

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

**НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

Пояснювальна записка
до магістерської роботи
на ступінь вищої освіти магістр
на тему: «**Реалізація інтелектуальної системи розпізнавання об'єктів
із застосуванням нейронних мереж**»

Виконав: студент 5 курсу, групи ПДМ-51
спеціальності:

121 – Інженерія програмного забезпечення

(шифр і назва спеціальності)

Ільюкевич Гліб Олегович

(прізвище та ініціали)

Керівник Дібрівний Олександр Андрійович

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтроль _____

(прізвище та ініціали)

Київ – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти - «Магістр»
Спеціальність – 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри
інженерії програмного забезпечення

_____ 2022 року
“ ____ ” _____

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Ільюкевич Гліб Олегович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Реалізація інтелектуальної системи розпізнавання об'єктів із застосуванням нейронних мереж»

Керівник роботи Дібрівний Олесь Андрійович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “__” 2022 року
№__.

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи:

1. Принципи і методи штучних інтелектуальних систем;
2. Методи пошуку об'єктів на зображенні;
3. Методи інтеграції результатів для локалізації об'єктів;
4. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

4.1 Дослідження актуальності розпізнавання об'єктів на зображеннях

4.2 Аналіз оптимальних методів розпізнавання об'єктів на зображеннях

4.3 Побудова моделі системи розпізнавання об'єкта на зображенні, що дозволяє досліджувати якісні характеристики результату;

4.4 Дослідити вплив параметрів вхідних даних на вибір оптимальної стратегії комбінування результатів розпізнавання об'єктів.

5. Перелік графічного матеріалу

1. Титульний слайд
 2. Мета, об'єкт дослідження, предмет дослідження
 3. Актуальність роботи
 4. Аналіз існуючих ІТ-рішень та моделей
 5. Аналіз існуючих архітектур
 6. Метод Віюлі-Джонсона
 7. Каскад класифікаторів
 7. Дескриптор
 7. Навчання та тестування класифікаторів
 7. Порівняння класифікаторів
 8. Висновки
-

6. Дата видачі завдання

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1.	Вибір теми проекту	01.09.2020 - 31.10.2020	
2.	Постановка задачі	01.11.2020 – 20.11.2020	
3.	Вибір засобу реалізації	21.11.2020 – 10.06.2021	
4.	Розробка архітектури		
5.	Розробка візуальної частини		
6.	Розробка функціональної частини		
7.	Здача диплому в деканат		

Студент _____
(підпис)

Ільюкевич Г.О.
(прізвище та ініціали)

Керівник роботи _____
(підпис)

Дібрівний О. А.
(прізвище та ініціали)

РЕФЕРАТ

Магістрської дипломної роботи Ільюкевича Гліба Олеговича на тему
«Реалізація інтелектуальної системи розпізнавання об'єктів із застосуванням
нейронних мереж»

Текстова частина дипломної роботи 44 с., 34 рис., 30 джерел.

Актуальність дипломної роботи полягає в використанні нейронних мереж для вирішення формалізованих задач в інтелектуальному аналізі даних. Зростання обсягів інформації, а також розширення кола технічно складних задач прийняття рішень вимагають систематизації існуючих методів і розробки нових методик і алгоритмів рішення. У магістерській роботі розглядається можливість застосування нейронної мережі при вирішенні задачі розпізнавання об'єктів на фотографіях і відеозаписах.

У роботі проведено аналіз існуючих проблем при створенні нейронних мереж для розпізнавання об'єктів та запропоновано рішення у вигляді розробленої нейронної мережі.

Об'єкт роботи: створення інтелектуальної системи для розпізнавання об'єктів за допомогою нейронних мереж.

Предмет роботи: нейронна мережа для розпізнавання об'єктів.

Мета роботи: створення варіанту вирішення проблеми розпізнавання об'єктів обчислювальними машинами і інтелектуальними системами за допомогою нейронних мереж.

Ключові слова: інтелектуальна система, нейронна мережа, розпізнавання об'єктів

Зміст

ВСТУП	8
РОЗДІЛ 1. ОСНОВНІ ПОНЯТТЯ НЕЙРОНИХ МЕРЕЖ ТА ШТУЧНИХ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ	10
1.1. Історія розвитку нейронних мереж.....	10
1.2 Поняття «штучна нейронна мережа».....	12
1.2.1 Біологічний нейрон і штучний нейрон.....	13
1.3 Класифікації штучних інтелектуальних систем.....	16
1.4 Основні проблеми, які вирішуються штучними нейронними мережами....	21
1.5 Алгоритм навчання нейронної мережі.....	23
1.6 Однонаправлені багатошарові мережі.....	25
1.7 Рекурентна нейронна мережа.....	27
1.8 Мережі Кохонена.....	29
1.9 Радіальні нейронні мережі.....	31
РОЗДІЛ 2. СУЧАСНІ ПРОГРАМНІ КОМПЛЕКСИ ТА МЕТОДИ ПОШУКУ ОБ'ЄКТІВ НА ЗОБРАЖЕННІ	32
2.1 DIGITS та мережа DetectNet	32
2.2. Мережа YOLO.....	37
2.3 Метод Віоли-Джонса	40
2.4 Алгоритм AdaBoost	43
2.5 Каскад класифікаторів	44
РОЗДІЛ 3. ПОБУДУВАННЯ КАСКАДНОГО КЛАСИФІКАТОРУ ДЛЯ ЛОКАЛІЗАЦІЇ АВТОМОБІЛІВ І ЙОГО ТЕСТУВАННЯ	46
3.1. Підготовка навчальної та тестової вибірки	46
3.2 Структура каскадного класифікатора	47
3.3 Навчання та тестування класифікаторів	49
ВИСНОВКИ	54
Список використаних джерел.....	55
Демонстраційні матеріали.....	58

ВСТУП

У сучасному світі штучні інтелектуальні системи завоювали велику популярність, завдяки багатим можливостям і ефективності використання.

Існує безліч завдань, де необхідно практичне дозвіл з використанням інтелектуальних систем вже зараз: економіка і бізнес, робототехніка, геологорозвідка, математика, біофізика, авіоніка, безпека і охоронні системи, медицина і багато інших. Така велика кількість областей застосування, говорить про те, що інтелектуальні системи - це унікальний набір для вирішення питань аналізу і обробки великого обсягу даних, рішення задач різного рівня складності.

В даний час в системах штучного інтелекту активно використовуються штучні нейронні мережі.

Нейронна мережа - це обчислювальна структура, яка складається з безлічі елементів одного типу. Ці елементи виконують прості функції, а всі процеси, що відбуваються в штучній нейронній мережі можна асоціювати з процесами, що відбуваються в нервовій системі живих організмів.

Нейронні мережі нелінійні за своєю природою, в них відсутня явна залежність, що дозволяє відразу використовувати розроблену технологію (інформаційну модель нейромережі). Протягом багатьох років лінійне моделювання було основним методом моделювання, оскільки для нього добре розроблені процедури оптимізації.

Нейронні мережі нелінійні за своєю природою, в них відсутня явна залежність, що дозволяє відразу використовувати розроблену технологію (інформаційну модель нейромережі). Протягом багатьох років лінійне моделювання було основним методом моделювання, оскільки для нього добре розроблені процедури оптимізації.

Актуальність магістерської роботи полягає в використанні нейронних мереж для вирішення формалізованих задач в інтелектуальному аналізі даних. Зростання обсягів інформації, а також розширення кола технічно складних задач прийняття рішень вимагають систематизації існуючих методів і розробки нових методів і алгоритмів рішення. У магістерській роботі розглядається можливість застосування нейронної мережі при вирішенні задачі розпізнавання об'єктів на фотографіях і відеозаписах.

Штучні нейронні мережі дають багатообіцяючі перспективи в розвитку, а програмне забезпечення має величезну перевагу від їх використання. Крім того, кожна реалізована задача має необмежений і нестандартний набір методів

вирішення. У магістерській роботі розглядається можливість застосування нейронної мережі при вирішенні задачі розпізнавання об'єктів.

Темою магістерської роботи є реалізація інтелектуальної системи розпізнавання об'єктів із застосуванням нейронних мереж.

Основною метою роботи є вибір інформаційної моделі нейромережі і реалізація алгоритму розпізнавання об'єктів. Нейронна мережа повинна бути оптимальна за внутрішньою структурою, способом управління інформаційними потоками між нейронами.

Обрана інформаційна модель буде використана для вирішення практичного завдання.

Об'єктом даного дослідження є підходи, методи розпізнавання об'єктів.

Предметом дослідження є інформаційні моделі штучних нейронних мереж, а також реалізація і опис алгоритму розпізнавання об'єктів.

Наукова новизна магістерської роботи полягає в використанні технологій нейромереж (інформаційних моделей) для реалізації алгоритму розпізнавання об'єктів.

Для реалізації нейронної мережі, яка буде розпізнавати об'єкти потрібно виділити ключові ознаки об'єкту і ключові його прояви, які відрізняють конкретний об'єкт від багатьох інших.

Складність реалізації полягає в навчанні нейронної мережі. Підбір початкових даних для завдання - це, перш за все математичне і геометричний опис об'єкту.

За рахунок зростання складності розв'язуваних науково-технічних завдань, автоматична обробка і аналіз візуальної інформації стають все більш актуальними питаннями. Вже зараз існують цифрові і інтелектуальні камери, а також програмне забезпечення, обробляє зображення.

Сучасні технології стрімко рухаються до моменту, коли комп'ютери стануть самостійними системами, здатними до самонавчання і зростання «особистості», як це відбувається з людиною з моменту його народження. Вже зараз багато розробників вирішують непрості завдання за допомогою машин комп'ютерного зору, застосовуючи для роботи системи різні методи, в тому числі і нейронні мережі.

Дисертаційна робота складається з вступу, --- розділів, висновків, списку використаної літератури

1. ОСНОВНІ ПОНЯТТЯ НЕЙРОНИХ МЕРЕЖ ТА ШТУЧНИХ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

1.1 Історія розвитку нейронних мереж

Вперше про штучні нейронні мережі у світі почали говорити в 1940-х роках. Саме як наукова дисципліна теорія нейронних мереж була відображена в роботі Мак Каллока і Пітса в 1943 році. У даній роботі стверджувалося, що майже будь-яку логічну або арифметичну функцію можна реалізувати за допомогою найпростішої нейронної мережі.

Однією з основних робіт слід відзначити модель Д. Хеба, який в 1949 році сформулював закон, що став відправною точкою для навчання нейронних мереж. Хебом першим припустив, що навчання, перш за все, полягає в зміні сили синаптичних зв'язків. Теорія Хебба - типовий випадок самонавчання, де випробувана система навчається виконувати необхідну завдання без втручання експериментатора. Свій внесок зробив і М. Мінський, завдяки дослідженню ряду задач, в тому числі відомої задачі «виключає» АБО ». Ф. Розенблат в 1958 році запропонував нейронну мережу, названу перцептроном, який був призначений для класифікації об'єктів. При навчанні перцептрон отримував повідомлення від «вчителя». Завдяки повідомленням, можна визначити до якого класу належить даний об'єкт. Крім того, навчений перцептрон був здатний самостійно класифікувати не використалися раніше об'єкти, роблячи при цьому мале число помилок.

Період затишшя в розвитку нейронних мереж припав на 1968-1985 рр. З появою високопродуктивних персональних комп'ютерів стало доступним моделювати нейронні мережі.

Настільною книгою фахівців, які цікавляться теорією нейронних мереж, стала робота Ф. Уоссермена «нейрокомп'ютерних техніка».

Після появи роботи Д. Хопфилда (1982 рік) інтерес до нейронних мереж різко зріс. Хопфилд, ґрунтуючись на правилах навчання Хебба, показав, що завдання з нейронами можуть бути зведені до узагальнення ряду моделей, розроблених на той час у фізиці неупорядкованих систем.

Потім в 1986 році з'явилася робота Вільямса, Румельхарта і Хінтона, яка відповідала на важливе питання про поведінку навчання шаруватих нейронних мереж. Після цього, запропонований Хінтон алгоритм зазнав безліч змін.

У 80-ті роки поступово сформувався міцний теоретичний фундамент, на основі якого сьогодні створюється більшість мереж. Розроблена теорія широко стала застосовуватися в останні два десятиліття для вирішення прикладних завдань. Стали з'являтися фірми, що займаються розробкою програмного забезпечення для конструювання штучних нейронних мереж. У 90-ті роки нейронні мережі стали використовуватися в бізнесі, де вони показали колосальну ефективність при вирішенні багатьох завдань - від передбачення попиту на продукцію до аналізу платоспроможності клієнтів банку.

У 2007 в університеті Торонто Джеффі Хінтон створив алгоритми глибокого вивчення нейронних мереж. При навчанні нижніх шарів мережі, Хінтон використовував обмежену машину Больцмана, яка представляє стохастическую рекуррентную нейронну мережу. Після навчання мережі отримане додаток могло швидко вирішувати поставлену завдання (наприклад, пошук осіб на фотографії). Ця функція на сьогоднішній день вбудована в усі цифрові фотоапарати. Подібна технологія використовується інтернет - пошуковими системами при класифікації картинок.

За оцінками фахівців, в області проектування нейронних мереж і нейрокомп'ютерів очікується технологічний зростання. Чимало нових можливостей було відкрито за останні роки, а роботи в даній області стають важливим внеском в науку, технології, економіку.

Незважаючи на те, що вивчення нейронного моделювання ведеться вже понад шістдесят років, немає жодної області мозку, де процес обробки інформації був би зрозумілий до кінця. Також немає жодного нейрона, для якого можна було б визначити код передачі інформації у вигляді послідовності імпульсів.

В даний час існує велика кількість конфігурацій нейронних мереж, які відрізняються за принципами функціонування, і, отже, спрямовані на різні завдання.

Майбутнє нейрокомп'ютерних технологій буде пов'язано з новими відкриттями в області нейронного моделювання - як тільки вдасться розгадати таємницю функціонування хоча б однієї області мозку, відразу стане ясно багато про інших його областях.

1.2 Поняття «штучна нейронна мережа»

Під штучної нейронної мережею розуміється математична модель, а також її програмна і апаратна реалізація, побудована за принципом біологічних нейронних мереж - нервових клітин живого організму. Це поняття виникло при спробі змодельовати процеси, протікають в мозку людини.

Прототипом для створення нейронних мереж послужили, як це не дивно, біологічні нейронні мережі. Дві третини всієї сенсорної інформації, яка до нас потрапляє, приходять з зорових органів сприйняття. Більше однієї третини поверхні нашого мозку зайняті двома найголовнішими зоровими зонами - дорсальний зоровий шлях і вентральний зоровий шлях. Знання даної інформації необхідно для розуміння принципів роботи нейронних мереж.

Штучна нейронна мережа являє собою систему простих процесорів (штучних нейронів), з'єднаних і взаємодіючих між собою. Кожен з процесорів мережі має справу з сигналами, які періодично надходять або передаються іншим процесорам. Велика мережа здатна вирішувати найскладніші завдання в найкоротші терміни.

З математичної точки зору нейронні мережі являють собою спосіб вирішення нелінійних задач оптимізації. Кібернетика використовує теорію нейронних мереж в рішенні задач адаптивного управління, побудові алгоритмів для робототехніки.

У програмуванні нейронна мережа один із способів вирішення проблеми ефективного паралелізму.

Програмування нейронних мереж має на увазі саме навчання мережі, а не написання програмного коду. Саме завдяки навчанню мережа здатна виявляти залежності між даними (вхідними і вихідними), узагальнювати, спрощувати результати, використовувати знання для розбиття складних завдань на більш прості.

На даний момент технології нейронних мереж мають великі перспективи, так як основна мета нейронної мережі полягає в тому, щоб максимально близько змодельовати роботу людської нервової системи - а саме, її здатності до навчання і виправлення помилок. У цьому полягає головна особливість будь-нейронної мережі - вона здатна самостійно навчатися і діяти на підставі попереднього досвіду, з кожним разом роблячи все менше помилок, як і в разі навчання і виконання роботи людиною.

1.2.1 Біологічний нейрон і штучний нейрон

Мозок людини і її нервова система складаються з нейронів, з'єднаних нервовими волокнами. Між нейронами передаються електричні імпульси за допомогою нервових волокон. Всі дії, які відбуваються з організмом, все роздратування шкіри, очей, біль, процеси мислення - є взаємодія між нейронами. Будова біологічного нейрона представлено на малюнку 1.

