МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет прикладної математики та інформатики
(повне найменування назва факультету)

програмування
(повна назва кафедри)

Магістерська робота

СЕМАНТИЧНА СЕГМЕНТАЦІЯ ТИПОВИХ ОБ'ЄКТІВ У МІСТІ

Виконала: студентка групи ПМі – 61м
спеціальності
122 – комп'ютерні науки
(шифр і назва спеціальності)

Горлата С. Я.
(підпис) (прізвище та ініціали)

Керівник Гошко. Б. М.
(підпис) (прізвище та ініціали)

Рецензент
(підпис) (прізвище та ініціали)

Львів – 2020

Реферат

Робота складається із вступу, шести розділів, висновків та списку використаної літератури.

Обсяг магістерської роботи: 45 сторінки тексту, 47 рисунків.

Список використаних джерел: 22 позиції.

Мета даної роботи – розробка нейронної мережі, яка сегментуватиме типові об’єкти міста на фото.

У ході роботи над магістерською роботою було опрацьовано наукові статті з конференцій та теоретичні матеріали по згортковим нейронним мережам.

Було досліджено різні архітектури нейронних мереж, після чого було обрано U-net – архітектуру для реалізації практичної частини.

Для тренування нейронної мережі було використано відкритий набір даних CityScapes, що містить 4000 зображень для тренування і 500 зображень для валідації результатів.

Реалізація архітектури та розробка введення та виводу даних відбулась з використанням мови програмування Python. У ході реалізації мережі відбулось ознайомлення, а згодом детальніше вивчення фреймворку Tensorflow. У практичній частині застосовано Keras – інтерфейс високого рівня для Tensorflow, ряд допоміжних бібліотек для лінійної алгебри та легшого опрацювання даних.

Результати роботи нейронної мережі продемонстровані на рисунках в шостому розділі.

ЗМІСТ

ВСТУП 4
1 Постановка задачі 5
2 Сегментація зображень 6
2.1 Що таке сегментація? 6
2.2 Семантична сегментація і сегментація екземплярів 7
2.3 Які задачі вирішує сегментація? 7
3 Машинне навчання 11
3.1 Класичне машинне навчання 13
3.1.1 Навчання з вчителем 13
3.1.2 Навчання без вчителя 14
3.1.3 Ансамблі класифікаторів 17
3.1.4. Глибоке навчання 18
3.2 Штучні нейронні мережі 19
3.2.1 Функції активації 21
4 Згорткова нейронна мережа 25
4.1 Шар згортки 26
4.2 Пулінг 30
4.3 Модельні архітектури 32
4.3.1 U-net 33
5 Програмна реалізація 35
6 Результати 38
ВИСНОВКИ 42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 43

ВСТУП

Людство стає свідком монументального подолання розриву між можливостями людей та машин. Дослідники, інженери та просто ентузіасти працюють над багатьма аспектами галузі, щоб здійснити дивовижні речі. Їхня мета – дати можливість машинам бачити світ так, як це роблять люди, сприймати його подібним чином і навіть використовувати знання для безлічі завдань, таких як розпізнавання зображень та відео, аналіз та класифікація зображень, рекомендаційні системи, обробка природних мов тощо. І завдяки машинному навчанню, доволі широкій області досліджень, яка об'єднує в собі величезну кількість теоретичних і практичних методів.

Історія машинного навчання розпочалася в 1960-ті роки, коли інженери навчити комп'ютер грати в ігри. Проте величезний прорив стався у 1990-тих роках. Тоді і з'явилось поняття «нейронна мережа»

Сьогодні машинне навчання переживає свої найкращі часи. Ми спостерігаємо його прояви фактично у всіх галузях людської діяльності: таргетована реклама, розпізнавання пісня у аудіозаписі (Shazam), розпізнавання об'єктів на фото (GoogleLens), «автопілот» в машині Tesla.

Сегментація зображення - це широко використовуваний метод цифрової обробки та аналізу контенту для поділу об'єкту (зображення) на кілька частин або областей, часто заснований на характеристиках пікселів. Цей метод передбачає процес роз'єднання переднього плану від фону або кластеризацію областей пікселів за подібністю кольору та/або форми.

Люди можуть виконувати сегментацію зображення, націлено не окреслюючи об'єктів. Їх усвідомлення під час сегментації без навмисного визначення є важливою частиною зорового сприйняття. Це може стати надійним підґрунтям для створення потужної моделі ідентифікації середовища, а також застосуватись для створення або ж вдосконалення відомих людству варіацій (а вже в нього є свої методи бачення) комп'ютерного зору.

1 Постановка задачі

Розробити систему сегментації зображення. Виконання завдання вимагає вирішення наступних підзавдань:

1) підготувати набір даних;
2) розробити нейронну мережу;
3) натренувати нейронну мережу на підготовленому наборі даних;
4) проаналізувати отримані результати.

2 Сегментація зображень

2.1 Що таке сегментація?

Сегментація зображень передбачає поділ введеного зображення на сегменти для спрощення його аналізу. Сегменти представляють об'єкти або частини об'єктів і містять набори пікселів. Сегментація зображення сортує пікселі більшими компонентами, усуваючи необхідність розглядати окремі пікселі як одиниці спостереження [1, 11].

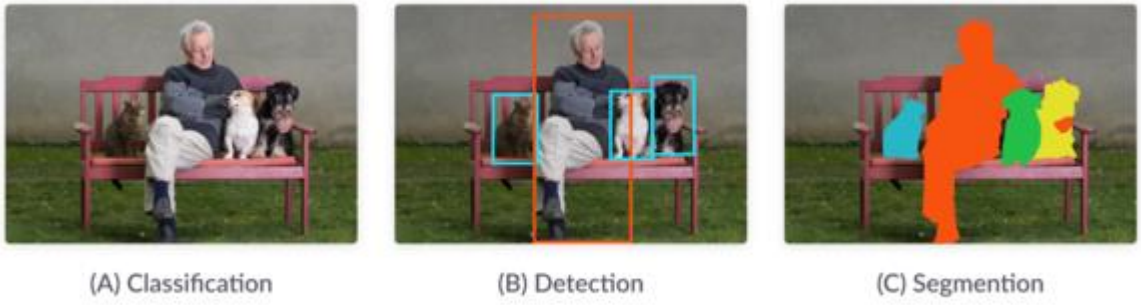

Рис. 2.1 – Типи аналізу зображень

Існує три типи аналізу зображень:

1) Класифікація - класифікація всього зображення за класом, таким як "люди", "тварини".
2) Виявлення об’єктів - виявлення об’єктів на зображенні та малювання прямокутника навколо них, наприклад, людини чи тварини.
3) Сегментація - виявлення частин зображення та розуміння, до якого шару вони належать. Сегментація закладає основу для виявлення та класифікації об’єктів.

2.2 Семантична сегментація і сегментація екземплярів

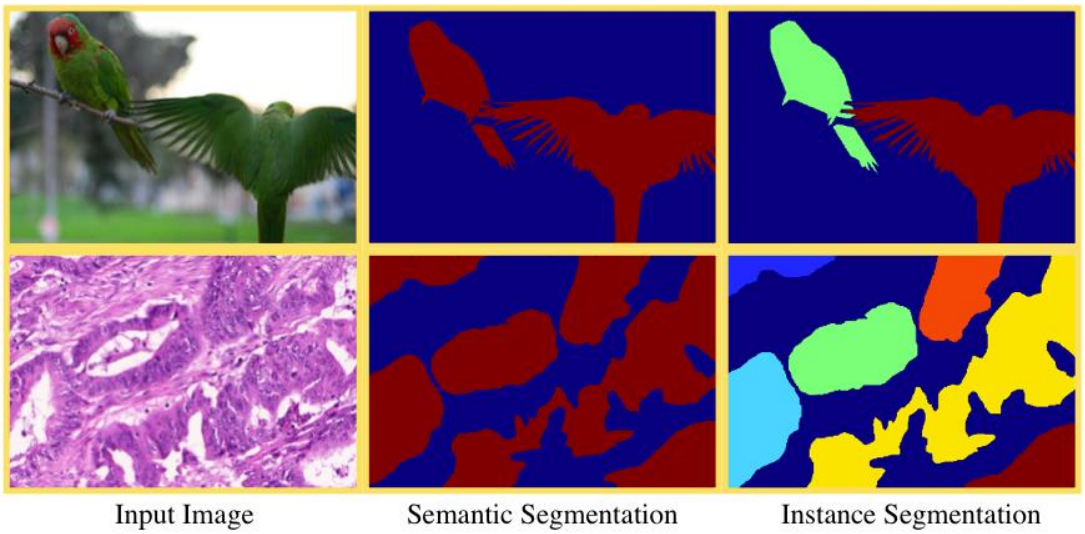

Рис.2.2 – Рівні деталізації сегментації

В рамках самого процесу сегментації існує два рівні деталізації:

1) Семантична сегментація - класифікує всі пікселі зображення на значущі класи об’єктів і відповідають реальним категоріям. Наприклад, можна виділити всі пікселі, пов’язані з папугами (рис.2.2), і забарвити їх у червоний колір.
2) Сегментація екземпляра - визначає кожен екземпляр кожного об’єкта на зображенні. Він відрізняється від семантичної сегментації тим, що не класифікує кожен піксель. Якщо на зображенні три папуги, семантична сегментація класифікує всіх, як один примірник, тоді як сегментація примірника ідентифікує кожного окремого папугу.

2.3 Які задачі вирішує сегментація?

Сегментація широко використовується в медицині. Завдяки сегментації рентген-знімків можна точніше дізнатись про форму внутрішніх органів (рис. 2.3) та локалізувати ділянки, уражені хворобою (рис. 2.4).

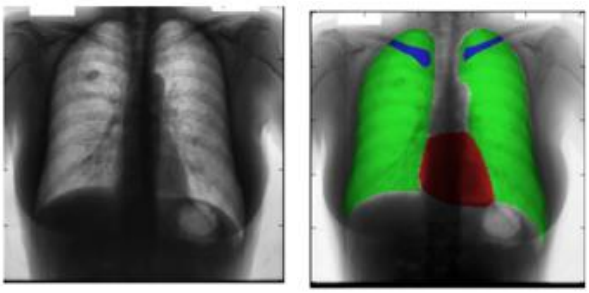

Рис.2.3 – Сегментація рентгену внутрішніх органів

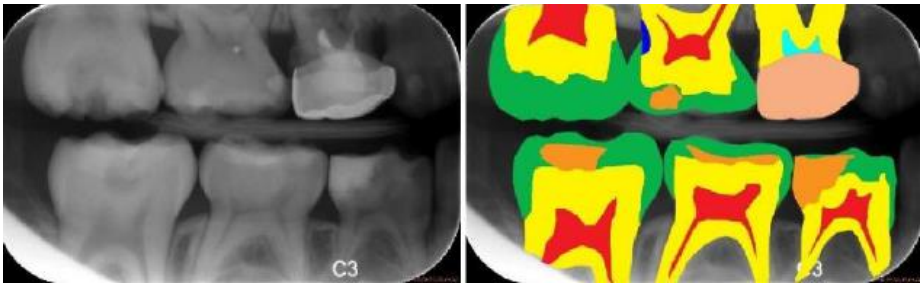

Рис.2.4 – Сегментація рентгену зубів у розрізі

У біології сегментація допомагає зрозуміти форму і структуру клітин, вивчити тканини і їх властивості, а також дослідити світ бактерій [2].

