

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації
та мехатроніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

РОЗРОБЛЕННЯ МЕТОДУ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ ДЛЯ МОБІЛЬНИХ
РОБОТІВ НА БАЗІ АСОЦІАТИВНОЇ ЛОГІКИ
(тема)

Виконав: студент 2 курсу, групи КІТПВм-20-2
Радченко Яків Олегович
(прізвище, ім'я, по батькові)

Спеціальності 151 Автоматизація та
комп'ютерно-інтегровані технології
(код і повна назва напрямку)

Тип програми Освітньо-професійна
Освітня програма Комп'ютерно-інтегровані
технологічні процеси і виробництва
(назва)

Керівник проф. каф. КІТАМ Євсєєв В.В.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри КІТАМ

(підпис)

Невлюдов І.Ш.
(прізвище, ініціали)

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Автоматики і комп'ютеризованих технологій

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки

Рівень вищої освіти другий (магістерський)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології

Тип програми Освітньо-професійна

Освітня програма Комп'ютерно-інтегровані технологічні процеси і виробництва
(назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри КІТАМ

(підпис)
« ____ » _____ 2021 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Радченку Якову Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення методу ідентифікації об'єктів для мобільних робіт на базі асоціативної логіки

Затверджена наказом по університету від 08.11.2021 р. № 1697 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 09.12.2021 р.

3. Вихідні дані до роботи: 3.1 Операційна система Windows/Linux; 3.2 Система виявлення YOLO + OpenCV; 3.3 Середовище розробки PyCharm/Brackets; 3.4 Камера NexiGo N960E 1080P 60FPS; 3.5 Мова програмування Python.

4. Перелік питань, що потрібно опрацювати в роботі: 4.1 Вступ; 4.2 Аналіз особливостей асоціативних процесів та розпізнавання образів; 4.3 Приклад дії асоціації з урахуванням кластеризації; 4.4 Порівняння різних алгоритмів кластеризації; 4.5 Розрахунки методів розпізнавання образів, основа штучного нейрона; 4.6 Імплементация мовою програмування Python; 4.7 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плака комп'ютерних ілюстрацій (слайдів)

Демонстраційний матеріал представлений у форматі презентації

PowerPoint (*.ppt) – 19 с. форматі

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз особливостей асоціативних процесів та способів розпізнавання образів	08.11 – 11.11.21	виконано
2	Приклад дії асоціації з урахуванням кластеризації	12.11 – 14.11.21	виконано
3	Порівняння різних алгоритмів кластеризації	15.11 – 17.11.21	виконано
4	Аналіз нейромережевої моделі асоціативного процесора та пам'яті	17.11 – 18.11.21	виконано
5	Математичні розрахунки методів розпізнавання образів, основа штучного нейрона	18.11 – 19.11.21	виконано
6	Розв'язання математичної нерівності вагомої функції	20.11 – 21.11.21	виконано
7	Імплементация програми з урахуванням методу асоціативних процесів	22.11 – 23.11.21	виконано
8	Виявлення об'єктів за допомогою YOLO	24.11 – 26.11.21	виконано
9	Експериментальні дослідження	26.11 – 27.11.21	виконано

Дата видачі завдання 8 листопада 2021 р.

Студент _____
(підпис)

Радченко Я.О.
(прізвище, ініціали)

Керівник роботи _____
(підпис)

професор каф. КІТАМ Євсєєв В.В.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 74 с., 1 табл., 36 рис., 2 дод., 24 джерела.

WINDOWS, PYTHON, PYCHARM, OPENCV, НЕЙРОННІ МЕРЕЖІ, YOLO, СИСТЕМА ВИЯВЛЕННЯ ОБРАЗІВ, АСОЦІАТИВНА ЛОГІКА.

Об'єкт дослідження – інтелектуальні системи прийняття рішень в робототехніці.

Предмет дослідження – моделі та методи асоціативної логіки.

Методи дослідження – методи асоціативної логіки на базі нейронних мереж за допомогою системи YOLO.

Мета кваліфікаційної роботи – імплементація програми ідентифікації об'єктів в промисловій зоні для мобільних роботів на базі нейронних мереж з правилами асоціативної логіки.

У кваліфікаційній роботі проаналізовано особливості асоціативних процесів, спосіб розпізнавання образів та приклади застосування в роботизованих системах, порівняно різні алгоритми кластеризації. Також зроблено математичний аналіз методів розпізнавання образів з урахуванням правил асоціативної логіки.