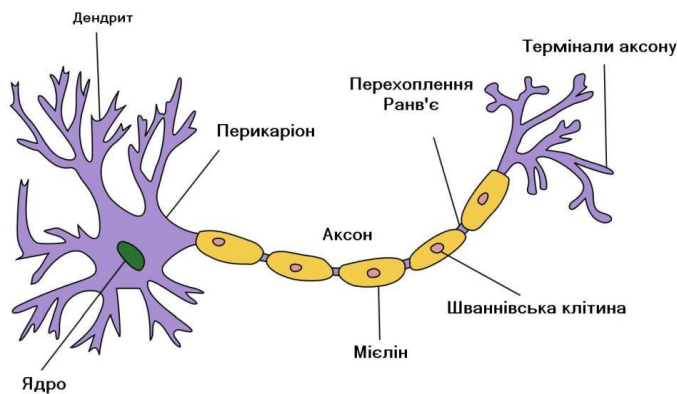


Рисунок 1 - Біологічний нейрон

Біологічний нейрон складається із тіла діаметром від 3 до 130 мкм. Тіло містить ядро (з великою кількістю ядерних пор) та органели, а також з відростків. Виділяють два види відростків: дендрити та аксон. Нейрон має розвинений цитоскелет, який проникає у його відростки. Цитоскелет підтримує форму клітини, його "нитки" служать "рейками" для транспорту органел і упакованих у мембранні бульбашки речовин (наприклад, нейромедіаторів).

Дендрити - короткі і сильно розгалужені в різні відростки нейрона, що є головним місцем для утворення збуджуючих та гальмівних синапсів, що впливають на нейрон (різні нейрони мають різне співвідношення довжини аксона та дендритів). Нейрон може мати кілька дендритів і зазвичай лише один аксон. Нейрони також передають збудження до тіла нейрона.

Аксон - довгий відросток нейрона, який пристосований для передачі збудження та інформації від тіла нейрона (нейросоми) до іншого нейрона або від нейрона до виконавчого органу. Іноді інформація і збудження можуть передаватися до цього самого нейрону.

Синапс — спеціалізоване утворення, яке забезпечує контакт між двома нейронами або між нейроном і клітиною збудливих тканин, що отримує сигнал. Служить передачі нервового імпульсу між двома клітинами, причому під час синоптичної передачі амплітуда і частота сигналу можуть змінюватися. Одні

синапси є збуджуючими, інші гальмівними. Зазвичай для збудження нейрона необхідне роздратування від кількох збудливих синапсів.

Штучний нейрон — вузол штучної нейронної мережі, він представляє собою спрощену модель природнього (біологічного) нейрона. З математичної точки зору, штучний нейрон зазвичай представляють як деяку нелінійну функцію від єдиного аргументу — лінійної комбінації всіх вхідних сигналів. Ця функція має назви "функція активації", "функція спрацьовування" або "передавальна функція". Отриманий результат посилається на єдиний вихід. Такі штучні нейрони об'єднують в мережі — з'єднують виходи одних нейронів з входами інших. Штучні нейрони та мережі є основними елементами ідеального нейрокомп'ютера.

При проходженні синапсу сила імпульсу змінюється в певне число разів (вага синапсу). Коли до нейрона з кількох дендритам надходять імпульси, то вони підсумовуються. Якщо у сумарного імпульсу перевищено поріг, то нейрон переходить в стан збудження, формує власний імпульс і надсилає його далі по аксону. Поведінка відповідного нейрона може змінюватися, так як ваги синапсів мають властивість змінюватися з часом. Математична модель описаного процесу представлена на малюнку 2:

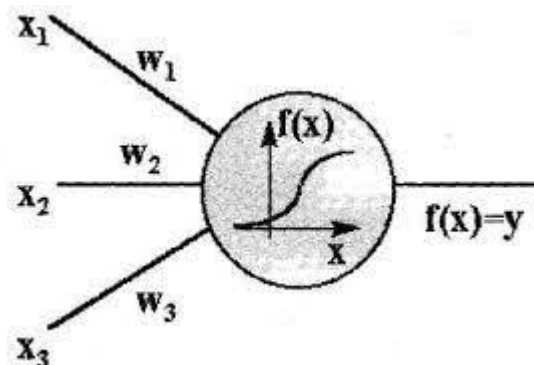


Рисунок 2 - Математична модель нейрона

Дана модель описує нейрон з трьома входами (дендритами), де синапси мають ваги w_1 , w_2 , w_3 , до яких надходять сили x_1 , x_2 , x_3 відповідно. До нейрона надходять імпульси x_1w_1 , x_2w_2 , x_3w_3 після проходження синапсів і дендритів.

Отриманий сумарний імпульс $x = x_1w_1 + x_2w_2 + x_3w_3$ нейрон перетворює відповідно до передавальної функцією $f(x)$.

$y = f(x) = f(x_1w_1 + x_2w_2 + x_3w_3)$ - сила вихідного імпульсу. Підводячи підсумок, отримуємо набір чисел x_k (вектор) у вигляді входів. Далі нейрон видає деяке число y на виході.

Штучний нейрон виглядає наступним чином: на його вхід надходить безліч сигналів, кожен з яких одночасно є виходом іншого нейрона. Такий вхід множиться на відповідну вагу, потім твори підсумовуються, визначаючи рівень активації нейрона. Модель, яка реалізує цю ідею, представлена на малюнку 3.

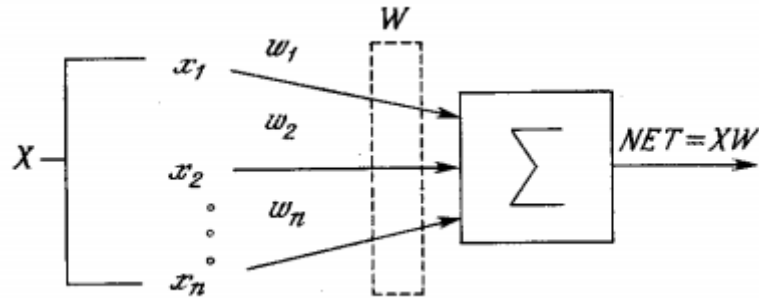


Рисунок 3 - Модель активації нейрона

Тут множина сигналів x_1, x_2, \dots, x_n , надходить на вхід штучного нейрона, в сукупності вони позначаються вектором X . Ці сигнали, аналогічні тим, що отримує на входи біологічний нейрон. Далі сигнал також множиться на відповідну вагу w_1, w_2, \dots, w_n , потім підсумовується в підсумовуючому блоці Σ . Кожен вага дорівнює силі однієї синаптичного зв'язку в біологічному нейроні. Вихід, який зазвичай називається NET, створюється з підсумовує блоку, де зважені елементи складаються алгебраїчно.

Сигнал NET перетвориться звичайної лінійної функцією, званої активаційною. Вона позначається F і дає вихідний сигнал OUT.

$OUT = K(NET)$, де K - постійна, пороговою функції

$OUT = 1$, якщо $NET > T$

$OUT = 0$ в інших випадках,

T - постійна порогова величина, яка більш точно моделює нейронну мережу.

Штучний нейрон з активаційний функцією представлений на рисунку 4:

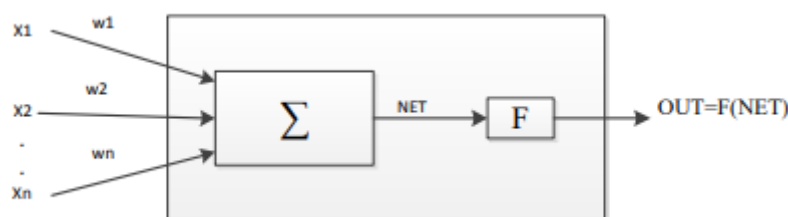


Рисунок 4 – Активаційна функція нейрона

Блок, позначений F, приймає сигнал NET і видає сигнал OUT. F-називається стискає функцією, якщо при будь-яких значеннях NET значення OUT належать деякому кінцевому інтервалу.

Модель штучного нейрона ігнорує більшість властивостей біологічного нейрона. Наприклад, затримки в часі, які впливають на динаміку системи. Вхідні сигнали відразу ж породжують вихідний сигнал. Крім того, штучний нейрон не враховує вплив синхронізуючої функції біологічного нейрона.

Однак слід зазначити виняткове подібність живого нейрона і штучного. Для визначення місця нейронних мереж в області інформаційних технологій необхідно звернутися до класифікації штучних інтелектуальних систем.

1.3 Класифікації штучних інтелектуальних систем

Штучні інтелектуальні системи мають характерні особливості:

По-перше, розвинені комунікативні здібності, які характеризують спосіб взаємодії між комп'ютером користувача з системою. Не виключена можливість звернення до системи з довільним запитом в діалозі з інтелектуальною системою. При цьому мова інтелектуальній системі повинен бути максимально наближений до природної мови.

По-друге, рішення погано формалізованих завдань, тобто таких завдань, які не мають конкретного вирішення, а вимагають нестандартного підходу, в залежності від ситуації, існуючих даних і кінцевого результату. Завдання, що погано формалізується, ефективно вирішуються за допомогою штучних нейронних мереж.

По - третє, здатність до самонавчання - тобто можливості вилучення знань інтелектуальною системою з накопиченого досвіду конкретних ситуацій. Для попереднього навчання системи необхідні оброблені початкові дані.

По-четверте, підтримка прийняття рішень, що означає наявність сукупності процедур, що забезпечує особу, що приймає рішення, необхідною інформацією і рекомендаціями, що полегшують процес ухвалення рішення. Ці експертні системи допомагають фахівцям вибрати і сформулювати потрібну альтернативу серед безлічі виборів при ухваленні відповідальних рішень.

По-п'яте, наявність функції планування. Під плануванням розуміється знаходження планів дій, що відносяться до об'єктів, здатних виконувати деякі функції. У таких системах використовуються моделі поведінки реальних об'єктів з тим, аби логічно вивести наслідки планованої діяльності.

Згідно з поданими ознаками, інтелектуальні системи можна розділити на наступні види та типи (таблиця 1):

Таблиця 1 - Види штучних інтелектуальних систем.

Вид штучної інтелектуальної системи	Тип штучної інтелектуальної системи
Системи з комутативними здібностями	- Інтелектуальна бази даних; - Природно-мовні інтерфейси; - Гіпертекстові системи; - Контекстні довідкові системи; - Когнітивна графіка.
Експертні системи	- Класифікуючі системи; - Довизначаючі системи; - Трансформуючі системи; - Багатоантенні системи.
Самонавчальні системи	- Індуктивні системи; - Нейронні мережі; - Системи на прецедентах; - Інформаційні сховища
Адаптивні системи	- CASE - технології; - компонентна технологія.

Інтелектуальні бази даних відрізняються від звичайних баз даних можливістю вибірки за запитом необхідної інформації, яка може явно не зберігатися, а виводитися з наявною в базі даних.

Природно, мовний інтерфейс перетворює конструкції природної мови у внутрішній рівень представлення знань. Використовується для доступу до інтелектуальних баз даних, контекстного пошуку документальної текстової інформації, команд голосового введення в системах управління, машинного перекладу з іноземних мов.

Гіпертекстові системи застосовуються в базах текстової інформації, де необхідний пошук за ключовими словами, мають більш складну семантичну організацію ключових слів.

У контекстних довідкових системах користувач описує проблему (ситуацію), а система за допомогою додаткового діалогу її конкретизує і виконує пошук рекомендацій, придатних для даної ситуації. Такі системи створюються як додаток до систем документації і належать до класу систем поширення знань.

Системи когнітивної графіки використовуються в моніторингу і управлінні оперативними процесами. Графічні образи в наочному і інтегрованому вигляді описують безліч параметрів досліджуваної ситуації.

Експертні системи призначені для вирішення завдань на основі накопичуваної бази знань, що відбиває досвід роботи експертів в даній проблемній області. Багатоагентні системи - це динамічні системи, для яких

характерна інтеграція в базі знань декількох різних джерел знань, які обмінюються між собою одержуваними результатами на динамічній основі.

Самонавчальні системи засновані на методах автоматичної класифікації прикладів ситуацій реальної практики. Характерними ознаками самонавчальних систем є:

- Самонавчальні системи «з учителем», коли для кожного прикладу задається в явному вигляді значення ознаки його приналежності певного класу ситуацій (класообразуючі ознаки);

- самонавчальні системи «без вчителя», коли за ступенем близькості значень ознак класифікації система сама виділяє класи ситуацій.

Індуктивні системи за принципом від часткового до загального узагальнюють приклади, а процес узагальнення здійснюється наступним чином:

- 1) Вибирається ознака класифікації з безлічі заданих (Послідовно або за правилом).

- 2) За значенням вибраної ознаки безліч прикладів розбивається на підмножини.

- 3) Виконується перевірка щодо належності прикладу до одного класу.

- 4) Якщо якийсь підмножина прикладів належить одному підкласу, тобто у всіх прикладів підмножини збігається значення класообразуючої ознаки, то процес класифікації закінчується (інші ознаки класифікації не розглядаються).

- 5) Для підмножин прикладів з незбіжним значенням класообразуючої ознаки процес класифікації триває, починаючи з пункту 1.

Нейронні мережі - це такі інструменти для паралельних обчислень, які складаються з безлічі простих процесорів, які періодично отримують і відправляють сигнали іншим процесорам. Далі слід звернутися до інтелектуальних інформаційних систем з точки зору вирішуваних завдань. Тут можна виділити наступні системи:

- системи управління і довідкові системи;
- системи комп'ютерної лінгвістики;
- системи розпізнавання;
- ігрові системи;
- системи створення інтелектуальних інформаційних систем.

Системи можуть вирішувати не одну, а кілька завдань або в процесі вирішення одного завдання вирішувати і ряд інших завдань. Інтелектуальні системи також класифікуються за таким критерієм як «використовувані методи». Тут можна виділити м'які, жорсткі і гібридні методи.

М'які обчислення (Soft Computing) - неточні чи наближені методи вирішення тих чи інших завдань, які найчастіше немає рішення за поліноміальний час (коли складність виражена поліномом чи многочленом), об'єднані в один клас[?]. Завдання, які можна вирішити методами такого класу, часто виникають у таких областях як: біологія, медицина, гуманітарні науки, менеджмент та інші.

Жорсткі обчислення (Hard Computing) - це стандартні комп'ютерні обчислення (які не відносяться до м'яких обчислень). Метод жорстких обчислень краще підходить для вирішення математичних завдань, хоча даний метод може використовуватися для вирішення реальних завдань. Основний недолік даного методу полягає в тому, що він споживає велику кількість часу та витрат на обчислення в порівнянні з методом м'яких обчислень. З цієї причини, м'які обчислення вважаються найкращою альтернативою для вирішення реальних проблем.

Гібридні системи - це системи, які використовують дві або більше комп'ютерних технологій (у разі використання інтелектуальних систем – технології штучного інтелекту).

Таблиця 2 - Порівняльна таблиця м'яких та жорстких обчислень

Основа для порівняння	М'які обчислення	Жорсткі обчислення
Основні особливості	Терпимий до неточності, невизначеності, часткової правди та наближення.	Використовується чітко заявлена аналітична модель.
На основі	Нечітка логіка та ймовірні міркування	Бінарна логіка та чітка система
Характеристики	Апроксимація та диспозиційність	Точність та категоричність
Природа	Стохастична	Детерміністична
Працює на	Двозначні та зашумлені дані	Точні вхідні дані
Обчислення	Може виконувати паралельні обчислення	Може виконувати послідовні обчислення
Результат	Здійснює наближений результат	Здійснює точний результат.

Традиційний обчислювальний підхід, заснований на використанні жорстких обчислень, ефективний у разі необхідності вирішення детермінованої проблеми. У міру того, як існуюча проблема збільшується у розмірах та складності її вирішення, простір для пошуку необхідного рішення також збільшується. Це ускладнює вирішення невизначеної та/або неточної проблеми за допомогою складних обчислень. Таким чином, м'які обчислення є відповідним рішенням для жорстких обчислень, яке також має безліч переваг, таких як можливість швидких обчислень, низька вартість використання та інше.