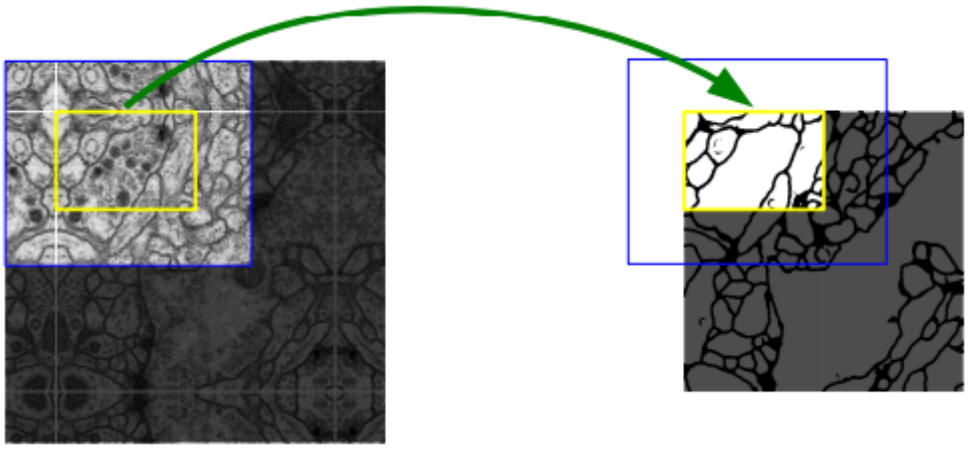

Рис.2.5 – Сегментація клітин епітелію

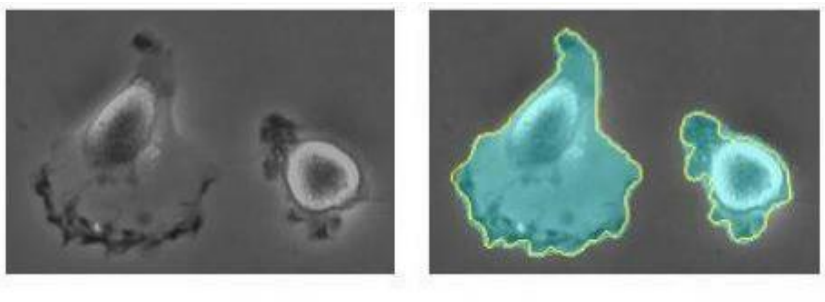

Рис.2.6 – Сегментація бактерій

Сегментація займає дуже важливе місце в картографії [10], адже саме із сегментованих знімків з супутника можна дослідити розмірність об’єктів, спроектувати ландшафт, виконати необхідні заміри для, наприклад, будівництва.

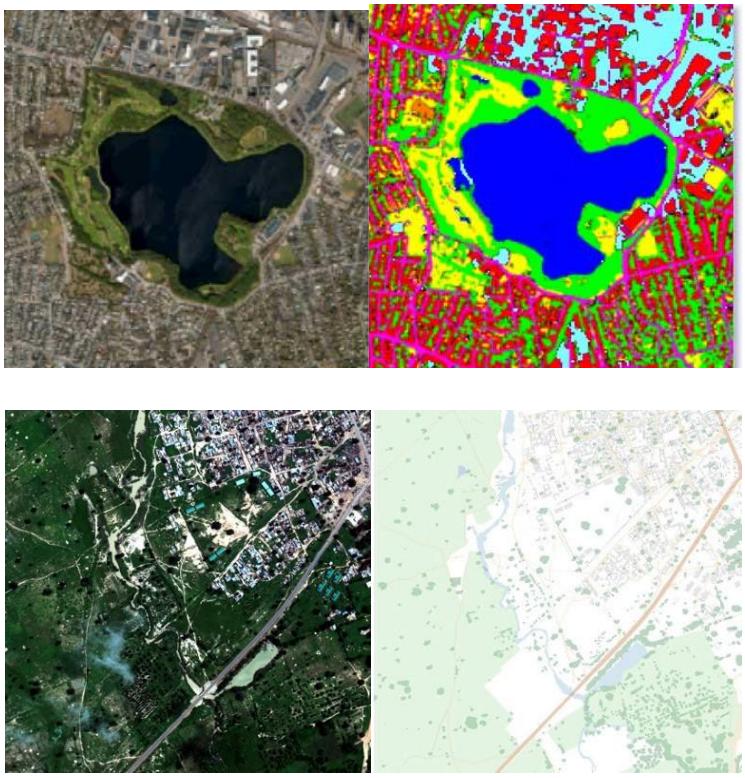

Рис.2.7 – Сегментація в картографії

Семантична сегментація використовується у вирішенні проблеми «автопілота» машини. Варто зазначити, що такого роду задача потребує дуже хорошого датасету і велику точність сегментації, щоб уникнути помилок під час поїздок.

Рис. 2.8 – Сегментація типових об’єктів у місті

3 Машинне навчання

Основною метою машинного навчання є правильне передбачення результату за вхідними даними. Чим різноманітніші вхідні дані, тим простіше машині знайти закономірності, а це гарантує більш точний результат. Для того, щоб добре «навчити машину», потрібно три основні складові [3]:

1) Велика кількість даних

Потрібно виявляти спам? Тоді розробникам необхідно зібрати якомога більше прикладів спам-листів.

Потрібно виявити закономірності падіння/злету курсу акцій? Необхідно зібрати історію попередніх цін.

Потрібно дізнатися про інтереси користувача? Тоді потрібно проаналізувати його лайки, збережені пости чи групи. І необхідно пам'ятати, що даних потрібно якомога більше. Але це тільки частина проблеми.

Насправді, на розробку дійсно хорошого датасету при найсприятливіших умовах можуть піти місяці, а іноді і роки. Дані збирають або вручну (процес довгий, даних менше, однак це гарантує збір більш якісних даних і менше помилок після тренування), або автоматично - просто дають на вхід машині все, що знайшли за запитом.

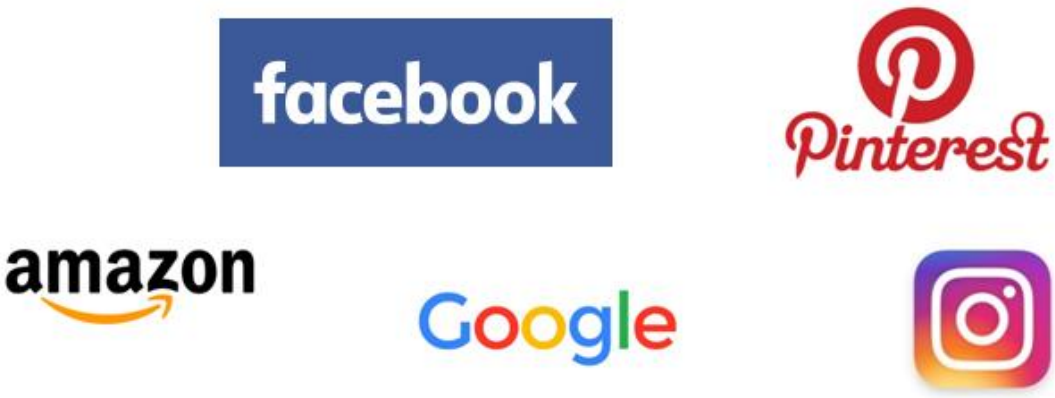

Рис.3.1

Великі компанії (Google, Facebook) використовують своїх користувачів для безкоштовної розмітки (наприклад, ReCaptcha, яка просить користувача знайти на фотографії всі дорожні знаки (рис.3.2)).

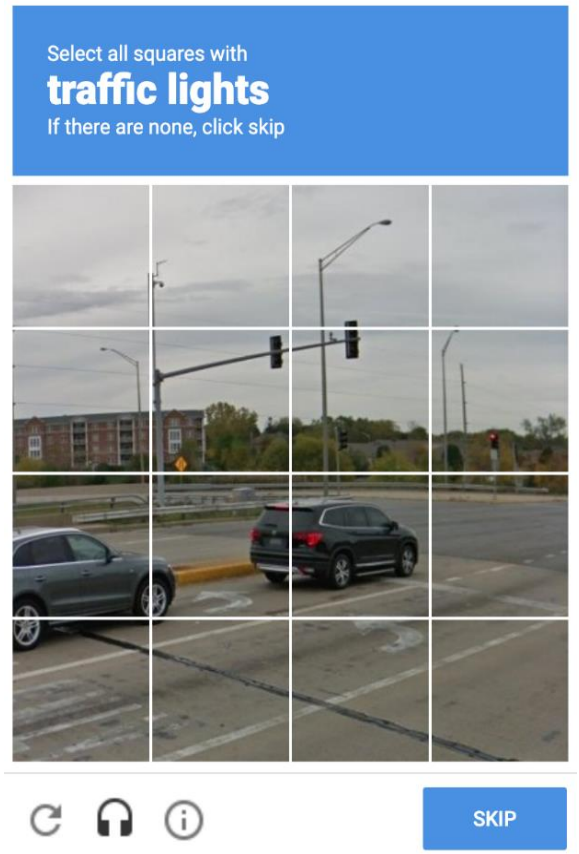

Рис.3.2 - ReCaptcha

Великі компанії можуть собі дозволити розкрити «таємниці» своїх алгоритмів, щоб заохотити розробників до вивчення і покращення, але свої датасети - вкрай рідко.

2) Ознаки

Розробники називають це властивостями, характеристиками – ними можуть бути пробіг мотору автомобіля, вік користувача, ціна акцій, частота появи слова в тексті. Машині потрібно знати, що їй конкретно треба шукати.

3) Алгоритм

Зазвичай, одну й ту ж задачу майже завжди можна розв'язати різними методами. Варто зазначити, що від вибору методу залежить точність, швидкість

роботи і розмір готової моделі. Проте, якщо дані - «сміття», тоді навіть найшвидший і найточніший алгоритм не зможе допомогти побудувати хорошу модель. Тому DataScience-спеціалісти радять збирати якомога більше різноманітних даних.

3.1 Класичне машинне навчання

Перші алгоритми були винайдені в кінці 50-х років минулого століття за допомогою чистої статистики. Вони шукали закономірності в числах, оцінювали близькість точок в просторі і вираховували напрямки.

Класичне навчання можна поділити на дві групи:

- навчання з вчителем;
- навчання без вчителя.

У першому випадку у машини є «хтось», хто допомагає правильно визначати, тобто «вчитель» вже заздалегідь знає всі властивості даних, і машина навчається на конкретних прикладах. У випадку навчання без вчителя машині дають на вхід велику кількість «нерозмічених» даних, і вона намагається сама знайти закономірності.