На базі проведеного аналізу було доведено актуальність даних досліджень, проведено аналіз особливостей асоціативних процесів, зроблено порівняння різних алгоритмів кластеризації, вибрано метод YOLO для написання програми, розроблено програму базуючись на YOLO та бібліотекою OpenCV, проведено експериментальні дослідження залежності розробленої програми та її переваги.

ABSTRACT

The explanatory note contains 74 p., 1 tables, 36 drawings, 2 pp., 24 sources.

WINDOWS, PYTHON, PYCHARM, OPENCV, NEURAL NETWORKS, YOLO, IMAGE DETECTION SYSTEM, ASSOCIATIVE LOGIC.

The object of research is intelligent decision-making systems in robotics.

Subject of research – models and methods of associative logic.

Research methods – methods of associative logic based on neural networks using the YOLO system.

The purpose of the qualification work is to implement a program for the identification of objects in the industrial zone for mobile robots based on neural networks with the rules of associative logic.

The qualification work analyzes the features of associative processes, the method of pattern recognition and examples of application in robotic systems, relatively different clustering algorithms. Mathematical analysis of image recognition methods taking into account the rules of associative logic is also made.

On the basis of the analysis the relevance of research data was proved, the peculiarities of associative processes were analyzed, different clustering algorithms were compared, YOLO method for writing a program was chosen, the program based on YOLO and OpenCV library was developed, experimental researches of program dependence and its advantages were conducted.

ЗМІСТ

Перелік умовних скорочень	8
Вступ	9
1 Аналіз особливостей асоціативних процесів та спосіб розпізнавання образів, приклади застосування в роботизованих системах	11
1.1 Впровадження в асоціативні процеси та спосіб розпізнавання образів	11
1.2 Класифікація задач розпізнавання образів	14
1.3 Систематизація розпізнавання образів	16
1.4 Принцип дії комунікативних та асоціативних процесів	17
1.5 Аналіз функціональної структури робота	19
1.6 Нейромережеві моделі асоціативного процесора та пам'яті ...	23
1.7 Інтелектуальні системи та різні методи представлення знань .	25
1.8 Приклад дії асоціації з урахуванням кластеризації	27
1.9 Порівняння різних алгоритмів кластеризації	29
1.10 Постановка завдання дослідження	32
1.11 Висновки до першого розділу	33
2 Математичні розрахунки методів розпізнавання образів, основа роботи штучного нейрона	34
2.1 Розв'язання математичної нерівності вагової функції	34
2.2 Висновки до другого розділу	42
3 Імплементация програми з урахуванням методу асоціативних процесів	43
3.1 Впровадження в YOLO. Виявлення об'єктів за допомогою YOLO	43
3.2 Уніфікація окремих компонентів виявлення об'єктів та поєднання в єдину нейронну мережу	45
3.3 Порівняння YOLO з іншими системами виявлення об'єктів та його обмеження	47

3.4 Розробка програми ідентифікації об'єктів мовою програмування Python за допомогою бібліотеки OpenCV з урахуванням всіх правил асоціативної логіки	50
3.5 Проведення експерименту	56
3.6 Охорона праці	61
3.7 Висновки до третього розділу	63
Висновки	64
Перелік джерел посилання	66
Додаток А Код програми для ініціалізації об'єктів на зображенні та їх ідентифікація в реальному часі	70
Додаток Б Демонстраційний матеріал у вигляді презентації	73

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ЕОМ – електронно-обчислювальна машина;

ЕС – експертна система;

КІС – комплексні інтелектуальні системи;

НМ – нейронна мережа;

НС – нейромережна система;

ПД – прогноз довіри;

РО – розпізнавання образів;

СС – самонавчальна система;

УА – уніфікована архітектура;

УМ – уніфікована модель;

ШС – штучна система;

IOU – Input output unit.

ВСТУП

На сьогоднішній день не можливо уявити сучасне виробництво без застосування інтелектуальних систем зору. Для побудови системи безпеки, системи контролю виробничого продукту або ж просту іграшку для дитини, в кожній сфері відбувається впровадження технологій комп'ютерного зору, спосіб асоціацій, спосіб навчання, а також самонавчання розробленої системи. Яка буде контролювати кожний процес і дасть поштовх в інших сферах науки, де проста людина не зможе так чітко відпрацювати, дослідити або ж надати контроль та безпеку.

Принцип асоціативної логіки дає можливість подивитись на розробку з іншої сторони, з простої. Саме правила асоціації є тим, що зараз потрібно при розробці ідентифікації об'єктів для мобільних роботів. Даний спосіб дозволить легше дивитись на проблему розробки та більше на питання самонавчання системи в подальшому використанні її.

Об'єкт дослідження – інтелектуальні системи прийняття рішень в робототехніці.

Предмет дослідження – моделі та методи асоціативної логіки.

