (дата звернення: 04.01.2022).

17. Explained: Neural networks Ballyhooed artificial-intelligence technique known as "deep learning" revives 70-year-old idea. : веб-сайт. URL: http://news.mit.edu/2017/explained-neural-networks-deep-learning-0414 (дата звернення: 04.01.2022).

18. Convolutional Neural Networks (CNNs / ConvNets) : веб-сайт. URL: https://cs231n.github.io/convolutional-networks/ (дата звернення: 05.01.2022).

19. Neural Network : веб-сайт. URL: https://www.investopedia.com/terms/n/neuralnetwork.asp (дата звернення: 05.01.2022).

20. Rich feature hierarchies for accurate object detection and semantic segmentation : веб-сайт. URL: https://arxiv.org/pdf/1311.2524.pdf (дата звернення: 05.01.2022).

21. Image segmentation in 2020: Architectures, Losses, Datasets, and Frameworks : веб-сайт. URL: https://neptune.ai/blog/image-segmentation-in-2020 (дата звернення: 05.01.2022).

22. Master the COCO Dataset for Semantic Image Segmentation : веб-сайт. URL: https://towardsdatascience.com/master-the-coco-dataset-for-semantic-image-segmentation-part-1-of-2-732712631047 (дата звернення: 05.01.2022).

23. Microsoft COCO: Common Objects in Context : веб-сайт. URL: https://arxiv.org/pdf/1405.0312.pdf (дата звернення: 19.04.2020).

24. The PASCAL Visual Object Classes (VOC) Challenge : веб-сайт. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/PascalVOC_IJCV2009.pdf (дата звернення: 06.01.2022).

25. The Cityscapes Dataset for Semantic Urban Scene Understanding : веб-сайт. URL: https://arxiv.org/pdf/1604.01685.pdf (дата звернення: 06.01.2022).

26. Розуміння семантичної сегментації за допомогою UNET : веб-сайт. URL: https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47 (дата звернення: 06.01.2022).

27. Посібник для початківців із семантичної сегментації на основі глибокого навчання за допомогою Keras : веб-сайт. URL: https://divamgupta.com/image-segmentation/2019/06/06/deep-learning-semantic-segmentation-keras.html (дата звернення: 06.01.2022).

28. LinkNet : веб-сайт. URL: https://hasty.ai/content-hub/mp-wiki/model-architectures/linknet (дата звернення: 06.01.2022).

29. FPN : веб-сайт. URL: https://hasty.ai/content-hub/mp-wiki/model-architectures/fpn (дата звернення: 06.01.2022).

30. Сегментаційні моделі з попередньо підготовленим хребтами : веб-сайт. URL: https://github.com/qubvel/segmentation_models (дата звернення: 06.01.2022).

31. Бібліотека Альбументації : веб-сайт. URL: https://albumentations.ai (дата звернення: 06.01.2022).

32. Делікатне введення в алгоритм оптимізації Адама для глибокого навчання : веб-сайт. URL: https://machinelearningmastery.com/adam-optimization-algorithm-for-deeplearning/#:~:text=Adam%20is%20a%20replacement%20optimization,sparse%20gradients%20on%20noisy%20problems (дата звернення: 06.01.2022).