3.1.1 Навчання з вчителем

Навчання з вчителем (Supervised Learning) ділиться на два типи: класифікація (передбачення класу об'єкта), і регресія (передбачення місця на числовій прямій). Сьогодні класифікацію використовують для:

1) спам-фільтрів;
2) визначення мови;
3) пошуку схожих документів;

4) аналізу тональності текстів;
5) розпізнавання рукописних букв і цифр;
6) визначення підозрілих транзакцій;

Популярні алгоритми класифікації:

1) Наївний Байєс
2) Дерева Рішень,
3) Логістична Регресія,
4) Метод K-найближчих сусідів,
5) Метод Опорних Векторів.

Регресію використовують для:

1) прогнозу вартості цінних паперів
2) аналізу попиту, обсягу продажів
3) медичних діагнозів

Популярні алгоритми регресії:

1) Лінійна регресія
2) Логістична регресія
3) Поліноміальна регресія
4) Ridge регресія (гребенева регресія)
5) Регресія LASSO
6) Elastic Net регресія (регресія еластичної мережі)

3.1.2 Навчання без вчителя

Навчання без вчителя (Unsupervised Learning) було винайдено у 90-ті, і на практиці використовується рідше.

Цікавий тип з цієї сфери є кластеризація - це класифікація, але без заздалегідь відомих класів. Машина сама вирішує, як краще розділяти об'єкти за невідомою ознакою. Вона сама шукає схожі об'єкти та об'єднує їх в кластери. Кількість кластерів можна задати самому або довірити це машині. Кластеризацію використовують для:

1) сегментації ринку (типів покупців, лояльності);
2) об'єднання близьких точок на карті;
3) стиснення зображень;
4) аналіз і розмітки нових даних;
5) детектори аномальної поведінки.

Популярні алгоритми:

1) Метод K-середніх;
2) Mean-Shift;
3) DBSCAN

Також виділяють тип - Зменшення Розмірності (Узагальнення). Такий тип збирає конкретні ознаки.

Узагальнення використовують для:

1) рекомендаційних систем;
2) красивих візуалізацій;
3) визначення тематики та пошуку схожих документів;
4) аналіз фейковий зображень;
5) ризик-менеджменту.

Алгоритми цього типу:

1) Метод головних компонент (PCA);
2) Сингулярне розкладання (SVD);
3) Латентне розміщення Діріхле (LDA);

4) Латентно-семантичний аналіз (LSA, pLSA, GLSA), t-SNE (для візуалізації)

Існує ще такий тип навчання без вчителя, як пошук правил (асоціація).

Його використовують для:

1) аналізу товарів, які купують разом;
2) прогнозу акцій і розпродажів;
3) розташування товарів на полицях;
4) аналізу патернів поведінки на веб-сайтах.

В цю множину входять методи аналізу продуктових кошиків, стратегій маркетингу і тд.

Алгоритми цього типу:

1) Apriori;
2) Eclat;
3) FP-growth

Більш складний тип навчання без вчителя – навчання з підкріпленням. Його застосовують там, де задача полягає не в аналізі даних, а у виживанні в реальному середовищі. Це використовується для:

1) автопілотів автомобілів
2) роботів-порохотягів
3) ігор
4) автоматизованої торгівлі
5) управління ресурсами підприємств

Популярні алгоритми:

1) Q-Learning;
2) SARSA;
3) DQN;

4) АЗС,
5) Генетичний Алгоритм

3.1.3 Ансамблі класифікаторів

Ансамбль класифікаторів – це множина класифікаторів, рішення яких комбінуються певним чином для фінальної класифікації спостереження [4]. Ансамблі класифікаторів використовують для:

1) всіх задач, які вирішують класичними алгоритмами (але тут це точніше);
2) пошукові системи;
3) розпізнавання об'єктів.

Є три перевірених способи створення ансамблів:

1) стекінг – послідовність кількох різних алгоритмів, де їх результати передають на вхід останньому, який приймає остаточне рішення.
2) бустинг – метод, що полягає у зважуванні спостережень навчальної вибірки. Ідея бустингу: класифікатор ансамблю будується послідовно і на кожній ітерації відбувається корекція (переважування) спостережень навчальної вибірки (на першій ітерації вага всіх спостережень є рівною). Корекція здійснюється таким чином, щоб існуючий класифікатор робив менше помилок на тих спостереженнях, на яких часто робили помилки класифікатори, що побудовані на попередніх ітераціях алгоритму. Крім цього, кожному класифікатору присвоюється певна вага, яка визначається з кількості помилок при класифікації.
3) бенінг (Bootstrap AGGregatING) – метод, що полягає в навчанні одного алгоритму багато разів на випадкових вибірках з вихідних даних. В кінці усереднюємо відповіді.

Найпопулярніший приклад бенінга - алгоритм Random Forest. Він застосовується в камері на телефоні (рис. 3.3) (окреслення облич людей в кадрі жовтими прямокутниками). Нейромережа буде занадто повільна в реальному часі, а бегінг тут підходить краще, адже він вміє обчислювати свої дерева паралельно.

Рис.3.3 – Приклад роботи бенінга під час зйомки камери

3.1.4. Глибоке навчання

Глибоке навчання - це популярний підхід до машинного навчання у сфері штучного інтелекту. Його використовують для створення моделей сприйняття та розуміння великої кількості інформації, таких як зображення та звук. В основному, цей підхід ґрунтується на глибоких архітектурах, які є більш структурно складними штучними нейронними мережами (ШНМ).

Для реалізації алгоритмів глибокого навчання потрібна велика кількість даних і апаратних засобів з високою обчислювальною потужністю. Прогрес в області паралельних обчислювальних потужностей на основі графічних процесорів(GPU) особливо сприяв розвитку глибокого підходу до машинного

навчання. В даний час цей тип навчання має багато областей застосування (автоматичне розпізнавання мови, розпізнавання зображень і обробка природної мови). Існує багато різних типів архітектур глибокого навчання. Основні архітектури глибокого навчання:

4) Глибока нейронна мережа;
5) Глибока мережа переконань;
6) Згорткова нейронна мережа;
7) Глибока ядрова мережа;
8) Машина Больцмана;
9) Автокодувальник;
10) Глибока мережа кодування

3.2 Штучні нейронні мережі

Нервова система людини є дуже складною, вона побудована з нейронів, де кожен нейрон (рис.3.4) володіє такими властивостями, як прийом, обробка і передача сигналів по нервових шляхах.

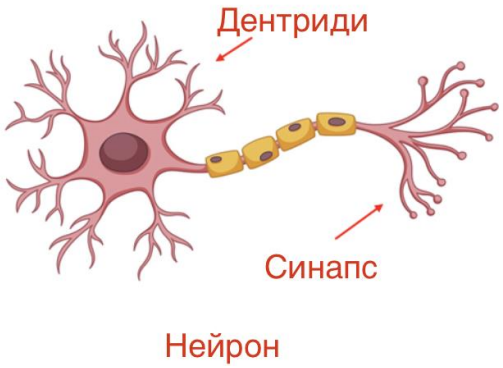

Рис. 3.4 – Біологічний нейрон

Штучні нейронні мережі були розроблені у процесі вивчення головного мозку людини. Нейрон називають базовою обчислювальною одиницею в штучних нейронних мережах. Структура штучного нейрона показана на рис.3.5.

Щоб краще зрозуміти будову нейрону, варто уявити його, як функцію з кількома входами і одним виходом. Завдання нейрона - взяти значення зі своїх входів, виконати над ними функцію і віддати результат на вихід.

Найпростіший приклад нейрона: знайти зважену суму - суму всіх добутків із входів, і, якщо їх сума більше якогось порогового значення, тоді видати на вихід одиницю, інакше - нуль.

Зв'язки між нейронами – це канали, через які ті передають один одному значення. Кожен зв'язок має свою вагу. Коли через зв'язок з вагою 0,7 передається число 10, воно перетворюється в 7 (відбувається операція множення). За допомогою ваг можна підказувати нейрону, на які входи реагувати, а на які – ні [20].

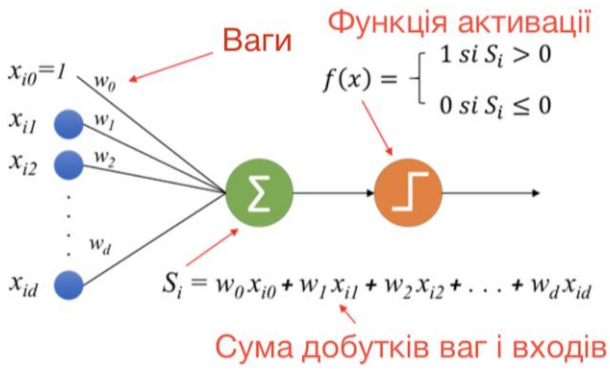

Рис 3.5 – Структура штучного нейрона

Сама ж нейронна мережа складається з 3 основних частин:

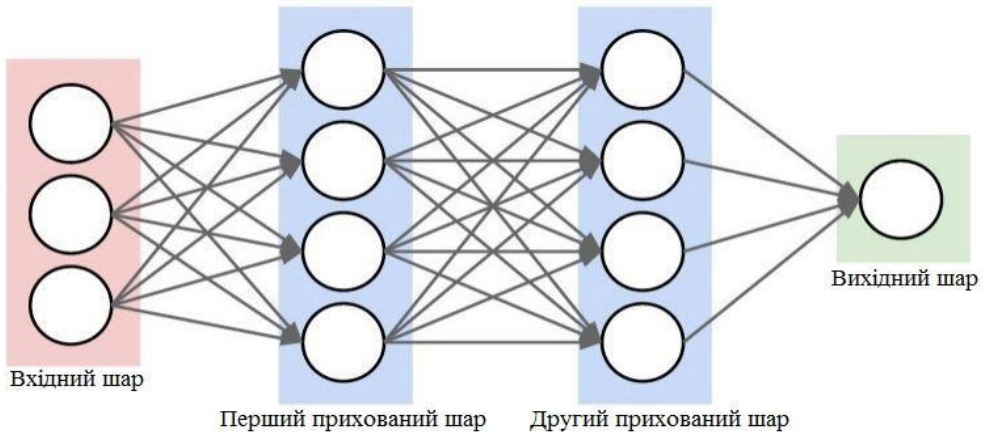

Рис 3.6 – Будова найпростішої нейронної мережі

1) Вхідний шар

Це шар, який вводить інформацію у нейронну мережу для аналізу. Кожне коло представляє 1 частинку (одиницю) інформації. Це може бути що завгодно. Це може бути квадратний метр будинку для програми прогнозування ціни на будинок, або значення пікселя на екрані для програми комп'ютерного зору.

2) Приховані шари

Ці шари виконують всю обробку для нейронних мереж. Ви можете мати стільки, скільки завгодно. Взагалі кажучи, чим більше шарів у вас є, тим точнішою буде нейронна мережа. Кожен шар складається з вузлів, які імітують нейрони нашого мозку.

3) Вихідний шар

Цей шар просто об'єднує інформацію з останнього прихованого шару мережі для виведення всієї необхідної інформації з програми.

Функція активації визначає вихідний сигнал, який нейрон генерує у відповідь на цей вхід, обробляючи чистий вхід, що надходить до нейрона. Сформований вихід направляється у зовнішній світ або в інший нейрон. Функція активації зазвичай є нелінійною функцією. Метою функції активації є перенесення нелінійності на вихід нейрона.