Методи дослідження – методи асоціативної логіки на базі нейронних мереж за допомогою системи YOLO.

Мета кваліфікаційної роботи – імплементація програми ідентифікації об'єктів в промисловій зоні для мобільних роботів на базі нейронних мереж з правилами асоціативної логіки.

У кваліфікаційній роботі проаналізовано особливості асоціативних процесів, спосіб розпізнавання образів та приклади застосування в роботизованих системах, порівняно різні алгоритми кластеризації. Також зроблено математичний аналіз методів розпізнавання образів з урахуванням правил асоціативної логіки.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- провести аналіз особливостей асоціативних процесів та способів розпізнавання образів;
- провести аналіз дії асоціації з урахуванням кластеризації;
- порівняти алгоритми кластеризації;
- зробити розрахунки методів розпізнавання образів з використання правил асоціативних процесів;
- проаналізувати приклади розв’язання математичних нерівностей вагомої функції;
- поєднати систему YOLO з бібліотекою OpenCV для подальшої роботи;
- імплементувати програму з урахуванням методу асоціативних процесів;
- розробити програму виявлення об’єктів в реальному часі;
- провести експериментальне дослідження на розробленій програмі.

Атестаційна робота виконана згідно ДСТУ 3008 – 15 [1] та керуючись навчальним посібником з дипломного проекту [2] та методичними вказівками [3] і [4].

1 АНАЛІЗ ОСОБЛИВОСТЕЙ АСОЦІАТИВНИХ ПРОЦЕСІВ ТА СПОСІБ РОЗПІЗНАВАННЯ ОБРАЗІВ, ПРИКЛАДИ ЗАСТОСУВАННЯ В РОБОТИЗОВАНИХ СИСТЕМАХ

1.1 Впровадження в асоціативні процеси та спосіб розпізнавання образів

Асоціативні процеси – термін «асоціація» був об’єктом вивчення протягом багатьох століть. Аналізом займалися філософи, психологи, лінгвісти, науковці. У сучасній науці «асоціація» використовується в усіх сферах життєдіяльності людини і є суб’єктивним образом існуючих поза свідомістю людини зв’язків між явищами і предметами реального світу.

Асоціація:

- асоціація (біологія) – основна систематична одиниця класифікації рослинних угруповань або угруповань мікроорганізмів;
- асоціація (суспільство) – союз, спілка, товариство, об’єднання для досягнення спільної мети;
- господарська асоціація – організаційно-правова форма об’єднань підприємств;
- асоціація (речовини) – сполучення, з’єднання чого-небудь в одне ціле;
- асоціація (психологія) – поняття, що спадає на думку при згадуванні іншого поняття;
- асоціативний експеримент – прийом, спрямований на вияв асоціацій, які склалися в індивіда у його попередньому досвіді [5].

Всі ці ознаки тісно зв’язані між собою, головне поняття терміну «асоціація» – «поняття, згадування іншого поняття», «прийом, на вияв асоціацій, які склалися у попередньому досвіді». Саме ці речення можуть чітко охарактеризувати, що ж значення «асоціація».

Щоб чітко зрозуміти яким чином відбувається дія асоціації – потрібно розглянути питання «образ».

Перше питання яке постає це теорія розпізнавання образів – розділ кібернетики, що розвиває теоретичні основи й методи класифікації, ініціалізації та ідентифікації предметів, явищ, процесів, сигналів, ситуацій і т. п. об'єктів, які характеризуються скінченним набором деяких властивостей і ознак. Задачі даних типів вирішуються досить часто, наприклад, при переході або проїзді вулиці за сигналами світлофора. Розпізнавання кольору лампи світлофора, що засвітилася, і знання правил дорожнього руху дозволяє прийняти правильне рішення про те, можна, чи не можна переходити вулицю в цей момент. У більшості випадків, сприймаючи явища навколишнього світу, людина здійснює їх класифікацію, тобто розбиває ці явища (предмети, ситуації) на групи схожих явищ (саме схожих, а не тотожних). По тим чи іншим причинам необхідно віднести в одну групу у чомусь "подібні" явища чи предмети, які можуть при цьому значно відрізнятись один від одного. Наприклад, усі фігури, зображені на наступному малюнку, ми називаємо "літерою А", незважаючи на їх значну відмінність у їх написанні (рис. 1.1) [6].



Рисунок 1.1 – Приклад подібності написання літери «А»

Суттєвим є той факт, що виділивши такі групи (множини) об'єктів, ми отримуємо здатність "упізнавати", тобто встановлювати належність до вже відомої множини, нові об'єкти, які ще не зустрічалися нам раніше, наприклад впізнавати букви, написані новим для нас почерком [7].