Рисунок 5 - Класифікація інтелектуальних інформаційних систем за методами

Класифікація штучних інтелектуальних систем охоплює велику теоретичну базу знань із різних областей науки. Створення та реалізація штучної інтелектуальної системи є дуже складним і трудомістким процесом від початкового до кінцевого етапів.

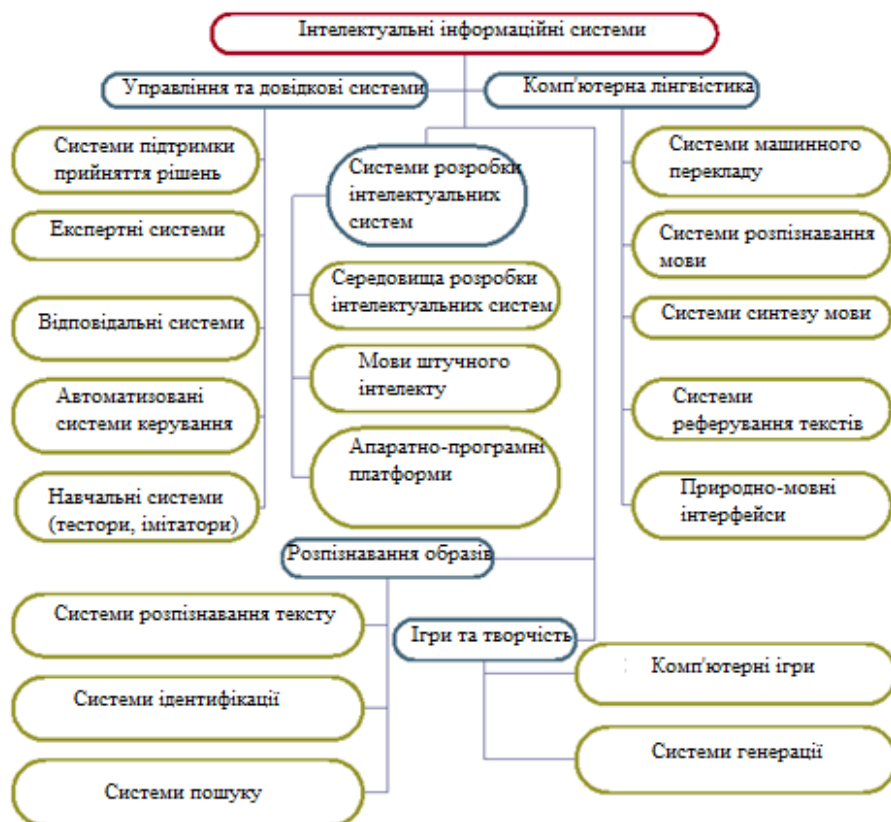


Рисунок 6 - Класифікація інтелектуальних інформаційних систем по вирішуваним завданням

Існує ще один варіант класифікації інтелектуальних систем за методами:

- системи, що використовують методи представлення знань;
- системи, що самоорганізуються;
- системи, створені за допомогою евристичного програмування.

До інтелектуальних систем загального призначення належать ті системи, які не тільки виконують задані процедури, але й основні метапроцедури пошуку рішення генерують і виконують процедури вирішення нових конкретних завдань.

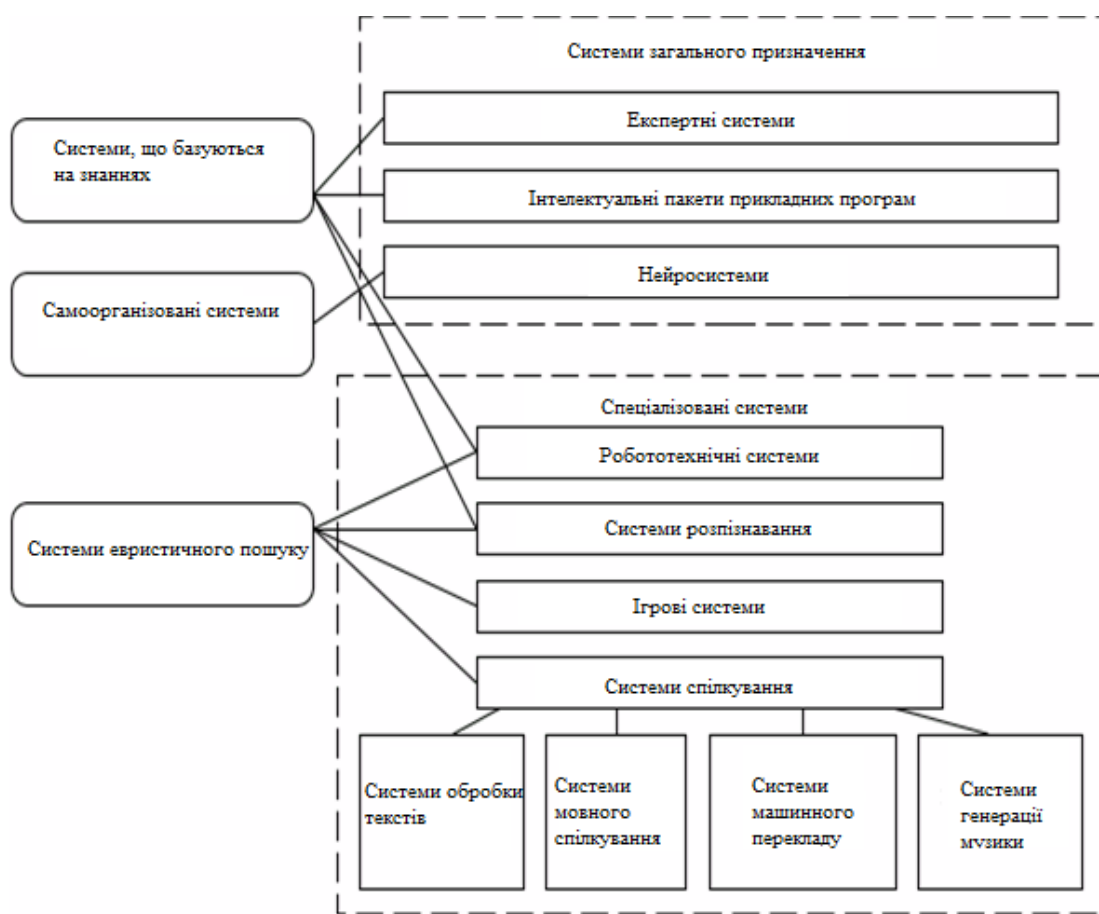


Рисунок 7 - Класифікація інтелектуальних систем за призначенням

1.4 Основні проблеми, які вирішуються штучними нейронними мережами

- Класифікація образів

Завдання для рішення даної проблеми полягає у вказівці приналежності вхідного образу, представленого вектором ознак, одному або декільком попередньо певних класів. До відомих додатків відносяться розпізнавання букв,

розпізнавання мови, класифікація сигналу електрокардіограми, класифікація клітин крові, завдання рейтингування.

- Кластеризація / категоризація

При вирішенні задачі кластеризації, яка відома також як класифікація образів без вчителя, відсутня навчальна вибірка із зразками класів. Алгоритм кластеризації заснований на подоби образів і розміщує близькі образи в один кластер. Відомі випадки застосування кластеризації для вилучення знань, стиснення даних і дослідження властивостей даних.

- Оптимізація

Проблеми в математиці, статистиці, техніці, науці, медицині та економіці, яким немає рахунку, можуть розглядатися як проблеми оптимізації. Завданням оптимізації є знаходження найкращого рішення, яке задовольняє системі обмежень і робить цільову функцію максимізованою або мінімізованою.

- Апроксимація функцій

Припустимо, що існує навчальна вибірка значень $((X_1, Y_2), (X_2, Y_2), \dots, (X_N, Y_N))$, що створюється невідомою функцією, яка спотворена шумами. Завдання апроксимації функцій полягає у знаходженні оцінки цієї функції.

- Пам'ять, яка адресується за її змістом

У моделі обчислень Джона фон Неймана, яка так само відома як "Принстонська архітектура", звернення до пам'яті доступне лише за безпосередньої вказівки адреси необхідної пам'яті, яка не залежить від змісту самої пам'яті. Більше того, якщо була припущена помилка під час обчислення адреси необхідної пам'яті, то може бути знайдено зовсім іншу, не підходящу інформацію. Пам'ять, яка адресується за змістом (так само називається як "асоціативна пам'ять"), доступна за вказівкою вказаного змісту. Вміст необхідної пам'яті може бути викликаний навіть у випадках, коли вміст частково або повністю викривлено. Наявність асоціативної пам'яті є дуже важливим моментом при створенні та розробці перспективних інформаційно-обчислювальних систем.

Яким чином нейронна мережа вирішує всі ці, часто формалізації або важко формалізовані завдання? Як відомо, для вирішення таких завдань традиційно застосовуються два основні підходи. Перший, заснований на правилах (rule-based), характерний для експертних систем. Він базується на описі предметної області у вигляді набору правил (аксіом) «якщо ..., то ...» і правил виведення. Шукане знання представляється в цьому випадку теоремою, істинність якої доводиться за допомогою побудови ланцюжка виведення. При цьому підході, однак, необхідно заздалегідь знати весь набір закономірностей, що описують предметну область. При використанні іншого підходу, заснованого на прикладах (case-based), треба лише мати достатню кількість прикладів для настройки адаптивної системи з заданим ступенем достовірності. Нейронні мережі являють собою класичний приклад такого підходу.

1.5 Алгоритм навчання нейронної мережі

В даний момент найбільш поширеним алгоритмом навчання інтелектуальних систем на основі нейронних мереж є метод градієнтного спуску (або як його ще називають метод зворотного поширення помилки), а так само його модифікації. Цей алгоритм належить до методів навчання з учителем. Це метод знаходження локального мінімуму або максимуму значень функції за допомогою руху вздовж градієнта (вектора, який своїм напрямком вказує напрямок найбільшого зростання деякої скалярної величини, а за величиною дорівнює швидкості зростання цієї величини в цьому напрямку). Для того, щоб мінімізувати функцію в напрямку градієнта, використовують методи одномірної оптимізації (до таких відноситься наприклад метод золотого перерізу). Також за даним методом можна шукати не найкращу точку в напрямку градієнта, а якусь точку, яка краща за поточну точку.

Під час навчання нейронної мережі типу «персептрон» потрібно змінювати вагові коефіцієнти те щоб мінімізувати середню помилку на виході нейронної мережі при подачі на вхід послідовності навчальних вхідних даних. Простіше кажучи, щоб зробити один крок за методом градієнтного спуску (зробити лише одну зміну параметрів мережі), необхідно на вхід мережі ввести абсолютно весь набір навчальних даних послідовно. Після того, як весь набір навчальних даних введено, для кожного об'єкта навчальних даних необхідно визначити помилку та оцінити необхідну корекцію коефіцієнтів мережі (але не робити цю корекцію), і вже після подачі всіх даних визначити суму в коригуванні кожного коефіцієнта мережі, тобто знайти суму градієнтів. Після цього необхідно зробити корекцію коефіцієнтів "на один крок". Зрозуміло, що при великому наборі навчальних даних алгоритм працюватиме дуже повільно, тому при виконанні практичних завдань найчастіше просто коригують коефіцієнти мережі після кожного елемента навчання, де значення градієнта апроксимуються градієнтом функції вартості, обчисленому тільки на одному елементі навчання. Цей метод називають стохастичним градієнтним спуском (або оперативним градієнтним спуском).

Існує три способи реалізації алгоритму зворотного розповсюдження помилки:

- Послідовний режим

Як було зазначено раніше, послідовний режим навчання також іноді називають стохастичним градієнтним спуском. У цьому режимі зміна ваги зв'язків відбувається після подачі кожного прикладу з навчальної вибірки.

- Пакетний режим

У разі застосування пакетного режиму навчання коригування ваги зв'язків відбувається тільки після подачі на вхід мережі всіх навчальних прикладів однієї епохи навчання.

- Mini-batch;

Як компроміс між методами градієнтного спуску і пакетного режиму рекомендують також застосовувати міні-пакети ("mini-batch"). У цьому випадку коригування синоптичних ваги мережі відбувається після невеликої кількості навчальних зразків.

Якщо дивитися на можливі методи точки зору продуктивності, послідовний режим навчання є кращим, ніж пакетний, так як для зберігання кожного синоптичного зв'язку потрібен менший обсяг внутрішньої пам'яті пристрою. Так само, пред'явлення навчальних прикладів у випадковому порядку в процесі навчання для послідовно режиму, робить пошук у просторі терезів стохастичним, тим самим зменшуючи ймовірність зупинки алгоритму в точці якого-небудь локального мінімуму.

Незважаючи на численні успішні застосування методу зворотного розповсюдження, даний метод навчання мережі не є універсальним рішенням. Найбільше неприємностей приносить невизначено тривалий процес навчання. У складних завданнях для навчання мережі можуть знадобитися дні або навіть тижні, вона може взагалі не навчитися. Причиною може бути одна з наведених нижче:

- параліч мережі;
- розмір кроку;
- локальні мінімуми.

На малюнку 8 можна спостерігати послідовні наближення після застосування градієнтного методу до наступної функції:

$$F(x, y) = \sin\left(\frac{1}{2}x^2 - \frac{1}{4}y^2 + 3\right) \cos(2x + 1 - e^y). \quad (1.1)$$

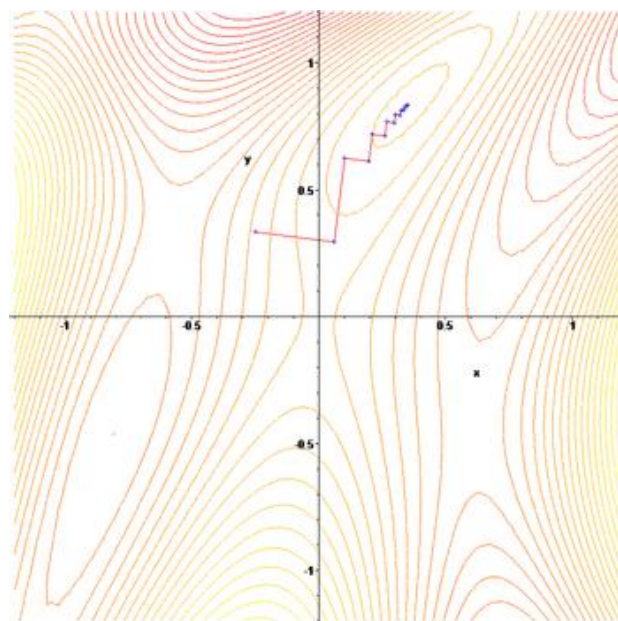


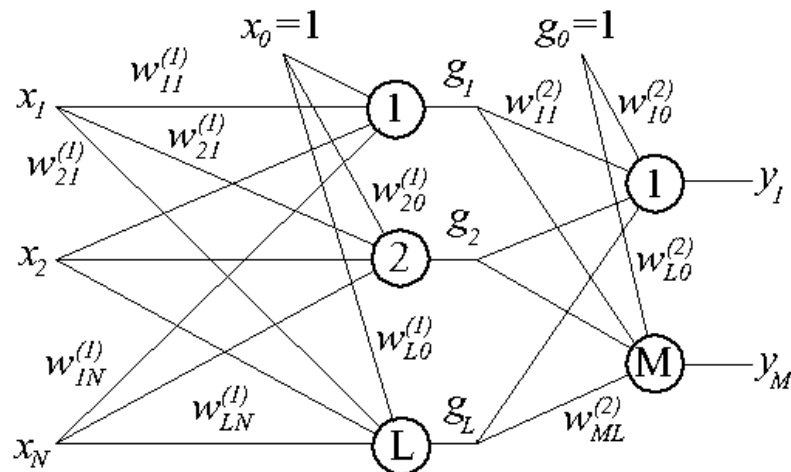
Рисунок 8 - послідовні наближення функції після застосування градієнтного методу

1.6 Однонаправлені багат шарові мережі

Мережі даного типу набули найбільшого поширення серед інших мереж при виконанні практичних завдань через простоту способу їх математичного опису.

Історія інтелектуальних нейронних мереж розпочалася з одношарових мереж. Так як саме вони є першими, можливості одношарових мереж дуже малі, оскільки розташовані на одному шарі нейрони функціонують незалежно один від одного, і властивості всієї нейронної мережі обмежені властивостями окремих нейронів. Багат шарові мережі отримали свій розвиток з кінця 70-х років, коли було запропоновано ефективніші алгоритми їх навчання.