3.2.1 Функції активації

Функція активації вирішує, чи слід активувати нейрон чи ні, обчислюючи зважену суму і надалі додаючи з нею зміщення. Метою функції активації є введення нелінійності у вихід нейрона [19].

Нейронна мережа складається з нейронів, які працюють відповідно до ваги, зміщення та відповідної функції активації. У нейронній мережі ваги та зміщення нейронів оновлювлюють на основі помилки на виході. Цей процес

відомий як метод зворотного поширення помилки (back-propagation algorithm). Функції активації якраз таки роблять можливим зворотне поширення помилки, оскільки градієнти подаються разом із помилкою для оновлення ваг та зміщень.

Нейронна мережа без функції активації є по суті лише моделлю лінійної регресії. Функція активації виконує нелінійне перетворення на вхід, що робить його здатним вчитися та виконувати більш складні завдання.

Є багато функцій активації: лінійна (рис. 3.7), ступінчаста (рис. 3.8), сигмоподібна (рис. 3.9), гіперболічна дотична (tanh – рис. 3.10), ректифікована лінійна одиниця (ReLU – рис. 3.11) та інші.

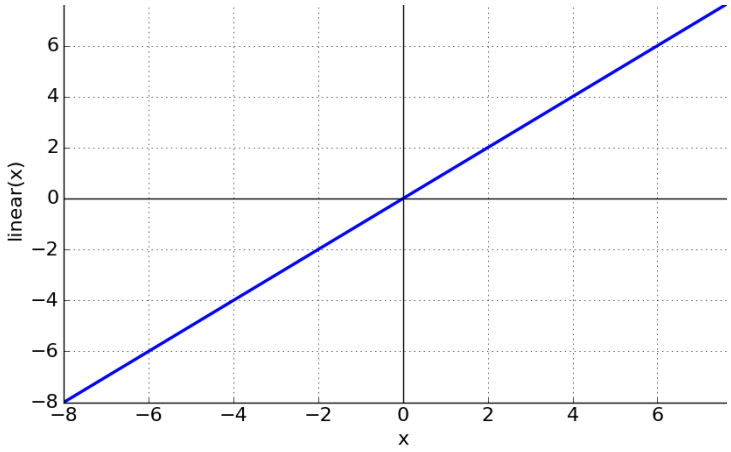

Рис. 3.7 – Лінійна функція активації

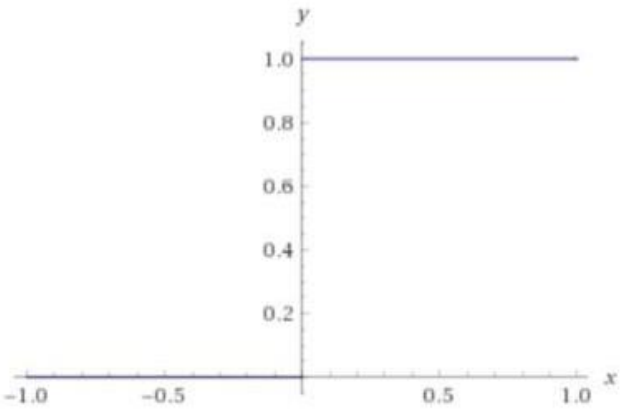

Рис. 3.8 – Ступінчаста функція активації

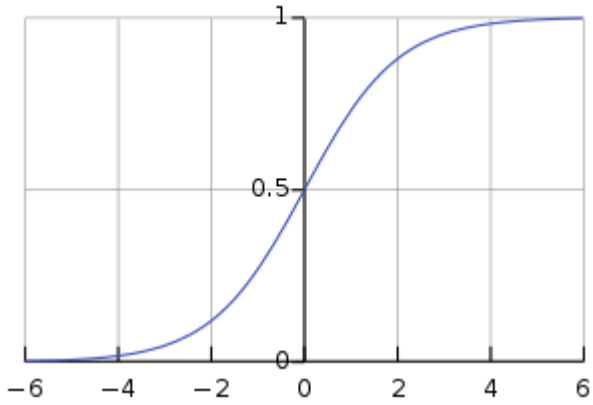

Рис. 3.9 - Сигмоподібна функція активації

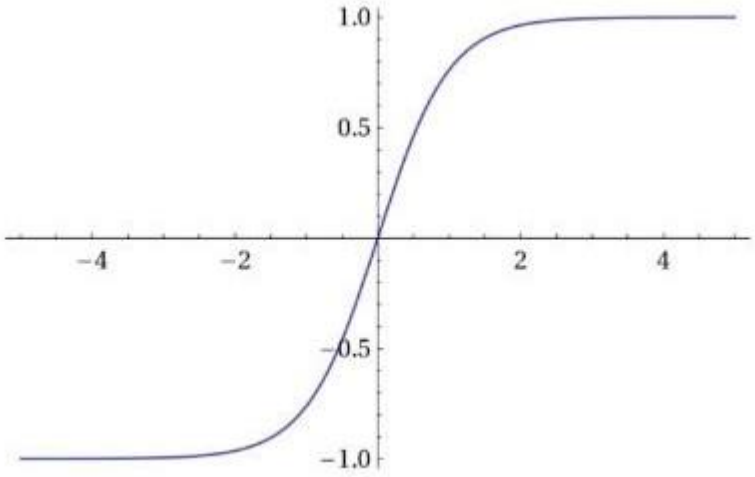

Рис. 3.10 - Гіперболічна дотична функція активації

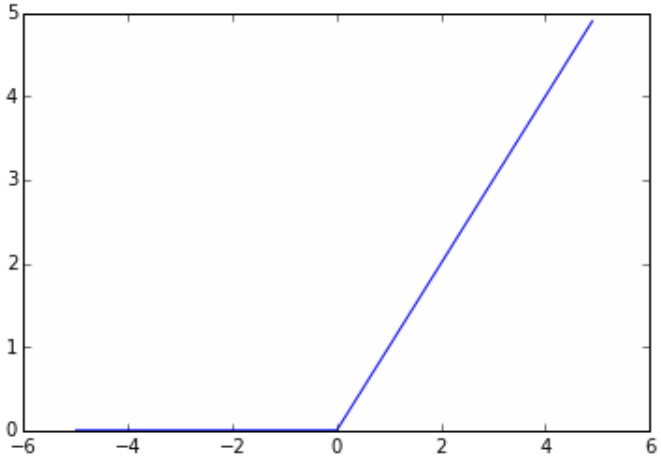

Рис. 3.11 - Функція активації ReLU

Функція активації ReLU (рис. 3.7) $f(x) = max(0, x)$, генерує висновок з порогом 0 для кожного з вхідних значень . Ця функція є дуже популярною, саме її було використано при реалізації нейронної мережі.

4 Згорткова нейронна мережа

Сьогодні згорткові мережі стають все більш популярними серед глибоких методів навчання, оскільки вони можуть успішно вивчати моделі для багатьох комп'ютерних та візуальних програм, таких як виявлення об'єктів, розпізнавання об'єктів та сегментація семантичного зображення.

Згорткова нейронна мережа, запропонована компанією LeCun в 90-тих роках для вирішення задачі Computer Vision, є типом багатошарового подання штучних нейронних мереж.

Всі нейрони в шарі в прямих штучних нейронних мереж з'єднані з усіма нейронами наступного шару. Такі пов'язані шари називаються повністю пов'язаними шарами.

Основний шар мережі – згортковий шар, його метою є згортання вхідного зображення за допомогою фільтрів і вилучення особливостей (ознак) зображення. Остаточний вихід із згоркового і об'єднувального шарів передається в один або більше повністю з'єднаних шарів. Проста архітектура згоркової нейронної мережі являє собою комбінацію згорткових, об'єднуючих і повністю пов'язаних шарів (рис.4.1)

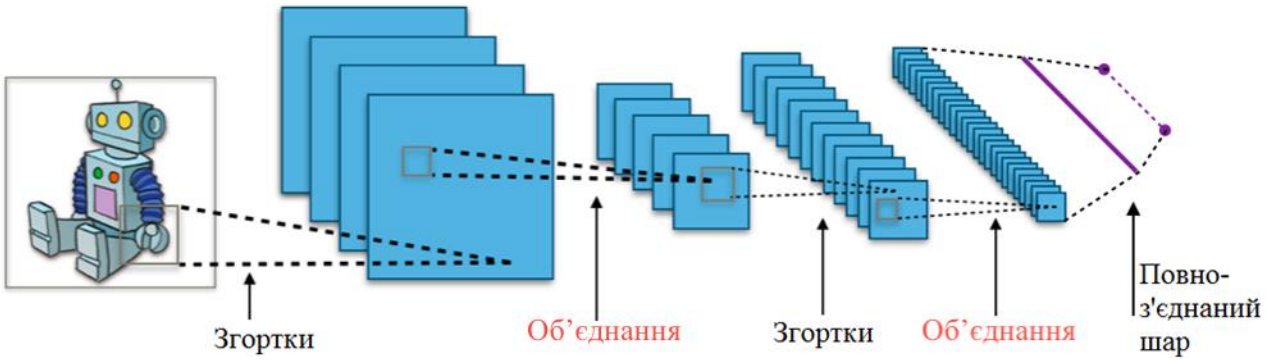

Рисунок 4.1 – Приклад простої архітектури згорткової нейронної мережі

Вхідне зображення згортається з кожним навченим фільтром, що використовується в цьому шарі, для генерації різних карт ознак. Карти ознак стають більш нечутливими до обертання і спотворень, і, отримані в згортковому шарі, піддаються шару об'єднання для зменшення просторової розмірності і збереження важливих особливостей (ознак). Саме прогнозування виходу отримують з класифікатора у повнозв'язному шарі. Класифікатор є кінцевим шаром для генерування значень ймовірності класу як вихідного.

Комп'ютери сприймають вхідне зображення, як масив пікселів. Залежно від роздільної здатності, зображення матиме розмірність h x w x d (h = висота, w = ширина, d = розмір). Наприклад, зображення масиву 6 x 6 x 3 матриці RGB (3 відноситься до значень RGB) є 3D-матрицею (рис. 4.2) [12].

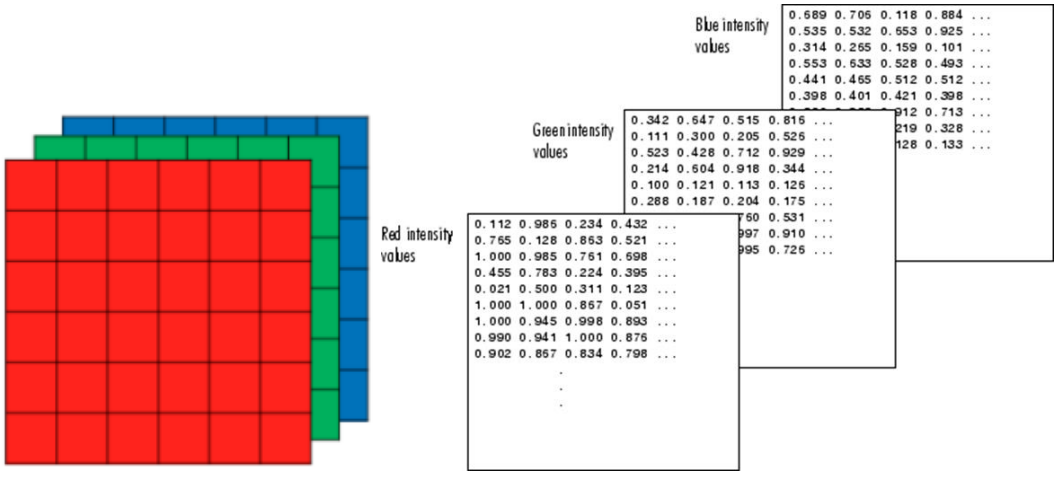

Рис.4.2

4.1 Шар згортки

Шар згортки – це перший шар, який виділяє карти ознак із вхідного зображення. Цей шар зберігає взаємозв'язок між пікселями, вивчаючи зображення, використовуючи невеликі квадрати з вхідних даних. Це математична операція, яка приймає два входи: матриця зображення та фільтр [12].

Розглянемо бінарне зображення (значення пікселів зображення – 0 або 1) розмірності 5x5, і матрицю фільтру 3x3, як показано нижче на рис. 4.3.

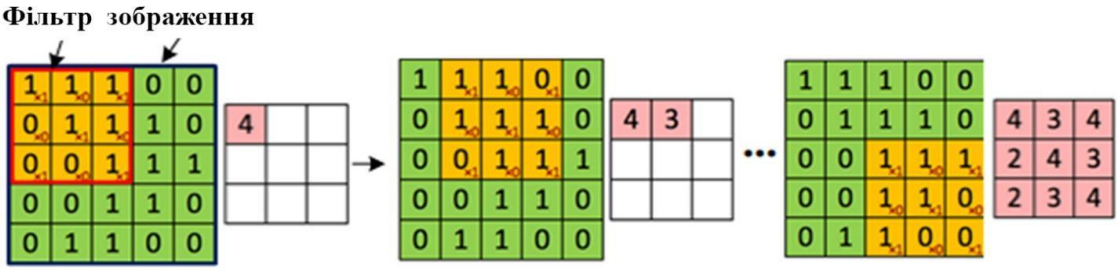

Рис.4.3

Добуток матриці зображення 5x5 і фільтру 3x3 називається картою ознак (заповнений рожевий квадрат на рис. 4.3).

Згортку зображення можна виконати за допомогою різних фільтрів: виявлення країв, розмиття, різкість, тощо. Нижче показано різні згортки зображень після застосування різних фільтрів [11, 12].

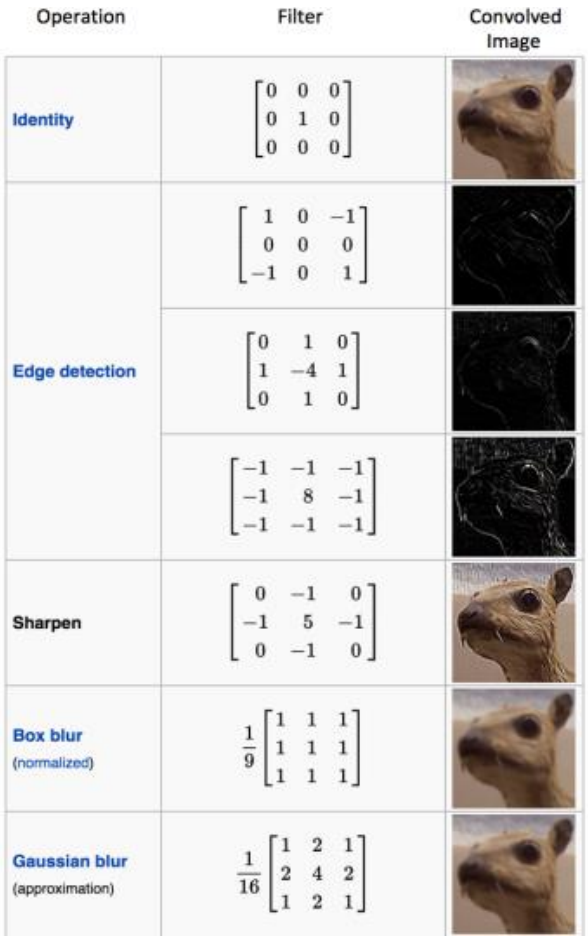

Рис.4.4 – Приклади застосування фільтрів

Іноді фільтр не зовсім підходить для вхідного зображення.

Є два варіанти, щоб вирішити проблему:

1. Заповнити зображення нулями (заповнення нулями), щоб воно відповідало фільтру.
2. Видалити частину зображення, де фільтр не підходить. Це називається допустимим заповненням, яке зберігає тільки правильну частину зображення.

Як зазначено раніше, наша згортка відбувається шляхом «ковзання» фільтру по вхідній матриці. Процес «ковзання» відбувається завдяки зсуву фільтра по пікселях. Такі зсуви називають кроками [12]. Коли крок дорівнює 1, ми переміщуємо фільтри на 1 піксель. Коли крок дорівнює 2, ми переміщуємо фільтри на 2 пікселі. На рис. 4.5 нижче показано, що згортка працює з кроком 2.

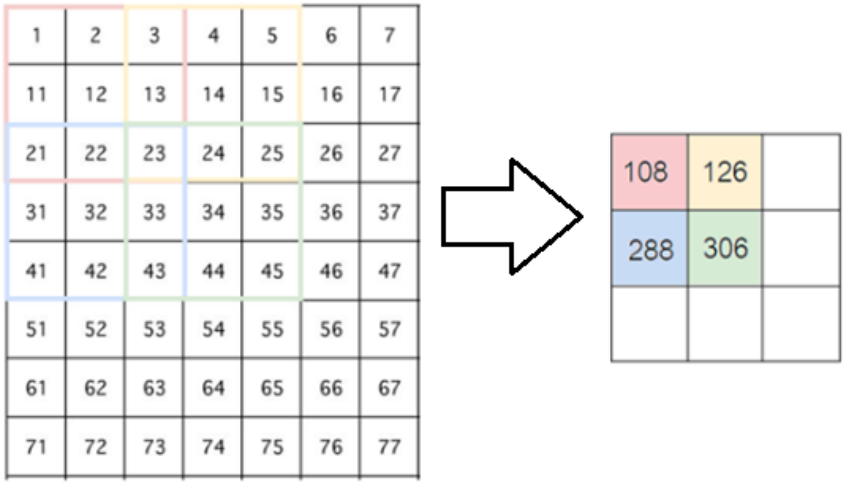

Рис. 4.5 – Операція згортки з кроком 2

Після згортки виконується активаційна функція ReLU. Метою ReLU є впровадження нелінійності в нашу згорткову мережу, оскільки вхідні дані, які нейронна мережа намагається вивчити, зазвичай нелінійні. Якби не використовувалась ReLU чи будь-які інші функції нелінійної активації, мережа була б великим лінійним класифікатором і могла б бути спрощена простим множенням вагових матриць (з урахуванням зміщення). А це унеможливлює класифікацію зображень чи передбачення тексту.

Нижче можна побачити приклад застосування ReLU над картою ознак (рис. 4.6)

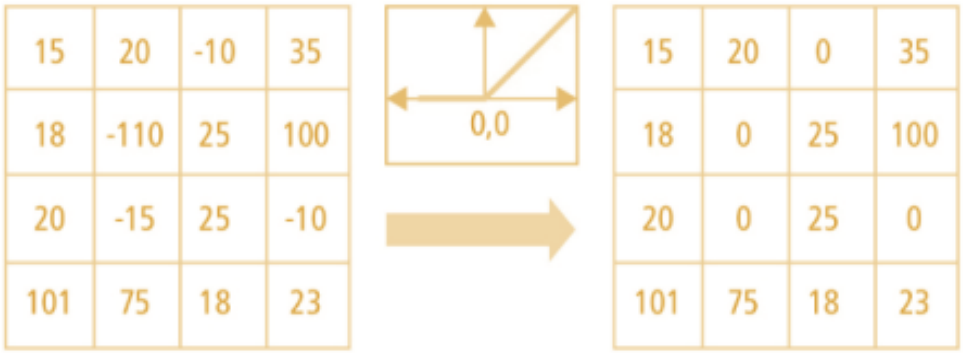

Рис. 4.6 - Застосування ReLU над картою ознак

На рис. 4.7 можна побачити, як виглядатиме карта ознак після першої згортки. Негативні входи стають нулями, позитивні входи передаються без змін. ReLU операція виконується окремо для кожного значення пікселя. Вважають, що чорні області представлені негативними пікселями, а білі області представлені позитивними пікселями у вхідній карті об'єкта [12].

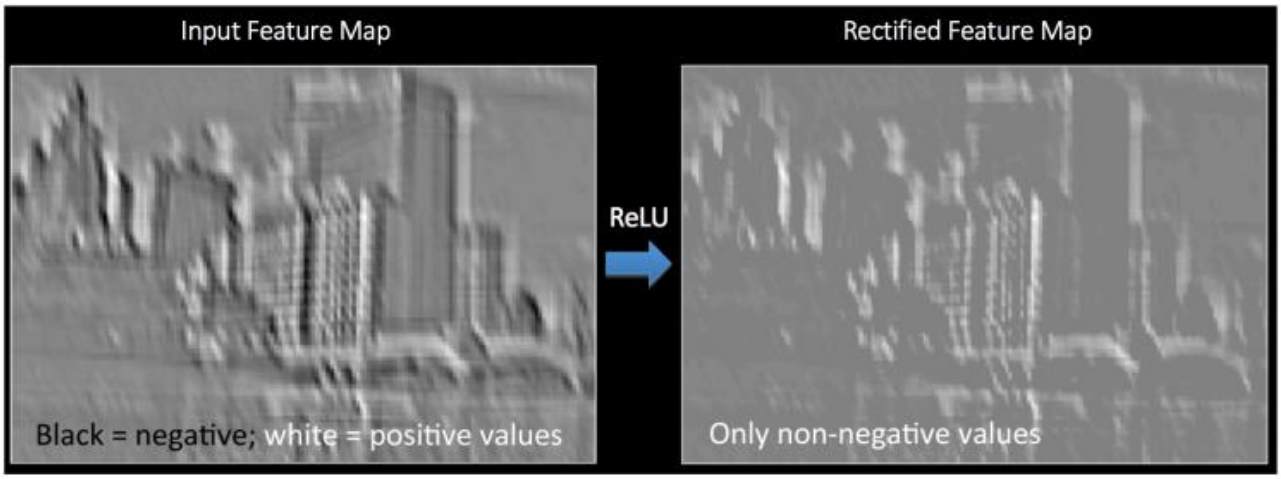

Рисунок 4.7 – Приклад операції ReLU

4.2 Пулінг

Пулінг або операція об’єднання зменшує розмірність кожної карти ознак, але зберігає важливу інформацію. Просторовий пул може бути різних типів:

1) Максимальний пул
2) Середнє пул
3) Пул суми

Для розробки нейронної мережі використано максимальний пул (об’єднання), що бере найбільший елемент із карти ознак. На рис.4.8 показана операція об'єднання, що виконується з макс-пулінгом шляхом переміщення фільтра 2×2.