Однонаправлені багат шарові мережі сигмоїдального типу мають усталену назву багат шаровий перцептрон MLP (MultiLayer Perceptron). На малюнку представлена структурна схема двошарового перцептрона.



Позначення:

$g_l, L = 1, 2, \dots, L$ - сигнали, що виходять з першого шару нейронів;

$(M), m = 1, 2, \dots$ - індекси в дужках вгорі означають номер шару нейрона;

x_1, x_2, \dots, x_N - вхідні сигнали, які формують вхідний шар;

g_1, g_2, \dots, g_L - перший прихований шар;

y_1, y_2, \dots, y_M - вихідні сигнали, які формують шар на виході.

Рисунок 9 - структурна схема багат шарового перцептрона та позначення до нього.

Слід зазначити, що найчастіше поняття "шар" застосовують по відношенню до самих сигналів ІНС (а не до нейронів). У цьому сенсі представлена малюнку мережа - тришарова: вхідні сигнали мережі x_1, x_2, \dots, x_N становлять вхідний шар, вихідні сигнали першого шару g_1, g_2, \dots, g_L утворюють

перший прихований шар, а вихідні сигнали y_1, y_2, \dots, y_M – вихідний шар. Для позначення структури мережі використовується кодування як "N-L-M".

Вихідні сигнали нейронних шарів легко розраховуються за такими формулами:

$$\begin{aligned} g_l &= f(\text{sum}[j=0:N](w^{(1)}_{lj} * x_j)), \quad l=1, 2, \dots, L; & (1.2) \\ y_i &= f(\text{sum}[l=0:L](w^{(2)}_{il} * g_l)) = f(\text{sum}[l=0:L](w^{(2)}_{il} * f(\text{sum}[j=0:N](w^{(1)}_{lj} * x_j))), \quad i=1, 2, \dots, M & (1.3) \end{aligned}$$

Необхідно підкреслити, що функції активації всіх нейронів мережі абсолютно ідентичні (зокрема мають однакове значення параметра b).

Мета навчання багат шарового нейрона - підбір таких значень всіх вагових коефіцієнтів мережі $w^{(1)}_{lj}$ та $w^{(2)}_{il}$, які можуть забезпечити максимальний збіг вихідного вектора Y_k та еталонного вектора очікуваних значень D_k при пред'явленні вхідного вектора X_k .

У разі одиничної навчальної вибірки $\langle X, D \rangle$ цільова функція визначається як:

$$E(W) = (1/2) * \text{sum}[i=1:M]((y_i - d_i)^2). \quad (1.4)$$

У разі кількох навчальних пар $\langle X_k, D_k \rangle$, $k=1, 2, \dots, p$, цільова функція перетворюється на суму по всіх парах:

$$E(W) = (1/2) * \text{sum}[k=1:p](\text{sum}[i=1:M]((y_i - d_i)^2)). \quad (1.5)$$

Представлені раніше вирази для цільової функції $E(W)$ характеризуються аналітичним виглядом та двічі диференційованістю. Це дає можливість використовувати для пошуку мінімуму $E(W)$ майже весь арсенал універсальних способів пошукової оптимізації. Крім того, спеціально для цілей навчання ІНС розроблено модифікації "канонічних" методів.

Серед універсальних методів для навчання мереж найбільшого поширення набули такі.

- Метод градієнта. Чергове наближення до оптимального рішення визначається за формулою. Де μ - коефіцієнт навчання, що адаптивно підбирається в ході пошуку так, щоб $E_{k+1} < E_k$;

$$W^{k+1} = W^k - \mu * \text{grad}E(W^k) \quad (1.6)$$

- Метод якнайшвидшого спуску. Коефіцієнт навчання μ кожному кроку пошуку такий, що гарантовано забезпечує досягнення мінімуму E_{k+1} у бік $-\text{grad}E(W_k)$. Розрахунок оптимального значення μ реалізується, як правило, методом поліноміальної апроксимації (одномірною оптимізацією в антиградієнтному напрямку);

$$W^{k+1} = W^k - \mu * \text{grad}E(W^k) \quad (1.7)$$

- Метод сполучених градієнтів. У цьому методі напрямком пошуку на поточному кроці визначається як градієнтом цільової функції у точці W_k , так і напрямком пошуку на попередньому кроці p_{k-1} ,

помноженому на коефіцієнт пари b^{k-1} . Існують різні правила розрахунку b^{k-1} ;

$$W^{k+1} = W^k + nu * p^k, \quad (1.8)$$

$$p^k = -gradE(W^k) + b^{k-1} * p^{k-1}. \quad (1.9)$$

$$b^{k-1} = (gradE(W^k)^T * (gradE(W^k) - gradE(W^{k-1}))) / (gradE(W^{k-1})^T * gradE(W^{k-1})). \quad (1.10)$$

- Ньютоноподібні методи. де $H^{-1}(W^k)$ - матриця Гессе (матриця других похідних цільової функції). У силу трудомісткості розрахунку матриці Гессе метод Ньютона як такої практично не використовується, але служить прототипом низки ефективних методів, пропонують економічні методи розрахунку наближених еквівалентів матриці Гессе і вектора градієнта кожному кроці пошуку.

$$W^{k+1} = W^k - H^{-1}(W^k) * gradE(W^k), \quad (1.11)$$

1.7 Рекурентна нейронна мережа

Рекурентна нейронна мережа є найскладнішим видом нейронних мереж та має зворотний зв'язок між своїми елементами: від протилежного до меншвіддаленим комп'ютером. Ідея RNN полягає у послідовному використанні інформації. У традиційних нейронних мережах мається на увазі, що це входи і виходи незалежні. Але для багатьох завдань це не підходить.

RNN називаються рекурентними, тому що вони виконують ту саму задачу для кожного елемента послідовності, причому вихід залежить від попередніх обчислень. Ще одна інтерпретація RNN: це мережі, які мають «пам'ять», яка враховує попередню інформацію. Теоретично RNN можуть використовувати інформацію у довільно довгих послідовностях, але практично вони обмежені лише кількома кроками.

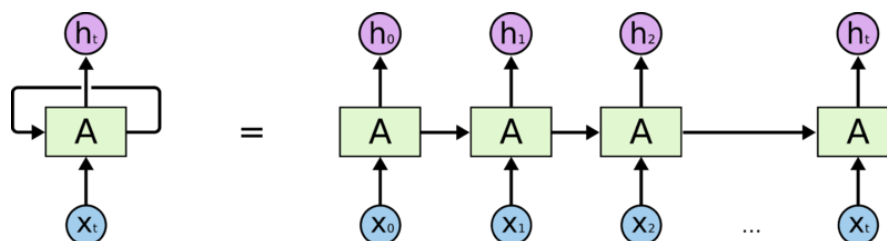


Рисунок 10 - RNN та її розгорнуте уявлення

Один з різновидів таких мереж - це рекуррентная мережа Елмана. Добре видно, що зворотні зв'язки йдуть від внутрішніх шарів, а не від виходів нейронної мережі. Саме така структура дозволяє накопичувати і запам'ятовувати інформацію.

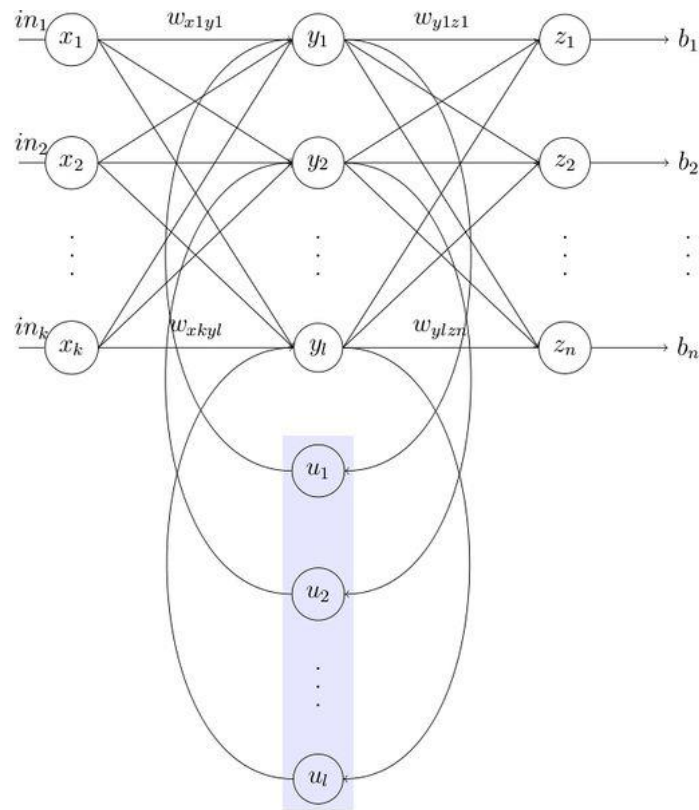


Рисунок 11 - Мережа Елмана

Нейронна мережа Елмана складається із трьох шарів: x , y , z . Додатково до мережі додано набір "контекстних блоків": u . Середній (прихований) шар з'єднаний з контекстними блоками з фіксованою вагою, що дорівнює одиниці. З кожним кроком часу на вхід надходить інформація, яка проходить прямий хід до вихідного шару відповідно до правил навчання. Фіксовані зворотні зв'язки зберігають попередні значення прихованого шару в контекстних блоках (до того, як прихований шар змінить значення в процесі навчання). Таким способом мережа зберігає свій стан, що може використовуватися у передбачанні послідовностей, виходячи за межі потужності багат шарового перцептронну.

Формула Елмана виглядає наступним чином:

$$\frac{\partial E^k(W)}{\partial w_{nm}^{(2)}} = e_n^k \cdot \frac{\partial f_2(g_n^k)}{\partial g_n^k} \cdot v_m^k \quad (1.12)$$

Нейронна мережа Джордана подібна до мережі Елмана, але в мережі Джордана контекстні блоки пов'язані не з прихованим шаром, а з вихідним шаром. Контекстні блоки таким чином зберігають свій стан, тим самим вони мають рекурентний зв'язок з собою.

$$\begin{aligned} h_t &= \sigma_h(W_h x_t + U_h y_{t-1} + b_h), \\ y_t &= \sigma_y(W_y h_t + b_y), \end{aligned} \quad (1.13)$$

1.8 Мережі Кохонена

Для вирішення різних завдань класифікації використовується окремий клас нейронних мереж, названий мережами Кохонена. Це клас нейронних мереж, основним елементом яких є шар Кохонена. Шар Кохонена складається з адаптивних лінійних суматорів (лінійних формальних нейронів). Як правило, вихідні сигнали шару Кохонена обробляються за правилом "Переможець отримує все": найбільший сигнал перетворюється на одиничний, інші звертаються в нуль.

За способами налаштування вхідних ваг суматорів та за розв'язуваними завданнями розрізняють багато різновидів мереж Кохонена. Найбільш відомі з них:

- мережі векторного квантування сигналів, що тісно пов'язані з найпростішим базовим алгоритмом кластерного аналізу (метод динамічних ядер або К-середніх);
- самоорганізовані карти Кохонена;
- мережі векторного квантування, які навчаються з учителем.

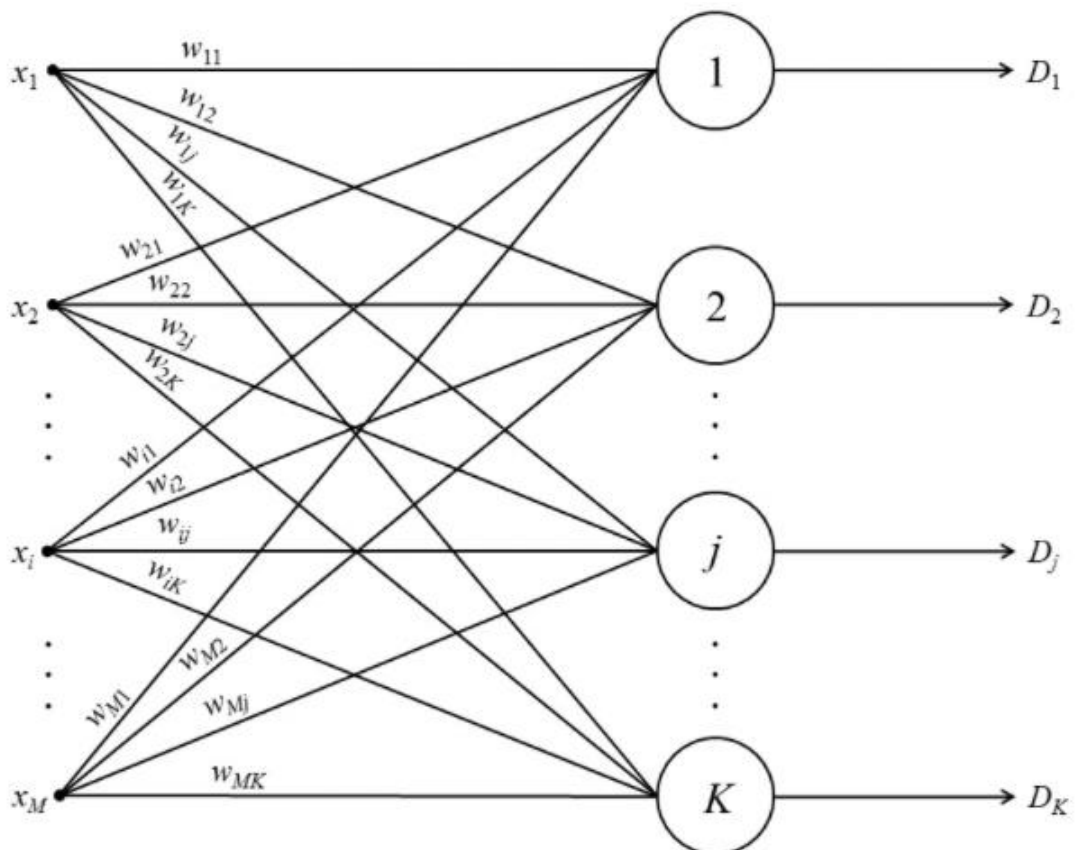


Рисунок 12 - Загальна структура нейронної мережі Кохонена

Нормалізація вхідних змінних виконується в межах $[-1, 1]$ або $[0, 1]$. Для життєвого циклу нейронних мереж даної архітектури характерні три основні стадії життєвого циклу: навчання, кластерний аналіз та практичне використання.

Мережі, які побудовані на основі нейронів типу WTA (Winner Takes All), які виглядають наступним чином:

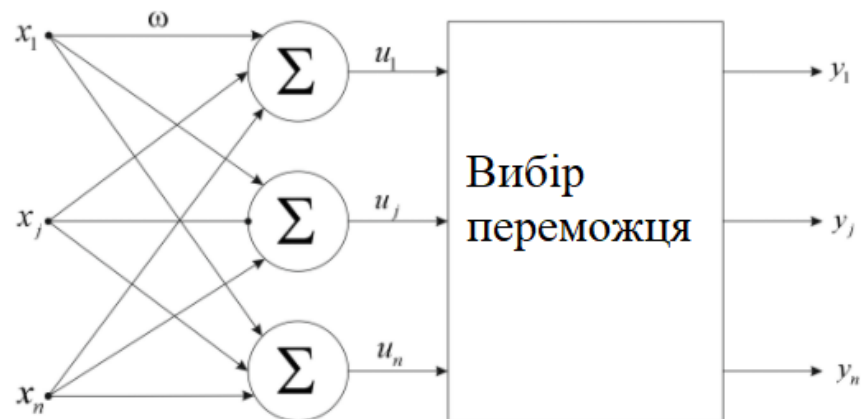


Рисунок 13 - Нейронна мережа типу WTA (Winner Takes All)

Вхідні сигнали для всіх конкуруючих нейронів однакові, вихідні сигнали вважаються за формулою:

$$u_i = \sum_{j=0}^N w_{ij} \cdot x_j \quad (1.14)$$

Проблеми, що виникають під час навчання:

- Переможцем етапу виявляється нейрон, вектор ваг якого виявляється найближчим до поточного навчального вектора x . Внаслідок перемоги нейрона уточнюються його вагові коефіцієнти, значення яких наближаються до значень поточного навчального вектора x . Якщо на вхід буде подаватися безліч близьких за значенням векторів, перемагатиме той самий нейрон, тому його ваги стануть рівними усередненому значенням тих векторів, завдяки яким нейрон став переможцем.
- Проблемою при навчанні нейронів типу WTA є проблема «мертвих» нейронів, які після ініціалізації жодного разу не перемогли у конкурентній боротьбі та залишилися у стані сформованому у початковий момент часу. Кожен мертвий нейрон зменшує ефективну кількість елементів, що проходять навчання, та збільшує загальну похибку розпізнавання даних.
- Для вирішення цієї проблеми пропонується модифіковане навчання, засноване на обліку перемог кожного нейрона та тимчасової дискваліфікації тих, що перемагали найчастіше. Вона може здійснюватися або після досягнення граничного числа перемог, або зменшенням фактичного значення числа при наростанні числа перемог нейрона.