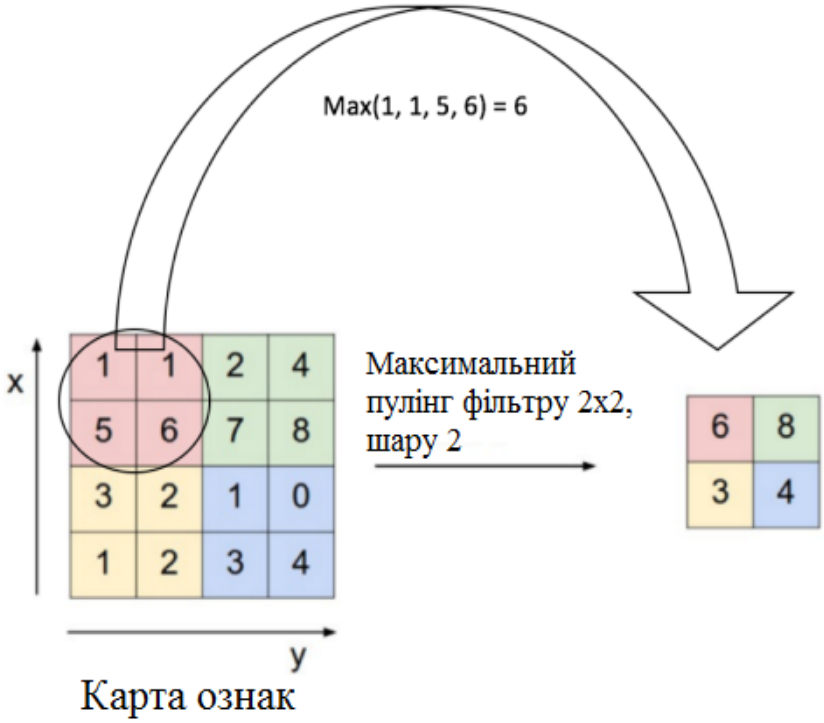

Рисунок 4.8 – Операція об'єднання, крок – 2

Після пулінгу карта ознак зменшується вдвічі (рис. 4.9)

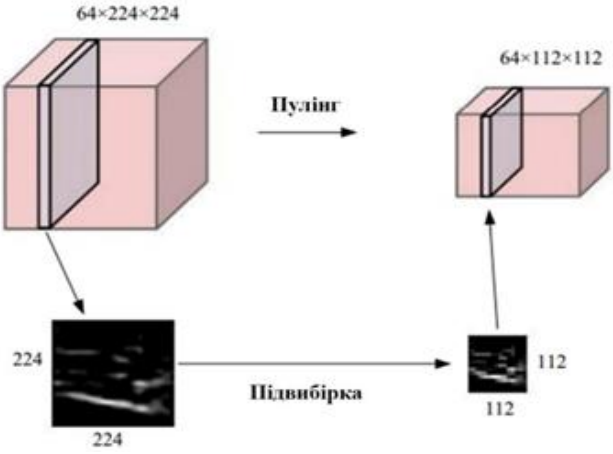

Рис. 4.9

Після згортки, пулу і ReLU шарів отримують матрицю карти об’єктів у вигляді вектора. Його передають у повністю зв'язний шар, що розміщується в кінці мережі (рис. 4.1). Нейрони в цьому шарі повністю залежать від усіх активацій у попередньому шарі. Найбільш важливою особливістю повністю зв'язного шару є те, що він дозволяє нейронам у цьому шарі визначити, які ознаки відповідають яким класифікаціям. Тут застосовують функцію активації, таку як softmax або sigmoid, щоб класифікувати виходи.

Наприклад, Softmax функція:

$$g(p) = \begin{cases} 1, \text{якщо } p \geq 0.5 \\ 0, \text{якщо } p < 0.5 \end{cases}$$

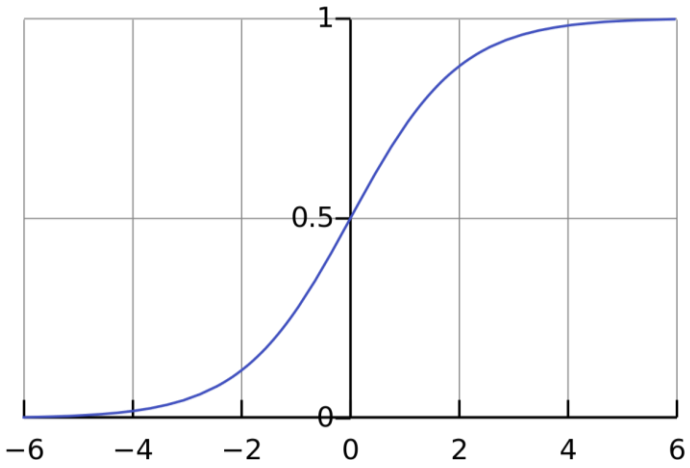

Рис. 4.10 - Softmax функція

4.3 Модельні архітектури

На сьогоднішній час існує багато різних модельних архітектур [9, 16]. Одна із найпростіших і найпопулярніших архітектур є Fully Convolutional Network (FCN – архітектура) (рис. 4.10). Тут відбувається зменшення вхідного зображення до меншого розміру через серію згорток.

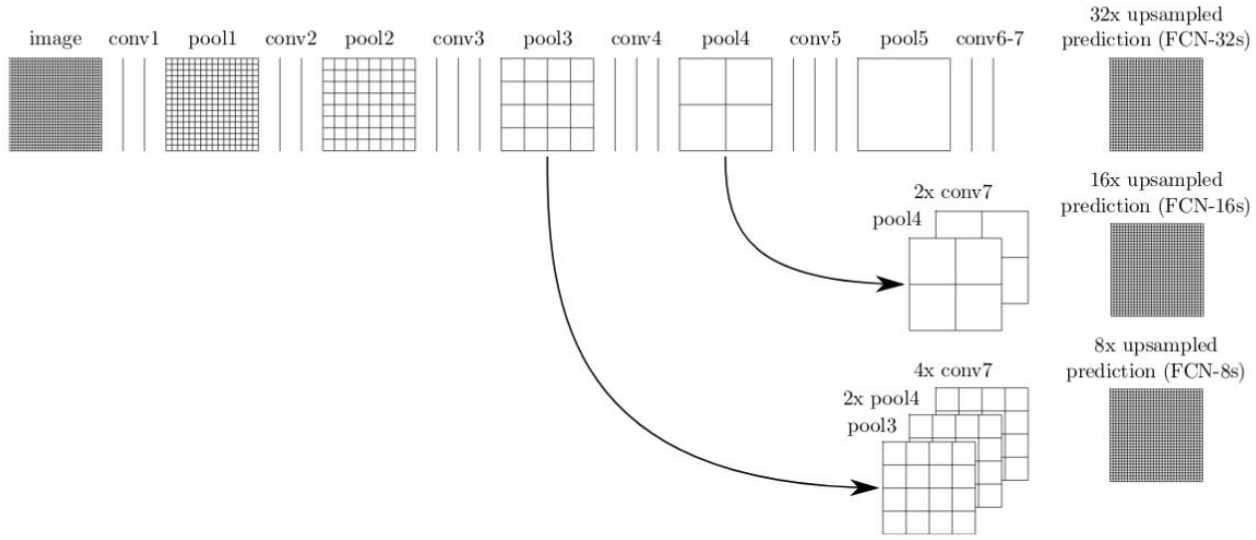

Рис. 4.11 – FCN архітектура

Архітектура SegNet [14] складається з послідовності нелінійних шарів обробки (кодерів) та відповідного набору декодерів, за якими слідує піксельний класифікатор. Як правило, кожен кодер складається з одного або декількох згорткових шарів із пакетною нормалізацією та нелінійною активаційною функцією ReLU з подальшим макс-пулінгом. Розріджене кодування внаслідок процесу об'єднання об'єднується в декодер, використовуючи індекси максимального об'єднання в послідовності кодування (рис. 4.12)

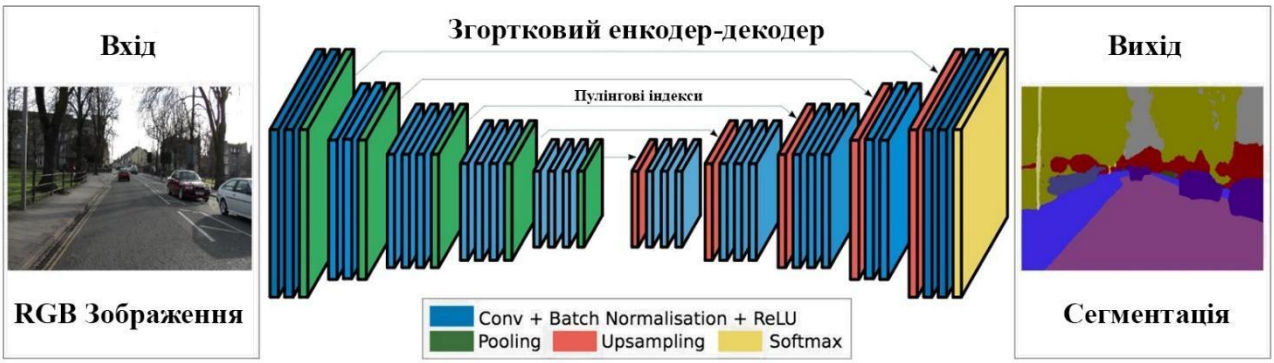

Рисунок 4.12 – Архітектура SegNet

4.3.1 U-net

U-net архітектура є одним з яскравих представників сегментаційних моделей нейронних мереж. Вперше заявлена, як мережа для сегментації біологічних знімків, вона добре зарекомендувала себе, як основа при вирішенні практичних завдань сегментації. Саме цю архітектуру використано для реалізації задачі.

Це особливий вид CNN, який використовує об'єднані шари для передачі даних з попередніх шарів до шарів, близьких до вихідних даних. Мережа має форму подвійної лійки [21]. Переваги використання U-Net:

- Обчислювально ефективна
- Тренування з невеликим набором даних

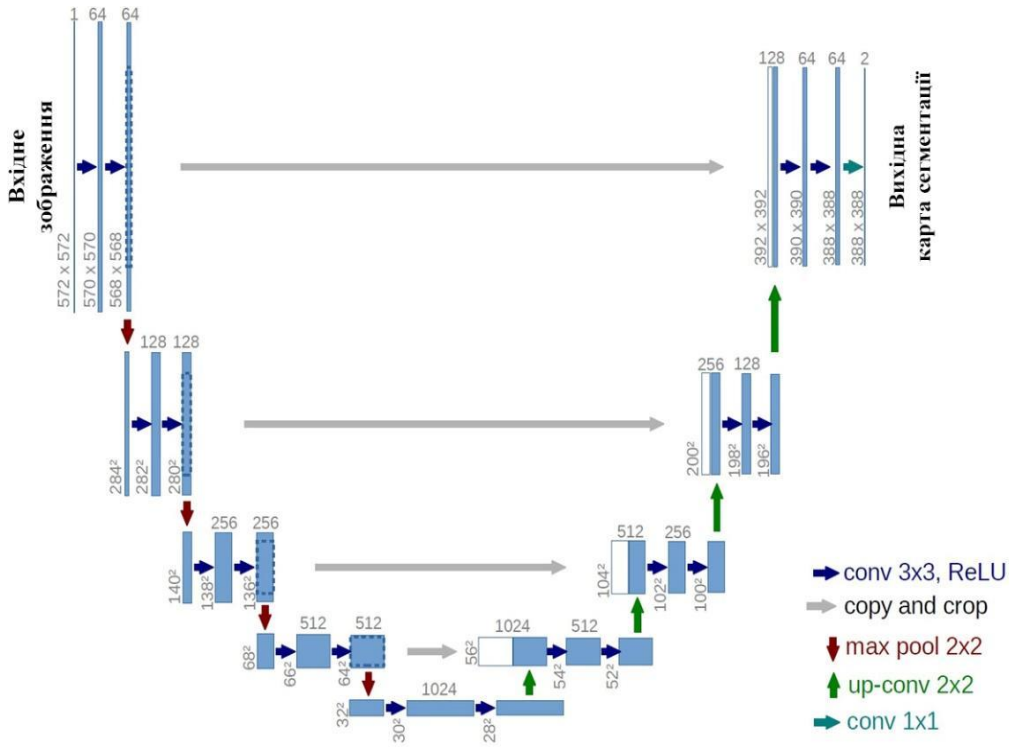

Рис. 4.13 - U-net архітектура

Кожен синій прямокутник відповідає багатоканальній карті ознак. Стрілки позначають різні операції.

Архітектура мережі виконана у формі «лійки», містить дві частини, де одна частинка звужується (енкодер), а інша - розширюється (декодер). Перша являє собою типову архітектуру згорткової класифікаційної нейронної мережі. Складається з повторюваних застосувань двох згорток з фільтрами 3x3, за якими слідують активація (Relu) і операція макс-пулінга 2x2 з кроком 2 (позначено стрілками на рис. 4.13).

Розширювана частина складається з кроків деконволюції (зворотної згортки) 2x2, зменшує кількість каналів, потім конкатенація з обрізаними картами ознак від відповідної частини (рис. 4.13 - білі прямокутники представляють скопійовані карти об’єктів), дві згортки 3x3 і активація (ReLU). На виході виконується згортка 1x1 для зіставлення класу кожному 64-компонентному вектору ознак.

5 Програмна реалізація

Для реалізації необхідно здійснити імпорт необхідних бібліотек:

1. Numpy - для лінійної алгебри
2. os - для отримання інформації про файли
3. matplotlib - для візуалізації
4. PIL та CV2 - для завантаження та редагування зображень
5. scikit-learn KMeans - для кластеризації кольорів
6. random - для генерації випадкових чисел та вибору
7. keras - інтерфейс високого рівня для tensorflow(для нейронних мереж)

Функція LoadImage() завантажує зображення із шляху до файлу, а також дозволяє прості варіанти збільшення даних, такі як перевертання та обертання. Зображення в даному випадку - це паралельні зображення вхідного зображення та кольорово сегментоване зображення. Функція LoadImage() також розділяє зображення на окремі зображення.

Для тренування мережі було використано датасет CityScapes [15], тут наявні 4000 фото для тренування і 500 фото для валідації.

Рис. 5.1

Кластеризація кольорів

Завдяки стисненню стандарту JPEG на сегментованому зображенні є лише кілька дискретних кольорів. Щоб знайти найважливіші кольори та визначити подібні кольори, ми можемо використовувати кластеризацію KMeans. Це також дозволяє нам перейти від кольорового подання до представлення класу.

Конвертація шарів до зображення RGB

Після кластеризації зображення містить інформацію про 13 різних шарів. Подання шару перетворюється на кольорове за допомогою функції LayersToRGB() (це лише для візуалізації).

Конвертація кольорів до класу

ColorToClass() функція перетворює дискретне представлення кольорів (вихід кластеризації кольорів) у 13-вимірне представлення класу. Це корисно для алгоритму машинного навчання пізніше. Внизу можна побачити приклад класифікації, відтворений над вихідним зображенням.

Рис. 5.2

Щоб зробити навчання більш ефективним, використовується генератор для подачі даних в алгоритм глибокого навчання. Цей генератор створює партії,

наприклад, 20 одночасно сегментованих пар зображень. Він також використовує збільшення зображень та випадково перегортає та обертає зображення, щоб збільшити ефективний розмір набору даних. Нижче можна побачити одну таку пару.

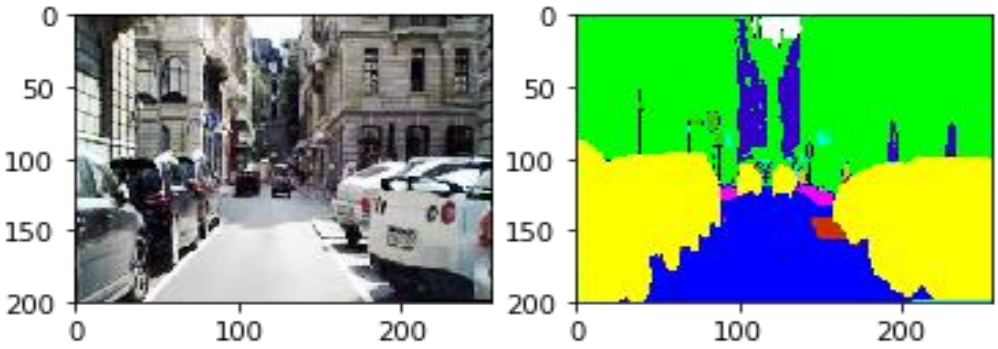

Рис. 5.3

Згорткова нейронна мережа - UNet

Для навчання побудовано мережу типу UNet. Мережа тренується протягом 1000 епох (це займає приблизно до шести годин) та використовує Model Checkpoint (keras), щоб зберегти модель із найменшими втратами при валідації.