1.9 Радіальні нейронні мережі

Мережа радіальних базисних функцій - нейронна мережа прямого поширення сигналу, яка містить проміжний (прихований) шар симетричних нейронів. Такий нейрон перетворює відстань відданого вхідного вектора до відповідного йому "центру" по деякому нелінійному закону. Вперше сформульовані в 1988 році Брумхедом і Лоу.

Радіальні мережі будуються з використанням радіальних нейронів, функція активації яких має ненульові значення лише на околицях свого центру. Тому апроксимація за допомогою таких мереж називається локальною апроксимацією.

Радіальна мережа має двошарову структуру, перший шар складають радіальні нейрони, вихідний – один або кілька лінійних. На малюнку представлена структурна схема радіальної мережі з одним вихідним нейроном.

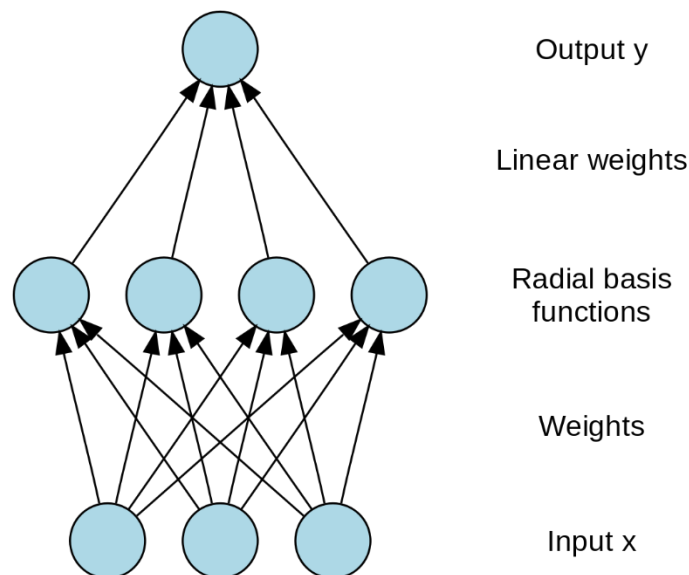


Рисунок 14 - Структурна схема радіальної мережі з одним вихідним нейроном

$f_i(|X-C_i|)$ - функція активації i -ого радіального нейрона, звана також базисною функцією.

Мережа виконує апроксимацію вхідних даних згідно з виразом (для спрощення викладок покладемо $w_0=0$):

$$y = \sum_{i=1:L} (w_i * f_i(|X-C_i|)). \quad (1.15)$$

Завдання навчання мережі полягає у підборі таких значень L , C_i та w_i , які доставляють мінімум цільової функції. p – кількість навчальних вибірок.

$$E = (1/2) * \sum_{k=1:p} ((\sum_{i=1:L} (w_i * f_i(|X-C_i|)) - d^k)^2). \quad (1.16)$$

2. СУЧАСНІ ПРОГРАМНІ КОМПЛЕКСИ ТА МЕТОДИ ПОШУКУ ОБ'ЄКТІВ НА ЗОБРАЖЕННІ

2.1 DIGITS та мережа DetectNet

DIGITS – це повністю інтерактивний інструмент, розроблений компанією NVidia для вирішення різноманітних загальних завдань у галузі глибокого навчання. З його допомогою дослідники-аналітики можуть:

- готувати дані,
- визначати згорткові мережі,
- паралельно навчати кілька моделей,
- спостерігати за процесом навчання в режимі реального часу,
- вибирати найкращу модель.

При цьому ця веб-програма повністю позбавляє користувача від програмування та налагодження, і можна зробити основний упор саме на навчання різних моделей мережі та пошук кращої для конкретної задачі.

У новій версії DIGITS 4 реалізовано новий підхід до завдання виявлення об'єктів на зображенні. Він дозволяє навчати нейронні мережі для пошуку об'єктів і визначати прямокутники навколо них. Одним із них є нова мережева архітектура DetectNet, розроблена інженерами Nvidia. На малюнку продемонстровано результат роботи цієї мережі для пошуку транспортних засобів на аерофотознімках.



Рисунок 15 - Застосування мережі DetectNet для пошуку транспортних засобів

З одного боку, для тренування навчальної вибірки можна використовувати зображення невеликого розміру, які містять один об'єкт, та класові мітки (цілочисленний ідентифікатор класу). З іншого боку, для пошуку об'єктів на зображенні необхідно більше інформації для навчання. Тому для мережі DetectNet вже використовуються зображення більшого розміру, що містять не один, а кілька об'єктів, для кожного з яких класова мітка доповнюється інформацією про розташування кутів прямокутника, що обмежує їх. Так як у цій ситуації кількість об'єктів на зображеннях навчальної вибірки можуть змінюватись, то може бути утруднено визначення функції втрат.

Для вирішення цієї проблеми мережа DetectNet використовує фіксований тривимірний формат мітки. Завдяки цьому можна використовувати зображення з різним розміром та будь-яким числом присутніх на них об'єктів.

Схема обробки для навчання мережі DetectNet зображень з розміткою з навчальної вибірки зображена на малюнку:

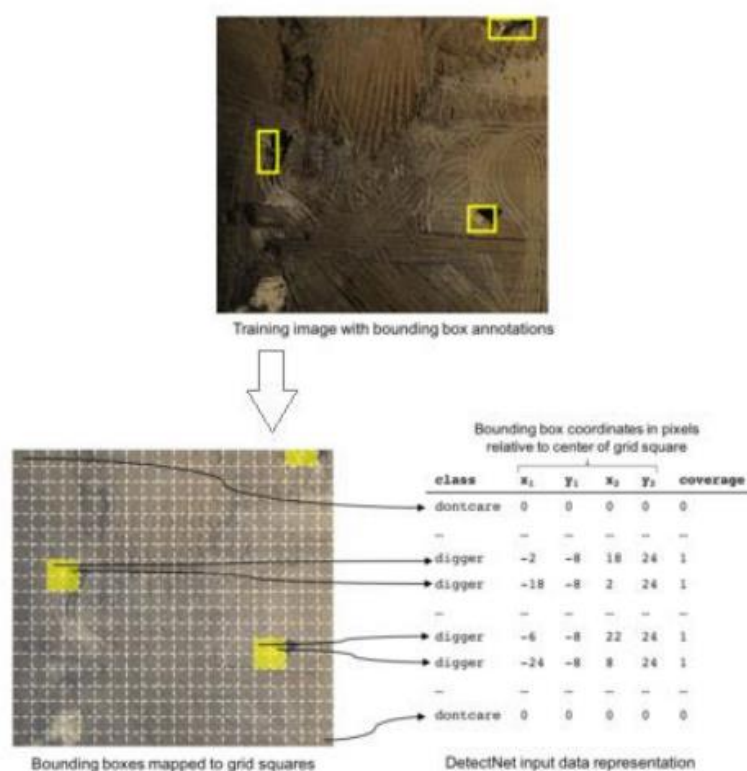


Рисунок 16 - Схема представлення вхідних даних DetectNet

На першому етапі на вихідне зображення накладається сітка, яка має фіксований розмір клітин. Цей розмір вибирається так, щоб він був трохи меншим за найменший об'єкт, який потрібно розпізнати.

На другому етапі, кожен осередок сітки доповнюється такою інформацією:

- клас об'єкта, що знаходиться в осередках сітки,
- координати пікселя кутів, що обмежує прямокутника.

Якщо жоден розпізнаваний об'єкт не потрапив у комірку сітки, то використовується особливий клас "dontcare", щоб дотримувався фіксований формат даних. Також у цей формат даних додається ще одне значення, яке називається "coverage". Воно може приймати два значення: "0" або "1". "Coverage" потрібно для того, щоб вказати, чи є в комірці сітки об'єкт. Для випадку, коли в один осередок сітки потрапляють кілька об'єктів, мережа DetectNet вибирає об'єкт, що займає найбільшу кількість пікселів.

Але бувають і випадки, коли два об'єкти мають однакову кількість пікселів. Тоді вибирається той, який має найменшу ординату (OY) прямокутника, що обмежує. Для завдання пошуку транспортних засобів на супутникових зображеннях та аерофотознімку такий вибір об'єктів зазвичай не є важливим.

Основною метою навчання мережі DetectNet є передбачення для кожної осередки сітки про наявність об'єкта в ній та обчислення відносних координат кутів прямокутника, що обмежує цей об'єкт.

Усього нейронна мережа DetectNet має 5 частин, визначених у файлі моделі мережі фреймворку Caffe. Існують два етапи: навчання та перевірка.

На малюнку показано схему архітектури мережі першому етапі:

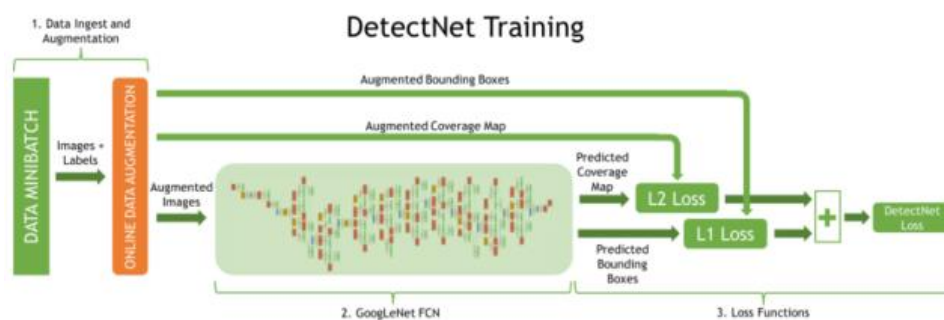


Рисунок 17 - Архітектура DetectNet для навчання

На даній схемі можна виділити три основні процеси:

- Зображення та мітки об'єктів з навчальної вибірки надходять на вхід шару даних. Потім, перетворюючий шар "на лету" доповнює дані.
- Повністю згорткова нейронна мережа (FCN) отримує ознаки, а також передбачає класи об'єктів і прямокутники, що обмежують їх, по осередках сітки.

- Функції втрат, паралельно, вважають помилку щодо завдань передбачення покриття об'єкта і кутів прямокутника, що обмежує його, по осередках сітки.

На наступному малюнку показано схему архітектури мережі на другому етапі:

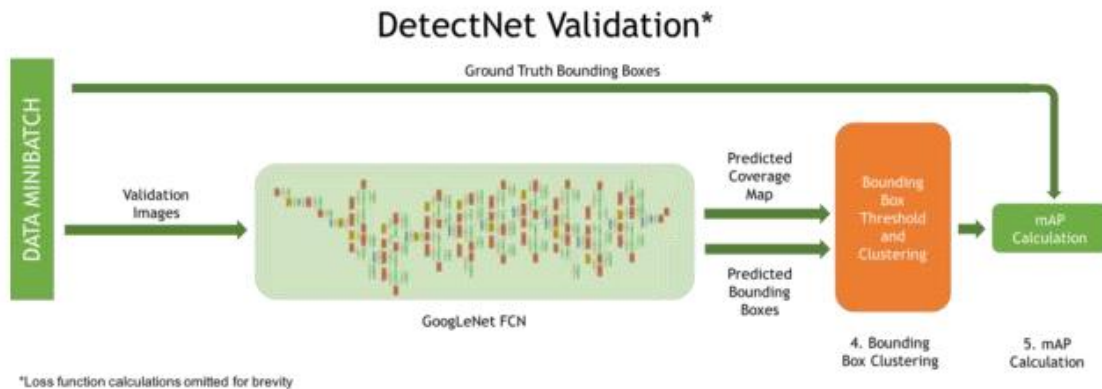


Рисунок 18 - Архітектура DetectNet для перевірки

На цьому етапі з'являються два додаткові процеси:

- Кластеризація прямокутників, які обмежують об'єкти, що шукаються, для отримання остаточного набору.
- 2. Підрахунок метрики mAP (mean Average Precision). Вона потрібна для вимірювання ефективності збудованої моделі по всій тестовій вибірці.

Розмір осередку сітки для навчальних міток можна змінити, задаючи крок для шару. У параметрах даного шару також можна вказати розмір зображень, що використовуються для навчання (`image_size_x`, `image_size_y`). Якщо вказати дані параметри, всі зображення, які потраплятимуть на вхід мережі DetectNet, будуть випадковим чином обрізатися за заданими розмірами. Це корисно в тому випадку, якщо навчальна вибірка складається з дуже великих за розміром зображень, а об'єкти на них, які потрібно розпізнати, дуже маленькі.

Процедура доповнення вхідних даних на льоту відіграє дуже важливу роль під час навчання мережі DetectNet. Завдяки їй отриманий детектор об'єктів стає високочутливим та точним. Параметри шару, приклад якого показаний вище, визначають різні випадкові перетворення над навчальною вибіркою, такі як зсув, відображення тощо. Дані перетворення даних призводять до того, що DetectNet не обробляє двічі одне й теж зображення.

Структура підмережі FCN, яка використовується в DetectNet, заснована на мережі GoogLeNet без шару вхідних даних, заключного шару та вихідних шарів. Завдяки цьому підходу знижується час навчання мережі DetectNet та покращується її точність. FCN – це повна згорткова нейронна мережа, шари якої пов'язані в повному обсязі. Отже, можна приймати на вхід зображення з будь-яким розміром і рахувати відгук, використовуючи техніку «ковзного» вікна з кроком. В результаті, GoogLeNet без заключного шару - нейронна мережа з рухомим вікном розміру 555x555 px і кроком 16 px.

Для створення остаточної функції втрат та оптимізації мережа DetectNet використовує лінійну комбінацію двох незалежних один від одного функцій втрат:

$$\frac{1}{2N} \sum_{i=1}^N |coverage_i^t - coverage_i^p|^2 \quad (2.1)$$

$$\frac{1}{2N} \sum_{i=1}^N [|x_1^t - x_1^p| + |y_1^t - y_1^p| + |x_2^t - x_2^p| + |y_2^t - y_2^p|] \quad (2.2)$$

У першій функції обчислюється квадратична помилка по всіх осередках сітки вихідних даних між справжнім та передбаченим покриттям об'єкта. У другій знаходиться середня помилка між справжніми і передбаченими кутами прямокутника, що обмежує об'єкт, що шукається, по всіх осередках сітки.

В останніх шарах мережі DetectNet відбуваються наступні два процеси: фільтрація та кластеризація набору згенерованих прямокутників, що обмежують об'єкт для комірки сітки. Фільтрування здійснюється за допомогою порогового методу за значенням передбаченого покриття об'єкта. Кластеризація проводиться з використанням критерію еквівалентності прямокутників. За допомогою даного критерію відбувається об'єднання фігур подібного розташування та розміру. За схожість прямокутників відповідає мінлива ерс. Якщо її значення дорівнює нулю, то прямокутники об'єднуються, а якщо прагне нескінченності, всі прямокутники потрапляють в один кластер. Після об'єднання відбувається порогова фільтрація за параметром `gridbox_rect_thresh`, яка відокремлює малі кластери, а за іншими розраховуються середні прямокутники, які потрапляють у вихідні дані. Цей метод кластеризації реалізований за допомогою Python у фреймворку Caffe через інтерфейс «Python Layers».

У мережі DetectNet для обчислення та виведення метрики mAP також використовується інтерфейс «Python Layers». Для передбаченого прямокутника, який обмежує об'єкт, що шукається, і справжнього прямокутника вважається параметр IoU (відношення площі перетину цих прямокутників до їх суми). Цей параметр використовується для розділення прямокутників на TP (True Positive)

або FP (False Positive) та FN (False Negative). У результаті, метрика mAP у цій мережі розраховується як добуток точності (precision) та міри повноти (recall).