6 Результати

Рис. 6.1 – 6.3 - Результати

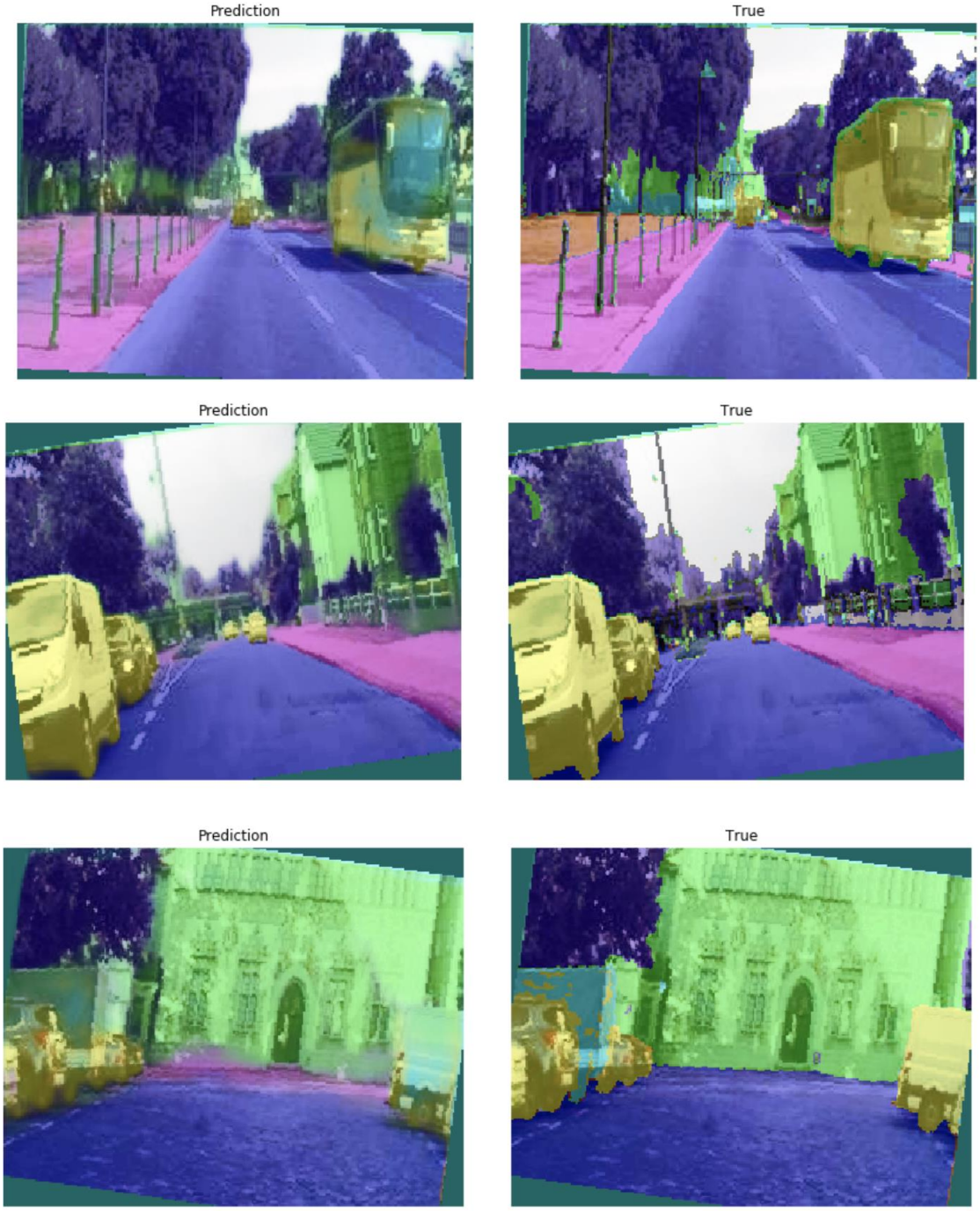

Рис. 6.4 – 6.6 - Результати

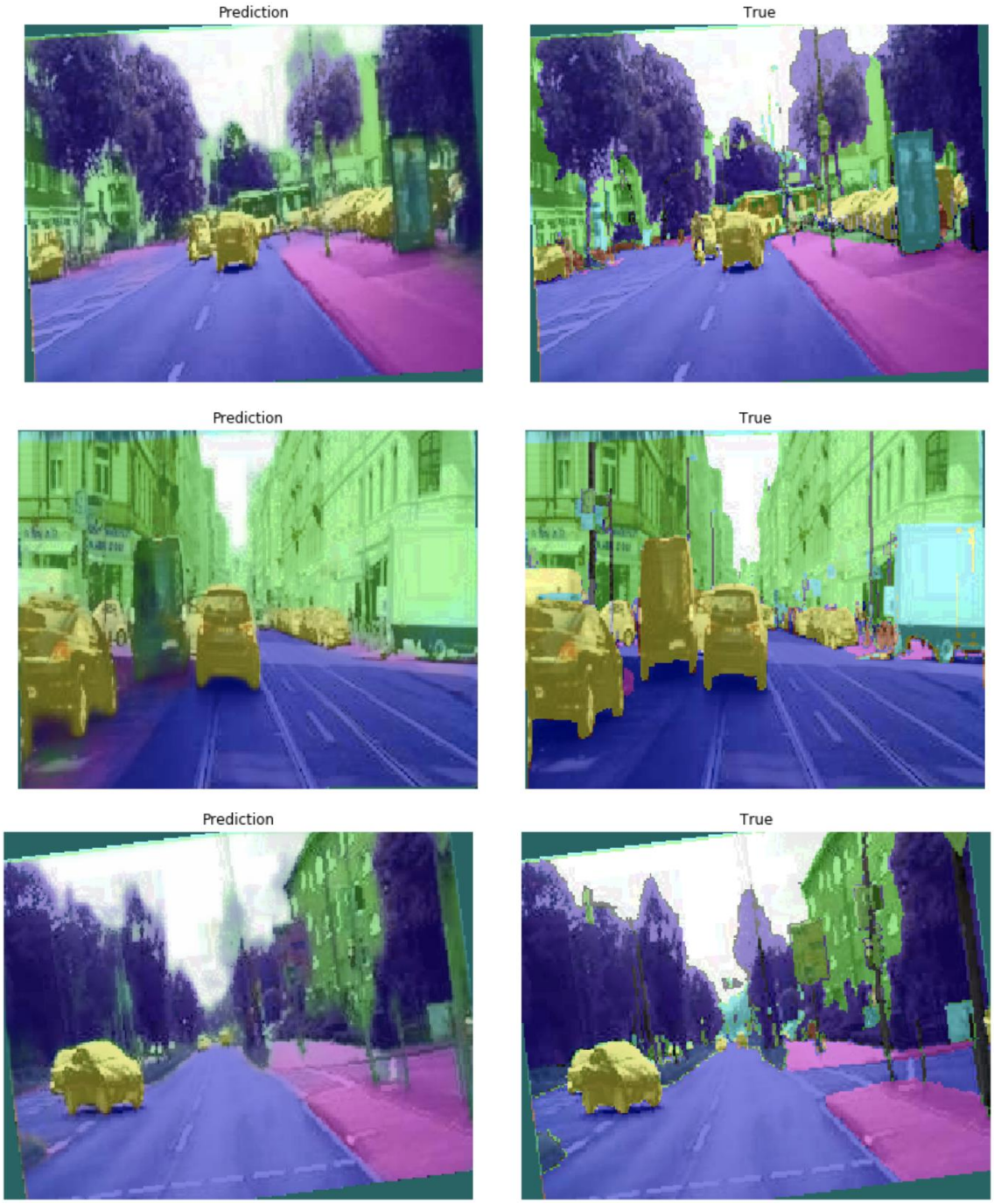

Рис. 6.7 – 6.9 - Результати

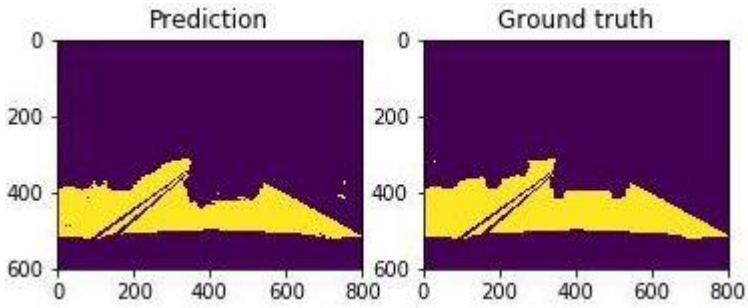

Рис. 6.10 – Попередні результати

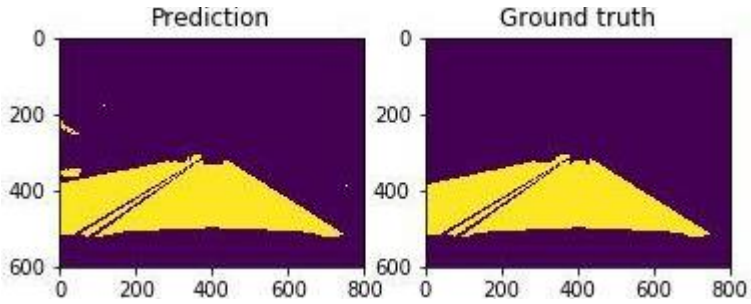

Рис. 6.11 – Попередні результати

Раніше було реалізовано бінарну сегментацію, де був тільки 1 сегментований клас і фон. Зараз вдалось реалізувати повноцінну сегментацію зображення. Нейронна мережа добре сегментує дорогу, тротуари, машини, будинки, але не завжди знаходить пішоходів, дорожні знаки, дерева. Це можна вирішити кращим датасетом, в якому буде більше фото з пішоходами і іншими об’єктами.

Нейронні мережі навчаються за допомогою функції втрат [17]. Це метод оцінює, наскільки конкретний алгоритм добре моделює дані. Поступово функція втрат вчиться зменшувати похибки прогнозування. Було обрано категоріальну перехресну ентропію (Categorical Cross-Entropy loss [18]), як функцію втрат для сегментаційної нейронної мережі. З рис. 6.12 видно, що функція втрат (loss function) поступово спадає і виходить на плато із незначними скачками. Протягом довгого часу (з приблизно 400-тої епохи) сильних змін на графіку нема, тому тренувати далі мережу нема сенсу.

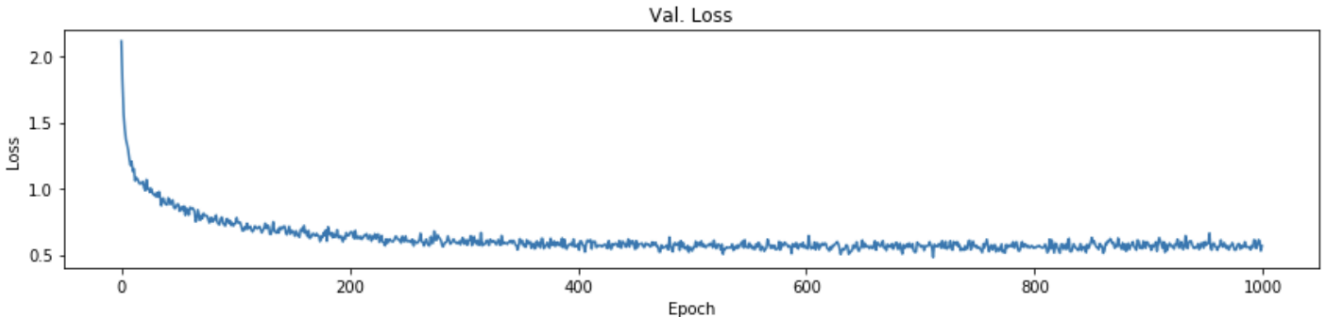

Рис. 6.12

ВИСНОВКИ

Під час виконання магістерської дипломної роботи було досліджено наукові статті і відеоматеріали по нейронним мережам, фреймворки Tensorflow і Keras, згорткові нейронні мережі та їх архітектури. Особливу увагу звернуто на U-Net архітектуру, яку було використано для побудови сегментаційної нейронної мережі.

Було створено згорткову нейронну мережу, яка вміє сегментувати типові об’єкти у місті.

Створена модель може бути використана, як вирішення проблеми «автопілот» для машин. Чим краще натренована мережа, тим простіше і значно безпечніше у навколишньому середовищі.

Під час виконання практичної частини магістерської роботи було вирішено наступні підзадачі:

1. підготовка набору даних
2. розробка U-Net архітектури нейронної мережі
3. тренування нейронної мережі
4. аналіз результатів

Проблеми, які залишились невирішені:

1. Мала кількість даних необхідних для навчання алгоритму.
2. Точність результатів сегментації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Image Segmentation in Deep Learning: Methods and Applications. Електронний ресурс]. – Режим доступу: https://missinglink.ai/guides/computer-vision/image-segmentation-deep-learning-methods-applications/
2. Olaf Ronneberger, Philipp Fischer, and Thomas Brox - U-Net: Convolutional Networks for Biomedical Image Segmentation [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/1505.04597.pdf
3. Машинне навчання простими словами. Частина 1. [Електронний ресурс]. – Режим доступу: http://www.mmf.lnu.edu.ua/ar/1739?fbclid=IwAR3lY-irfgMy9_bl8jT6VTx7n3BVYsLkm1HJQqu_i_4dWnpTGRtk9vkVzY8
4. Машинне навчання простими словами. Частина 2. [Електронний ресурс]. – Режим доступу: http://www.mmf.lnu.edu.ua/ar/1743
5. Sumit Saha: A Comprehensive Guide to Convolutial Newral Network [Електронний ресурс]. – Режим доступу: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
6. Kaymak C. Brief Survey and an Application of Semantic Segmentation for Autonomous Driving [Електронний ресурс]. – Режим доступу: https://arxiv.org/ftp/arxiv/papers/1808/1808.08413.pdf
7. Krizhevsky A., Sutskever I., Hinton Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks // Advances in Neural Information Processing Systems 25 / Ed. by F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger. – Curran Associates, Inc., 2012. – P. 1097–1105.
8. Liaw A., Wiener M. et al. Classification and regression by randomForest // R news. – 2002. – Vol. 2, no. 3. – P. 18–22.
9. Raj B. A simple guide to semantic segmentation [Електронний ресурс]. – Режим доступу: https://medium.com/beyondminds/a-simple-guide-to-semantic-segmentation-effcf83e7e54

10. Satellite Image Segmentation: a Workflow with U-Net [Електронний ресурс]. – Режим доступу: https://medium.com/vooban-ai/satellite-image-segmentation-a-workflow-with-u-net-7ff992b2a56e
11. PULKIT SHARMA Computer Vision Tutorial: A Step-by-Step Introduction to Image Segmentation Techniques [Електронний ресурс]. – Режим доступу: https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/
12. Understanding of Convolutional Neural Network (CNN) — Deep Learning [Електронний ресурс]. – Режим доступу: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148
13. An Intuitive Explanation of Convolutional Neural Networks [Електронний ресурс]. – Режим доступу: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
14. Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation Segmentation [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/1511.00561.pdf
15. CitySpaces dataset [Електронний ресурс]. – Режим доступу: https://www.cityscapes-dataset.com/
16. Aria. Explain fully convolutional network and Unet. [Електронний ресурс]. – Режим доступу: https://zhuanlan.zhihu.com/p/65398511
17. Common Loss functions in machine learning. [Електронний ресурс]. – Режим доступу: https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23
18. Raul Gomes. Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names. [Електронний ресурс]. – Режим доступу: https://gombru.github.io/2018/05/23/cross_entropy_loss/

19. Akanksha Rai. Activation functions in Neural Networks. [Електронний ресурс]. – Режим доступу: https://www.geeksforgeeks.org/activation-functions-neural-networks/
20. Shayan Pal. Implementing Artificial Neural Network training process. [Електронний ресурс]. – Режим доступу: https://www.geeksforgeeks.org/implementing-ann-training-process-in-python/
21. Understanding Unet. [Електронний ресурс]. – Режим доступу: https://bond-kirill-alexandrovich.medium.com/understanding-unet-27de538e08d8
22. GitHub repo – Режим доступу: https://github.com/SofiaGorlata/ImageSegmentation