Дана метрика є зручною характеристикою чутливості до відкидання помилкових результатів, виявлення об'єктів навчальної вибірки, так і точності знайдених прямокутників, які обмежують об'єкт, що шукається.

$$precision = \frac{TP}{TP + FP} \quad (2.3)$$

$$recall = \frac{TP}{TP + FN} \quad (2.4)$$

Мережа DetectNet є достатньо ефективною та точною завдяки тому, що використовує FCN. Навчання мережі DetectNet в DIGITS 4 на вибірці з 307 навчальних і 24 тестових зображень (1536x1024px) з Nvidia Caffe 0.15.7 і cuDNN RC 5.1, займає близько 63 хвилини, використовуючи одну відеокарту Titan X. 24 fps) на зображеннях розміром 1536x1024px (розмір сітки 16 px, одна відеокарта TitanX, Nvidia Caffe 0.15.7, cuDNN RC 5.1).

2.2 Мережа YOLO

Ще однією ефективною та достатньою швидкою нейронною мережею є мережа YOLO (You Only Look Once), розроблена Джозефом Редмоном та його командою. Ця мережа найпростіша, на відміну від Faster R-CNN. Оскільки мережа DetectNet базується на мережі YOLO, то й архітектури у них схожі.

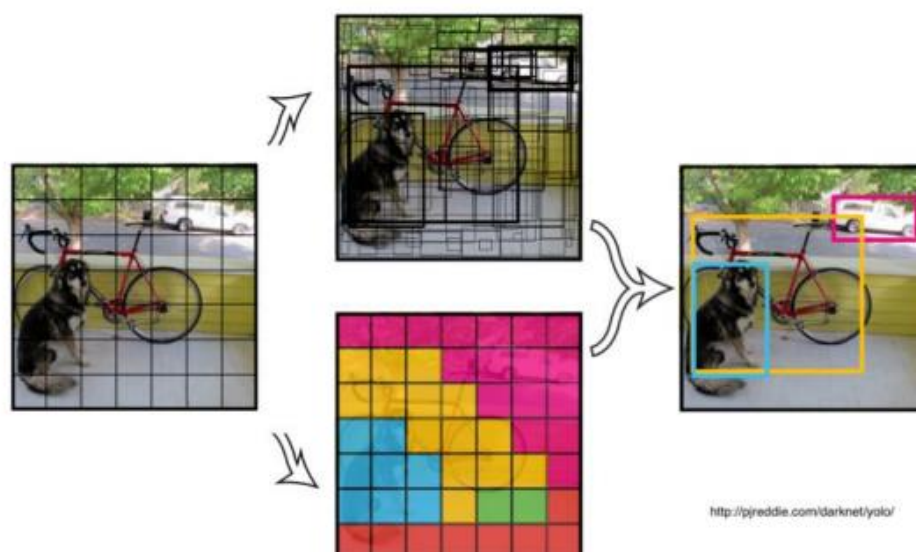


Рисунок 19 - Застосування мережі YOLO для розпізнавання об'єктів на зображенні

На зображення накладається сітка (в даному випадку 7x7), яка ділить його на кілька осередків, у кожному осередку сітки окремо застосовується класифікатор, і після цього робиться пророцтво про те, де знаходиться центр шуканого об'єкта і де знаходиться прямокутник, що обмежує його. Працює мережа YOLO набагато швидше за R-CNN, Fast R-CNN і Faster R-CNN, при цьому точність майже не падає.

Для навчання та тестування нейронної мережі YOLO в рамках поставленого завдання використовуватиметься відкритий набір даних COWC (Cars Overhead With Context). Навчальна вибірка складалася з 2418 зображень розміром 416x416 пікселів. Всього на цих зображеннях присутні 13303 унікальні автомобілі.



Рисунок 20 - Приклади зображень із навчальної вибірки

Навчання було зроблено для 1600 епох (повний прохід через усі навчальні дані), що зайняло 4 дні на одній відеокарті NVIDIA Titan X GPU.

Для тестування використовувалися 23 зображення розміром 4000x4000 пікселів із набору даних штату Юта. Загалом дані зображення містили 25980 автомобілів. В результаті обробки вийшли такі підсумки, подані далі на малюнках. Як міра оцінки якості розпізнавання була обрана міра F1, яка обчислюється наступним способом:

$$F_1 = 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}} \quad (2.5)$$

За цими двома прикладами видно, що мережа YOLO досить добре F1=0,95 впоралася для зображення з міською місцевістю і де автомобілі розташовувалися не дуже щільно один одному. Але для зображення з замиською місцевістю, на якому автомобілі розташовані дуже щільно один до одного, результати розпізнавання були набагато гіршими за F1=0,67.



Рисунок 21 - Приклад 1 із тестової вибірки



Рисунок 22 - Приклад 2 із тестової вибірки

2.3 Метод Віоли-Джонса

У минулому розділі були докладно розглянуті дві нейронні мережі для пошуку об'єктів на зображенні, остання з яких показала досить добрий результат у рамках пошуку автотранспортних засобів на супутникових зображеннях. У цьому розділі буде детально розглянутий метод розроблений Віолою і Джонсом в 2001 році, який дуже добре показав себе в задачі пошуку обличчя і тіла людини на зображеннях і відеопотоку в реальному часі, на основі якого надалі буде побудований каскадний класифікатор для завдання, поставленої в цій роботі.

В алгоритмі Віоли-Джонса можна виділити чотири основні етапи:

- Переведення зображення в інтегральне уявлення;
- Використання ознак Хаара як дескрипторів;
- Використання алгоритму AdaBoost як класифікатор;
- Використання каскаду із класифікаторів.

Навчання даних класифікаторів відбувається дуже повільно, але детектуються об'єкти, що шукаються, дуже швидко. Цей метод працює на принципі «ковзного» вікна.

Щоб скоротити кількість операцій для обчислення ознак Хаара, вихідне зображення перетворюється на своє інтегральне уявлення. Це уявлення дозволяє досить швидко розрахувати сумарну яскравість прямокутника на зображенні незалежно від цього, якого розміру буде даний прямокутник. Інтегральне уявлення зображення – це матриця, розмір якої дорівнює вихідному зображенню, а значення її елементів розраховується так:

$$L(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j), \quad (2.7)$$

де $I(i, j)$ – це яскравість пікселя початкового зображення.

Кожен елемент матриці $L(x, y)$ – є сума пікселів у прямокутник $(0,0)(0,x)(x,y)(0,y)$. Розрахунок такої матриці займає лінійний час, який пропорційно кількості пікселів у цьому зображенні. Тому інтегральне уявлення розраховується повністю за прохід.

Наочне обчислення площі прямокутника в інтегральному зображенні показано на наступному прикладі:

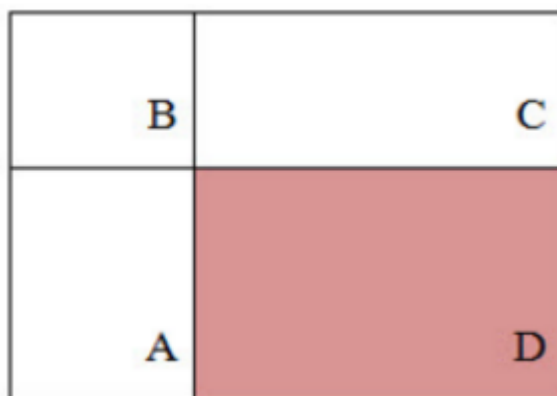


Рисунок 23 - Знаходження площі прямокутника ABCD в інтегральному поданні зображення

Нехай в інтегральному зображенні потрібно знайти площу прямокутника із вершинами ABCD. Це обчислення можна звести всього до трьох операцій.

$$S(ABCD) = L(D) - L(A) - L(C) + L(B) \quad (2.8)$$

Історично склалося, що алгоритми, які працюють лише з інтенсивністю зображення, мають дуже велику обчислювальну складність. Папагеоргиу у роботі розглянув безліч ознак, засновані на вейвлетах Хаара. Віола та Джонс адаптували цю ідею та розробили ознаки Хаара.

У класичному методі Віоли-Джонса використовуються прямокутні ознаки Хаара, які виглядають так:

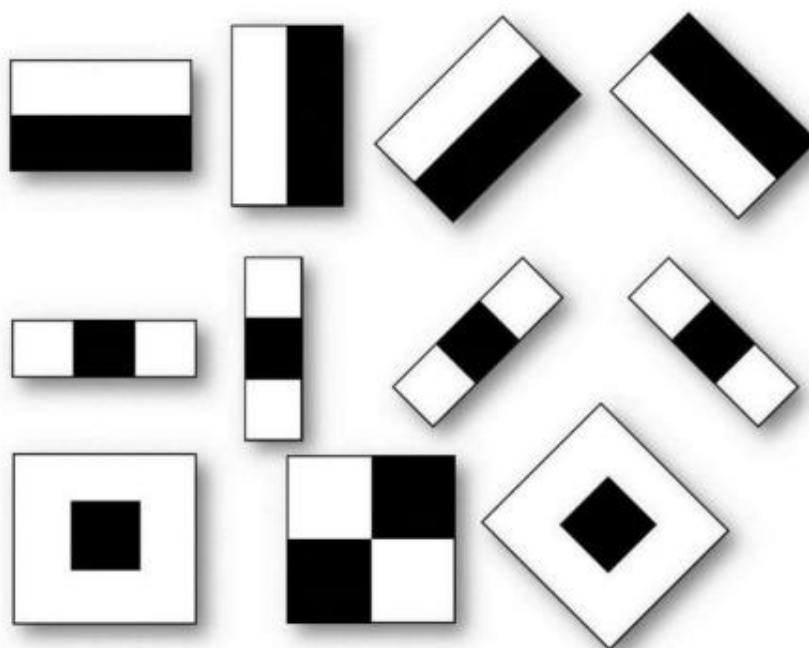


Рисунок 24 - Прямокутні ознаки Хаара

Ці ознаки ще називаються примітивами Хаара. У розширеному методі Віюлі-Джонса ці ознаки доповнилися ще декількома модифікаціями, продемонстрованими на малюнку далі.

Обчислюються дані ознаки наступним шляхом. Де X – це сума значень яскравостей пікселів зображення, що закриваються білою областю, а Y – чорною областю. Власне, для цього використовується інтегральне уявлення зображення, розглянуте вище.

$$F = X - Y \quad (2.9)$$

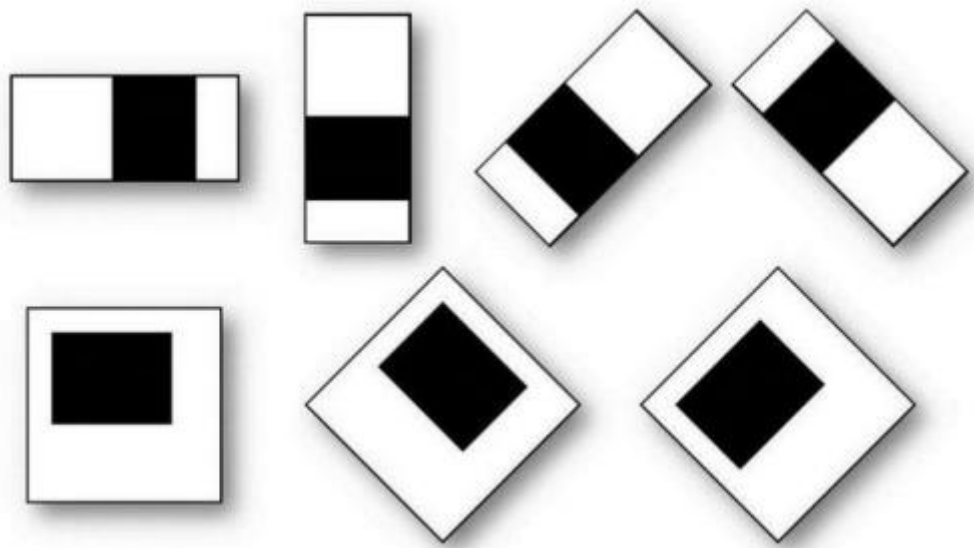


Рисунок 25 - Додаткові ознаки Хаара

Кількість різних варіацій ознак, розраховане для віконця 24x24, становить приблизно 160,000+, і багато хто з них марні для детектування того чи іншого об'єкта. Щоб вибрати корисні ознаки, використовується алгоритм AdaBoost, який буде розглянутий далі.

2.4 Алгоритм AdaBoost

Бустинг (від англ. boosting – підвищення, посилення, покращення) – це комплекс методів, що сприяє підвищенню точності аналітичних моделей. Модель називається «сильною», якщо припускає невелику кількість помилок класифікацій. «Слабка» модель, навпаки, робить велику кількість помилок і не дозволяє надійно розділити класи. Тому бустинг - це така процедура, яка в результаті послідовної побудови композиції алгоритмів машинного навчання прагне усунути недоліки всіх попередніх алгоритмів у кожному новому алгоритмі.

Найбільш досконалим алгоритмом бустингу є AdaBoost (Adaptive Boosting), запропонований 1999 р. Фройндом і Шапіром. Алгоритм AdaBoost адаптивний тому, що кожен наступний класифікатор будується за тими об'єктами, які були неправильно класифіковані минулими. Даний алгоритм чутливий до шуму в даних і викидів, а також він менш схильний до перенавчання.

Плюси AdaBoost:

- Досить проста реалізація;
- Дуже хороша узагальнююча здатність, яка може покращуватися зі збільшенням числа базових алгоритмів;
- Можливість виявити об'єкти, які є шумовими викидами;
- Власні накладні витрати цього алгоритму невеликі.

Мінуси AdaBoost:

- Вимагає досить довгі навчальні вибірки.
- Іноді будуються неоптимальні набори базових алгоритмів. Для цього можна часом повертатися до раніше побудованих алгоритмів та навчати їх наново.
- Іноді трапляється перенавчання за наявності досить високого рівня шуму даних, тобто. експоненційна функція, яка відповідає за втрати, дуже збільшує ваги «найважчих» об'єктів, на яких помиляються багато «слабких» алгоритмів. Найчастіше саме ці об'єкти виявляються шумовими викидами. В результаті чого алгоритм AdaBoost починає налаштовуватися на шум, що веде до перенавчання. Ця проблема вирішується шляхом видалення цих шумових викидів або застосування таких функцій втрат, які є менш «агресивними».

- Побудови досить великих і громіздких композицій, що з сотень «слабких» алгоритмів. Дані композиції виключають можливість змістовної інтерпретації, а також вимагають великих обсягів пам'яті для зберігання «слабких» алгоритмів і досить тимчасових витрат на обчислення класифікацій.

Після побудови класифікатора, на основі даного алгоритму, Віоле та Джонсу вдалося скоротити кількість обчислюваних ознак для віконця 24x24 з 160,000 до 6000. Але і це досить велика кількість ознак для детектування об'єктів у реальному часі. Тому було вирішено використати замість одного "сильного" класифікатора каскад із класифікаторів.

2.5 Каскад класифікаторів

Каскад із «сильних» класифікаторів – це якесь дерево прийняття рішень, в якому кожен вузол побудований таким чином, щоб розпізнавати практично всі об'єкти, що цікавлять, на зображенні і відхиляти області, в яких цих об'єктів немає. Крім того, вузли даного дерева прийняття рішень влаштовані таким чином, що чим ближче вузол розташований до кореня дерева, тим менше дескрипторів він містить, а отже, вимагає менше часу на прийняття рішення.

Дана каскадна модель класифікаторів дуже добре підходить для обробки зображень, на яких міститься порівняно невелике об'єктів, що детектуються. У цьому випадку класифікатор побудований таким чином може швидше прийняти рішення про те, що ця область не містить об'єкта, і перейти до наступної.

Каскад із класифікаторів можна представити у вигляді наступної схеми:

- На вхід подається одна з областей оброблюваного зображення і до неї застосовується найпростіший «сильний» класифікатор з каскаду.
- Якщо він вирішив, що шуканого об'єкта там точно немає, то ця область відкидається і на вхід подається така.
- Якщо перший класифікатор прийняв рішення, що швидше за все об'єкт присутній у цій галузі, то вона подається на наступний класифікатор, який більш пристосований і складніший за минуле.
- Цей процес триває доти, доки область із зображенням не надійде в обробку на останній найскладніший «сильний» класифікатор з каскаду, який ухвалить остаточне рішення про те, чи є там об'єкт, чи ні.

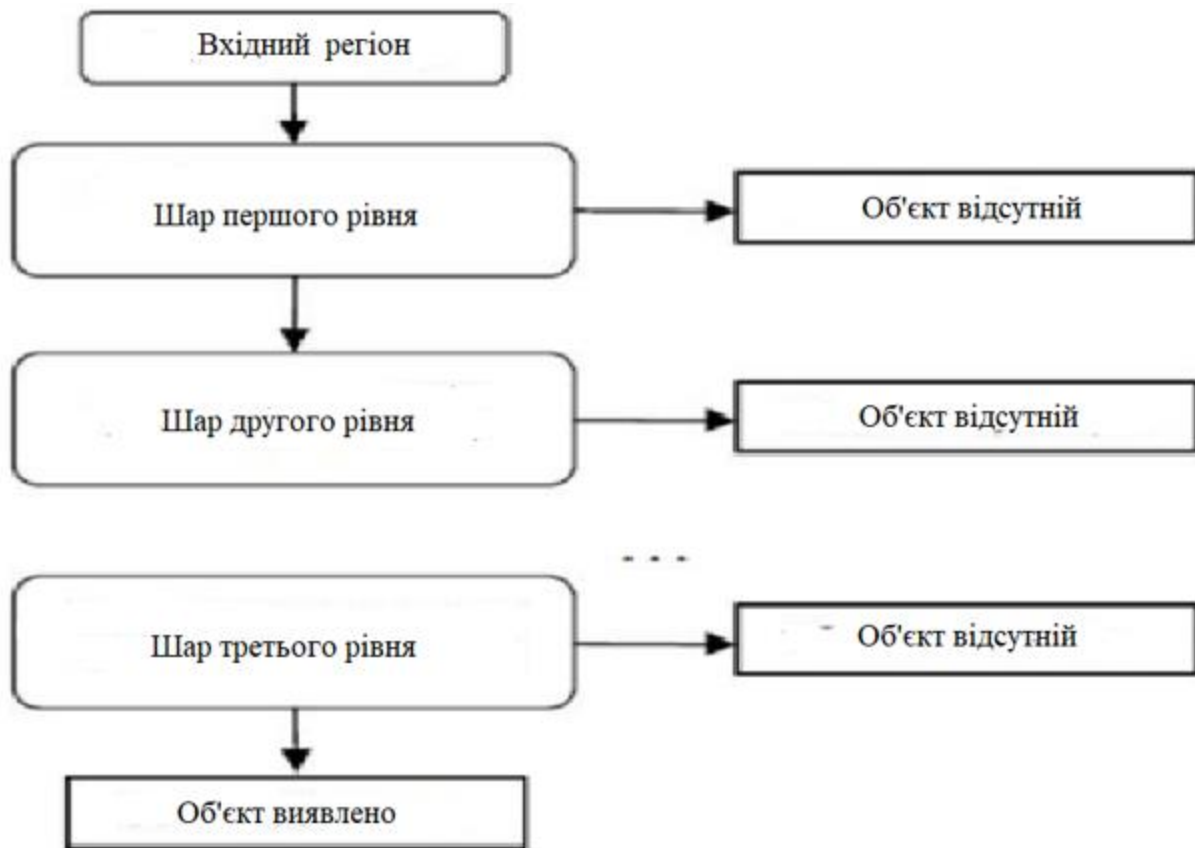


Рисунок 26 - Приклад каскадної моделі "сильних" класифікаторів

Для тренування такого каскадного класифікатора потрібні такі дії:

- Встановлення значення рівня помилок кожного класифікатора з каскаду. Дані параметри називаються detection та false positive rates. Потрібно щоб рівень detection був високим, а false positive rates низьким.
- Додавання дескрипторів до тих пір, поки параметри для класифікатора, що навчається, з каскаду не досягнуть поставленого рівня.
- Якщо false positive rates високий, додається наступний шар.
- Неправдиві виявлення, знайдені на поточному шарі, використовуються як негативні на наступному шарі.

У результаті завдяки цьому підходу, Віолі та Джонсу вдалося отримати досить хороший і швидкий класифікатор для детектування осіб.

3. ПОБУДУВАННЯ КАСКАДНОГО КЛАСИФІКАТОРУ ДЛЯ ЛОКАЛІЗАЦІЇ АВТОМОБІЛІВ І ЙОГО ТЕСТУВАННЯ

3.1 Підготовка навчальної та тестової вибірки

В якості набору даних із супутників знімків було обрано відкриту базу COWC (Cars Overhead With Context). Ця база складається з великої кількості анотованих автомобілів. Всього представлено шість регіонів міського типу: Торонто – Канада, Селуїн – Нова Зеландія, Постдам та Файнген – Німеччина, Коламбус та Юта – США.

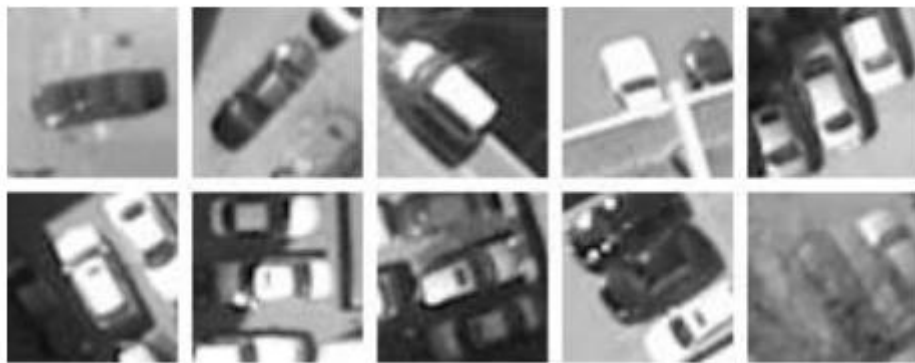


Рисунок 27 - Приклади зображень із позитивної вибірки

Як дослідження було обрано набір даних, що належить Торонто. Всі зображення в даній мережі мають розмір 256x256 пікселів з роздільною здатністю 15см на піксель. З цих зображень була нарізана позитивна вибірка, що складається з 10 000 зображень 48x48 пікселів, що містять автомобіль. Як негативна вибірка, що не містить автомобіль, спочатку були нарізані 20000 зображень 50x50 пікселів з досить складними об'єктами фону, потім 18000 зображень 190x190 пікселів зі складними об'єктами та простими.



Рисунок 28 - Приклади зображень із негативної вибірки

З отриманих позитивної та негативних вибірок були складені навчальна та тестова:

- Навчальна вибірка складається з 7000 позитивних та 17000 негативних прикладів.
- Тестова вибірка складається з 1500 позитивних та 500 негативних прикладів.

3.2 Структура каскадного класифікатора

Спочатку зображення переводилося з RGB у градацію сірого. Як дескриптори були обрані локальні бінарні шаблони (LBP), які повинні добре підходять для поставленого завдання. Ці ознаки реалізовані в бібліотеці OpenCV.

Локальні бінарні шаблони – це деякі бінарні коди певної розрядності, які застосовуються для класифікації в комп'ютерному зорі. Дані ознаки інваріантні, як і змін в умовах освітлення, так і до поворотів зображення.

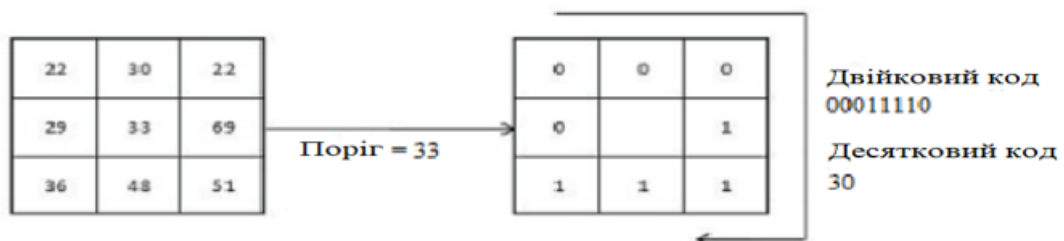


Рисунок 29 - Базовий локальний бінарний шаблон

ЛБШ визначає околицю пікселя зображення у двійковому уявленні. Щоб обчислити в деякій точці зображення базовий ЛБШ використовуються значення інтенсивності восьми пікселів, що оточують дану точку, а як поріг приймається значення інтенсивності в цій точці. Якщо інтенсивність пікселя більша або дорівнює порогу, то вони набувають значення «1», що залишилися «0». В результаті чого виходить восьмирозрядний бінарний код, що описує околицю даної точки.

Також, для більш гнучкого аналізу текстурних особливостей зображення будуються розширені ЛШ з довільним числом точок і радіусом, завдяки круговій околиці та білінійної інтерполяції значень інтенсивності пікселів.

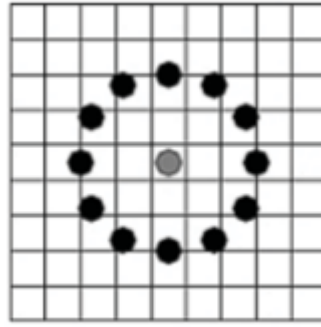


Рисунок 30 - Розширений локальний бінарний шаблон

Деякі види бінарних кодів несуть у собі більше інформації, ніж інші. Йдеться про рівномірні локальні шаблони – шаблони, які містять не більше трьох серій «0» та «1» (наприклад, 001110000). РЛБШ визначають найважливіші локальні особливості зображення. Також вони забезпечують економію пам'яті.

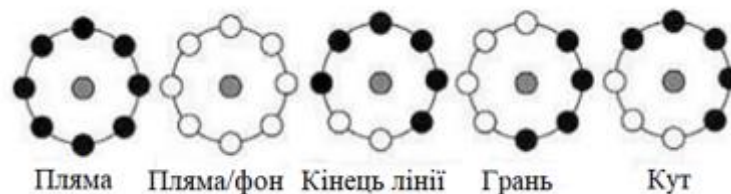


Рисунок 31 - Локальні особливості, що детектуються РЛБШ

ЛБШ обчислюються всім пікселів зображення. Далі будується гістограма, в якій усім РЛБШ відповідає окремий стовпець. Також враховується і число нерівномірних ЛБШ, значення якого записується в останній стовпець гістограми.

Оскільки побудови даного класифікатора було обрано ЛБШ ознаки, то переведення зображення у його інтегральне уявлення непотрібен.

Як алгоритм класифікації був обраний AdaBoost, як і в методі Віола-Джонса. Надалі мною було застосовано чотири типи цього алгоритму, які також реалізовані в OpenCV: дискретний AdaBoost (DAB), речовий AdaBoost (RAB), LogitBoost (LB), гладкий AdaBoost (GAB).

Також було застосовано каскадний підхід для побудови класифікатора з метою скорочення часу розпізнавання об'єкта.

3.3 Навчання та тестування класифікаторів

Для початку було проведено навчання на вибірці з 3000 позитивних (розмір рухомого вікна було обрано 48x48) та 7000 жорстких негативних (розмір 50x50) прикладів. Як бустинг-класифікатор був обраний GAB (Gentle AdaBoost). Каскад складався із 8 етапів. Але цей підхід показав себе з поганого боку, як і за швидкістю навчання та детектування, так і за якістю.



Рисунок 32 - Приклад обробленого зображення

Детектування автомобілів на даному знімку (розмір 2048x2048 пікселів) зайняло близько 1 хв і було отримано дуже велику кількість TN.

Як негативна вибірка далі були використані зображення (190x190 пікселів), що містять, як і складні об'єкти фону, так і прості.

Далі на вибірці з 4000 позитивних і 10000 негативних прикладів були побудовані 4 класифікатори з різним розміром вікна: 30x30, 36x36, 40x40 і 48x48. Як бустингкласифікатор всі вони використовували GAB. Каскади у трьох класифікаторів склалися з 16 етапів, в одного з 13, оскільки задана точність була досягнута раніше.

Отримані класифікатори були протестовані на тестовій вибірці з 1500 позитивних та 500 негативних прикладів. Результати даного тестування наведено у таблиці 3.

Таблиця 3 - Вибір розміру рухомого вікна

Тип AdaBoost	Розмір вікна	Кількість етапів у каскаді	TP	FN	FP	F1
GAB	48X48	16	1413	87	83	0.95
GAB	40X40	16	1407	93	93	0.94
GAB	36X36	13	1371	129	102	0.91
GAB	30X30	16	1178	322	23	0.86

Найкраще за якістю себе показав перший класифікатор, але швидкість детектування була набагато нижчою від другого, який майже не поступається за якістю першому. Результати решти двох класифікаторів були гіршими за другий. Тому для подальшого дослідження було вибрано рухоме вікно розміром 40x40 пікселів.

Наступним етапом дослідження була побудова класифікаторів з різним типом бустинг-алгоритму, з уже вибраним раніше розміром «ковзного» вікна.

На тій же навчальній вибірці було навчено ще три класифікатора, з іншими типами AdaBoost: DAB, RAB та LB. Каскади склалися із 16 етапів. Найдовше навчався DAB. Отримані класифікатори протестували на тестовій вибірці.

Таблиця 4 - Вибір алгоритму AdaBoost

Тип AdaBoost	Розмір вікна	Кількість етапів у каскаді	TP	FN	FP	F1
GAB	40X40	16	1407	93	93	0.94
DAB	40X40	16	1126	374	314	0.79
RAB	40X40	13	1314	186	151	0.88
LB	40X40	16	1152	348	297	0.78

Найкраще показав себе класифікатор з алгоритмом типу GAB, який був обраний спочатку.

Далі, був змінений параметр, що відповідає за максимальний бажаний рівень помилкових спрацьовувань для кожного етапу каскаду, який спочатку

дорівнював 0.5. Було навчено ще 2 класифікатори, з іншими значеннями цього параметра, а саме 0.4 та 0.3.

Зміна даного параметра не призвела до помітного покращення якості розпізнавання.

Таблиця 5 - Зміна параметра max false alarm rate

Max false alarm rate	Розмір вікна	Кількість етапів у каскаді	TP	FN	FP	F1
0.5	40X40	16	1407	93	93	0.94
0.4	40X40	16	1396	104	85	0.93
0.3	40X40	13	1388	112	82	0.93

Останнім кроком було збільшення навчальної вибірки. На вибірці з 7000 позитивних і 17000 негативних прикладів було побудовано класифікатор з розміром вікна, що «ковзає» 40x40. Як бустинг-класифікатор використовувався GAB, обраний раніше. Каскад складався з 16 етапів.

Таблиця 6 - Порівняння класифікаторів, навчених на різних вибірках

Тип <u>AdaBoost</u>	Розмір вікна	Кількість етапів у каскаді	TP	FN	FP	F1
GAB	40X40	16	1407	93	93	0.94
GAB	40X40	16	1415	85	73	0.95

Завдяки збільшенню навчальної вибірки міра F1 збільшилася до 0.95.



Рисунок 33 - Порівняння роботи класифікаторів, навчених на різних вибірках

Далі, як приклад, отриманий класифікатор був застосований до двох зображень з різною щільністю автомобілів. Зі своїм завданням він справився досить добре, при порівняно невеликій навчальній вибірці.

Цей класифікатор, наприклад, зміг виявити автомобілі, заховані в тіні будівлі, але не зміг знайти автомобіль за деревом. Якщо повернутися до знімка, до якого був застосований перший навчений класифікатор, і застосувати останній, то можна побачити помітну різницю як розпізнавання. На першому зображенні занадто багато помилкових спрацьовувань і майже немає позитивних. На другому ж майже всі автомобілі були виявлені, але також були і незначні неправдиві спрацьовування через те, що зображення з даним типом місцевості не були присутніми в навчальній вибірці.



Рисунок 34 - Фрагмент обробленого знімку

Дослідивши цей підхід до локалізації автотранспортних засобів на аерокосмічних зображеннях, було з'ясовано, що оптимальний розмір «ковзного» вікна для роздільної здатності 15см на піксель повинен становити 40x40 пікселів. Також було встановлено, що з чотирьох типів AdaBoost алгоритмів, реалізованих в OpenCV, найкраще справляється із поставленим завданням GAB. Зміна параметра, який відповідає за максимальний бажаний рівень помилкових спрацьовувань для кожного етапу каскаду, особливо не вплинув на якість розпізнавання

ВИСНОВОК

В результаті виконання магістерської роботи були детально розглянуті дві нейронні мережі DetectNet та YOLO, розглянуто застосування мережі YOLO у межах завдання розпізнавання об'єктів.

Було розглянуто метод Віюли-Джонса, який непогано себе зарекомендував у задачі пошуку осіб людей на зображеннях та відеопотоці. На основі його принципів було побудовано каскадний класифікатор для локалізації автомобілів на супутникових зображеннях. Для вирішення поставленого завдання було проведено модифікацію даного підходу. У якості дескриптора використовувалися не ознаки Хаара, а локальні бінарні шаблони. У ході проведеного дослідження, яке полягало в побудові каскадних класифікаторів при різних параметрах, було отримано оптимальне рішення.

Замість стандартного DAB, який використовувався у методі Віюли-Джонса, було прийнято рішення використовувати GAB.

На додаток до всього, для збільшення швидкості розпізнавання, були обрані певні рамки розміру об'єктів, що розпізнаються, через їх невеликі зміни (усі автомобілі майже однакового розміру) на зображенні і однорідності даних.

ПЕРЕЛІК ПОСИЛАНЬ

1. Hu J., You S., and Neumann U. Integrating lidar, aerial image and ground images for complete urban building modeling. Third International Symposium on 3D Data Processing, Visualization, and Transmission, pp. 184–191, June 2006.
2. Liu K., Mattyus G. Fast Multiclass Vehicle Detection on Aerial Images. *IEEE Geosci. Remote Sens. Lett.* 2015;12:1–5.
3. Chen Z., Wang C., Luo H., Wang H. Vehicle Detection in HighResolution Aerial Images Based on Fast Sparse Representation Classification and Multiorder Feature. *IEEE Trans. Intell. Transp. Syst.* 2016; 17: 2296–2309.
4. Crawford, J. Beyond supply and demand: Making the invisible hand visible. Re-Work Deep Learning Summit, San Francisco. 2016.
5. Van de Sande K.E.A., Uijlings J.R.R., Gevers T., Smeulders A.W.M. Segmentation as selective search for object recognition. Proceedings of the International Conference on Computer Vision; Barcelona, Spain. 2011; pp. 1879– 1886.
6. Alexe B., Deselaers T., Ferrari V. What is an object? Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); San Francisco, CA, USA. 2010; pp. 73–80.
7. Uijlings J.R.R., Sande K.E.A.V.D., Gevers T., Smeulders A.W.M. Selective Search for Object Recognition. *Int. J. Comput. Vis.* 2013; 104: 154–171.
8. Kuo W., Hariharan B., Malik J. DeepBox: Learning Objectness with Convolutional Networks. Proceedings of the IEEE International Conference on Computer Vision; Los Alamitos, CA, USA. 2015; pp. 2479–2487.
9. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv.* 2016.
10. Girshick R., Donahue J., Darrell T., Malik J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 2015; 38: 1.
11. Girshick R. Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision; Los Alamitos, CA, USA. 2015.

12. Christian S., Wei L., Yangqing J., Pierre S., Scott R., Dragomir A., Dumitru E., Vincent V., Rabinovich A.; Going Deeper with Convolutions. 2014.
13. Jie H., Weilong Y. A multi-size convolution neural network for RTS games winner prediction. *MATEC Web of Conferences*, 2018.
14. Hoiem D., Chodpathumwan Y., and Dai Q., Diagnosing Error in Object Detectors. ECCV. 2012.
15. Redmon J., Farhadi A. YOLO9000: Better, Faster, Stronger, 2016.
16. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection, 2016.
17. Viola P., Jones M.J. Rapid Object Detection using a Boosted Cascade of Simple Features. Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2001.
18. Papageorgiou, Oren, Poggio, A general framework for object detection. International Conference on Computer Vision, 1998.
19. Freund Y., Schapire R.E. A Short Introduction to Boosting. Shannon Laboratory, USA, 1999., pp. 771-780.
20. Liao S., Zhu X., Lei Z., Zhang L., Stan Z. Li. Learning Multi-scale Block Local Binary Patterns for Face Recognition. International Conference on Biometrics (ICB), 2007, pp. 828-837.
21. Madokoro H., Kainuma A., Sato K. Non-Rectangular RoI Extraction and Machine Learning Based Multiple Object Recognition Used for Time-Series Aerial Images Obtained Using MAV. Proceedings of 22nd International Conference on KnowledgeBased and Intelligent Information & Engineering Systems, 2018. – pp. 462–471.
22. Mayer H. Automatic object extraction from aerial imagery - a survey focusing on buildings. *Computer Vision and Image Understanding*. 1999. 74(2): pp. 138–149.
23. Lin C. and Nevatia R. Building detection and description from a single intensity image. *Computer Vision and Image Understanding*. 1998. Vol. 72, No. 2, pp. 101-121(21).

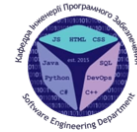
24. Kai G., Shuai W., Yong X. Face recognition using both visible light image and near-infrared image and a deep network. Proceedings of CAAI Transactions on Intelligence Technology, 2017. – pp. 39–47.
25. Li S., Yi D., Lei Z., Liao S. The CASIA NIR-VIS 2.0 face database. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. (CVPRW), 2013. – pp. 348-353.
26. Yong X., Aini Z., Jian Y., Zhang D. Bimodal biometrics based on a representation and recognition approach. Proceedings of SCI, 2011. – p. 50.
27. Namatevs I., Sudars K., Polaka I. Automatic data labeling by neural networks for the counting of objects in videos. Proceedings of ICTE in Transportation and Logistics (ICTE 2018), 2018. – pp. 151–158.
28. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A. C. SSD: Single Shot MultiBox Detector. Proceedings of European Conference on Computer Vision (ECCV), 2016. – pp. 21-37.
29. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, 2018. – pp. 4510-4520.

Cordts M., Omran M., Ramos S., Rehfeld T., Enzweiler M., Benenson R., Franke U., Roth S., Schiele B. The Cityscapes Dataset for Semantic Urban Scene Understanding. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. – pp. 3213-3223.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ



Кафедра інженерії програмного забезпечення

МАГІСТЕРСЬКА РОБОТА «Реалізація інтелектуальної системи розпізнавання об'єктів із застосуванням нейронних мереж»

Виконав: студент 6 курсу, групи ПДМ-61 спеціальності
121 Інженерія програмного забезпечення, Ільюкевич Гліб
Олегович
Керівник: доктор філософії, доцент кафедри інженерії
програмного забезпечення
Дібрівний О. А.

Київ - 2021

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: спрощення процесу розпізнавання об'єктів за рахунок використання моделей нейронних мереж для аналізу даних.

Об'єкт дослідження: процес розпізнавання об'єкту на зображеннях, що використовує нейронні мережі.

Предмет дослідження: технології аналізу даних за допомогою нейромереж.

АКТУАЛЬНІСТЬ РОБОТИ

Недоліки існуючих процесів

- Наявність рішень з більш якісним детектуванням

Показники ефективності для оцінки нового рішення

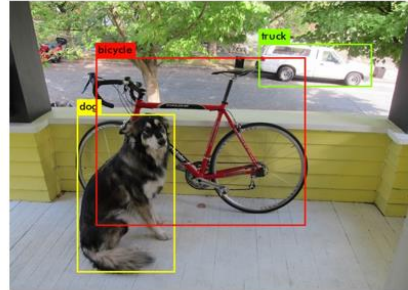
- Час швидкості розпізнавання
- Точність та повнота

АНАЛІЗ ІСНУЮЧИХ ІТ-РІШЕНЬ ТА МОДЕЛЕЙ

DIGITS DetectNet

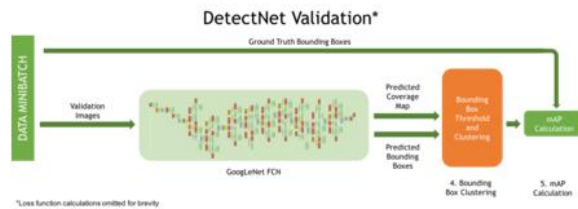
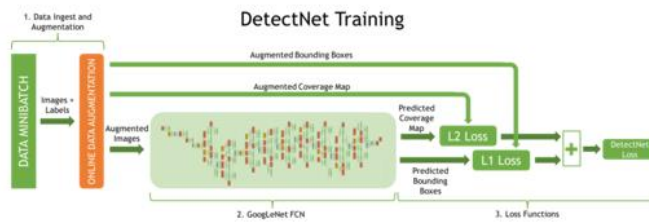


YOLO



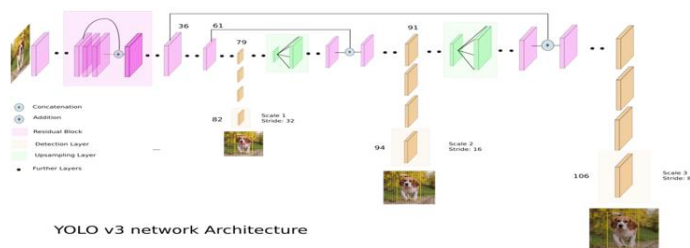
АНАЛІЗ ІСНУЮЧИХ АРХІТЕКТУР

DetectNet



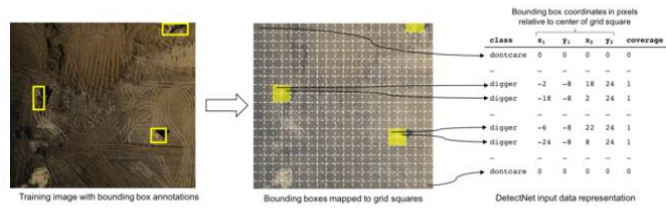
АНАЛІЗ ІСНУЮЧИХ АРХІТЕКТУР

YOLO



YOLO v3 network Architecture

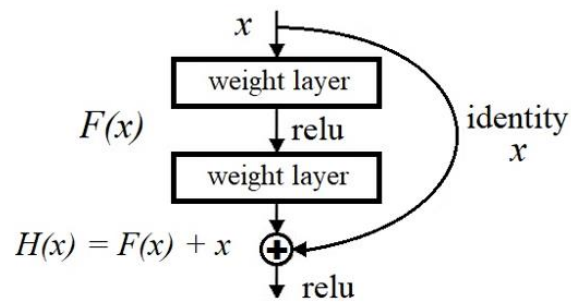
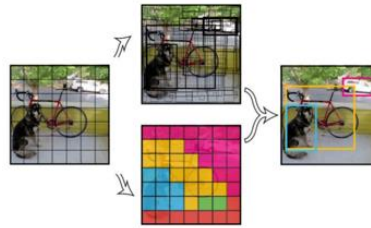
DetectNet



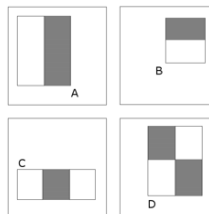
$$\frac{1}{2N} \sum_{i=1}^N |coverage_i^t - coverage_i^p|^2 \quad (6.1)$$

$$\frac{1}{2N} \sum_{i=1}^N [|x_1^t - x_1^p| + |y_1^t - y_1^p| + |x_2^t - x_2^p| + |y_2^t - y_2^p|] \quad (6.2)$$

YOLO



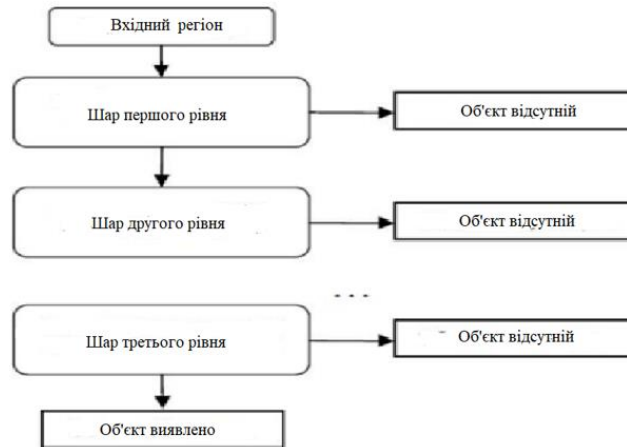
МЕТОД ВІОЛІ-ДЖОНСОНА



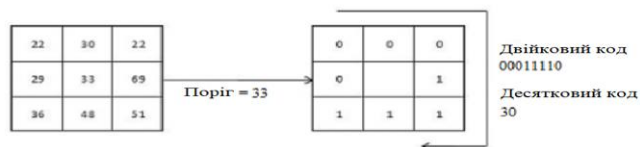
Основні етапи методу:

- Переведення зображення в інтегральне уявлення.
- Використання ознак Хаара у якості дескрипторів.
- Використання алгоритму AdaBoost як класифікатора.
- Використання каскаду із класифікаторів.

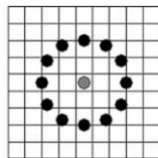
КАСКАД КЛАСИФІКАТОРІВ



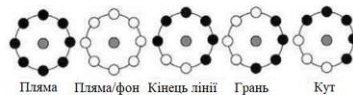
ДЕСКРИПТОР



Базовий локальний бінарний шаблон



Розширений локальний бінарний шаблон



Локальні особливості, що детектуються РЛБШ

НАВЧАННЯ ТА ТЕСТУВАННЯ КЛАСИФІКАТОРІВ

Тип AdaBoost	Розмір вікна	Кількість етапів у каскаді	TP	FN	FP	F1
GAB	48X48	16	1413	87	83	0.95
GAB	40X40	16	1407	93	93	0.94
GAB	36X36	13	1371	129	102	0.91
GAB	30X30	16	1178	322	23	0.86

Вибір розміру вхідного вікна

Тип AdaBoost	Розмір вікна	Кількість етапів у каскаді	TP	FN	FP	F1
GAB	40X40	16	1407	93	93	0.94
DAB	40X40	16	1126	374	314	0.79
RAB	40X40	13	1314	186	151	0.88
LB	40X40	16	1152	348	297	0.78

Вибір алгоритму AdaBoost

ПОРІВНЯННЯ КЛАСИФІКАТОРІВ

Тип <u>AdaBoost</u>	Розмір вікна	Кількість етапів у каскаді	TP	FN	FP	F1
GAB	40x40	16	1407	93	93	0.94
GAB	40x40	16	1415	85	73	0.95

Порівняння класифікаторів, навчених на різних вибірках



Порівняння роботи класифікаторів, навчених на різних вибірках

ВИСНОВКИ

В результаті виконання магістерської роботи були детально розглянуті дві нейронні мережі DetectNet та YOLO, розглянуто застосування мережі YOLO у межах завдання розпізнавання об'єктів.

Було розглянуто метод Віолі-Джонса, який непогано себе зарекомендував у задачі пошуку осіб людей на зображеннях та відеопотоці. На основі його принципів було побудовано каскадний класифікатор для локалізації автомобілів на супутникових зображеннях. Для вирішення поставленого завдання було проведено модифікацію даного підходу. У якості дескриптора використовувалися не ознаки Хаара, а локальні бінарні шаблони. У ході проведеного дослідження, яке полягало в побудові каскадних класифікаторів при різних параметрах, було отримано оптимальне рішення.

Замість стандартного DAV, який використовувався у методі Віолі-Джонса, було прийнято рішення використовувати GAB.

На додаток до всього, для збільшення швидкості розпізнавання, були обрані певні рамки розміру об'єктів, що розпізнаються, через їх невеликі зміни (усі автомобілі майже однакового розміру) на зображенні і однорідності даних.

АПРОБАЦІЯ

Статті:

1. Льюкевич Г. О. Класифікація інтелектуальних систем для аналізу даних та вимоги до них// COMPUTER SYSTEMS AND INFORMATION TECHNOLOGIES. №2, 2021.

Тези доповідей:

1. Льюкевич Г. О. Інтелектуальні системи для полегшення процесу і оптимізації завдань розпізнавання об'єктів.
// XIII Міжнародна науково-технічна конференція студентства та молоді «Світ інформації та телекомунікацій». – Київ: ДУТ, 2021.

ДЯКУЮ ЗА УВАГУ!