Отримавши уявлення про те, що являє собою буква "А" на основі деякої, зазвичай невеликої кількості екземплярів цієї букви, ми спроможні упізнати як завгодно велику кількість інших її екземплярів.

Проте, далеко не всі множини об'єктів дають змогу на основі невеликої частини множини упізнати як завгодно багато інших невідомих нам її представників. Наприклад, фотографії студентів деякого вузу утворюють множину. Проте неможливо після ознайомлення зі скажімо десятьма фотографіями студентів визначити по новій фотографії є людина студентом цього вузу чи ні.

Проте, самі образи у той самий час є у певній мірі "розпливчастими" і питання про належність об'єкта до даного образу не завжди може бути вирішене однозначно.

На рисунку 1.2 показано, як за допомогою невеликих послідовних змін можна перетворити цифру 7 у цифру 1. Рисунок 1.2 ілюструє важливу особливість об'єктів кожного образу. Можна в деяких межах змінювати об'єкт, але він усе одно залишиться об'єктом того самого образу [8].

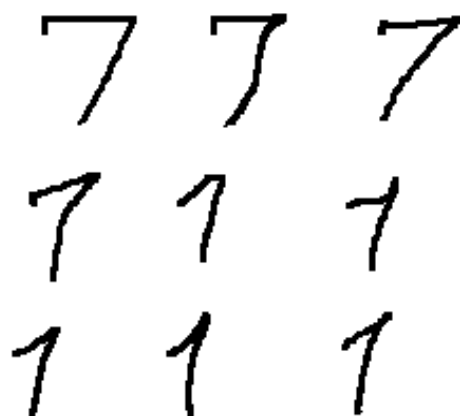


Рисунок 1.2 – Невелика зміна образу провадить до іншого результату

Створення штучних систем розпізнавання образів залишається складною теоретичною й технічною проблемою. Необхідність у такому розпізнаванні

виникає в самих різних областях – від військової справи й систем безпеки до оцифрування різних аналогових сигналів.

Що саме собою являє образ? Завдяки ньому можемо зробити аналіз існуючих вихідних даних та отримати той чи інший результат.

Образ – це процес, об’єкт або явище реального та абстрактного світу, який розпізнається за даними (ознаками), що збираються та оброблюються індивідуально і у сукупності. Розпізнавання образу (РО) не є безцільною грою і завжди супроводжується дією. В штучних інтелектуальних системах дія має вигляд порівняння з еталоном, увімкнення в роботу виконавчого пристрою, вилучення з розгляду образу, запису або видачі інформації тощо [9].

Образи розпізнають (класифікують, виділяють, описують) по ознаках, станах, зв’язках; по сукупності власних наявних субобразів. Образи у вигляді вектора $X = (x_1, x_2, x_j, x_n)$; вони частіше є сукупністю чітких та нечітких, повних та неповних, прямих та непрямих, постійних та змінних даних із загальною масою неістотних, взаємно суперечливих та зв’язаних між собою деталей [10].

1.2 Класифікація задач розпізнавання образів

В сьогоdnішній час інформаційні системи РО є складовою часткою будь-яких штучних інтелектуальних систем і використовуються в експертних системах, базах даних та базах знань, в інформаційних, прогнозних та інших системах. Прикладами можуть бути комплексні інтелектуальні системи наприклад, «Здобуття знань» («DataMining») для прийняття рішень, які використовують ряд методів РО, на рисунку 1.3 наведено систему пошуку та спосіб здобуття знань [11].

Будь-які існуючі або майбутні автоматичні чи автоматизовані інтелектуальні системи включно з РО повинні мати однакові риси («мету»,

«об'єкт зовнішнього середовища», «розпізнавання об'єкта по ознаках») та повинні виконувати відповідні «оптимальні інтелектуальні дії» [12].



Рисунок 1.3 – Система «Здобуття знань» [12]

Але різниця між «Розпізнаванням образів» та згаданими вище напрямками все ж є: дисципліна «Розпізнавання образів» узагальнює, об'єднує, досліджує і удосконалює основні оптимальні методи, алгоритми, моделі і засоби інтелектуальних систем при обробці даних реального чи абстрактного середовища, в той час як інші згадані інтелектуальні системи повинні використовувати і поглиблювати ці загальні методи дисципліни РО з урахуванням особливостей процесів у конкретних напрямках і ситуаціях.

1.3 Систематизація розпізнавання образів

Система рівнів розпізнавання образу ґрунтується на характеристичних ознаках, які використовуються у процесі ініціалізації. Можлива класифікація систем розпізнавання показана на рисунку 1.4.

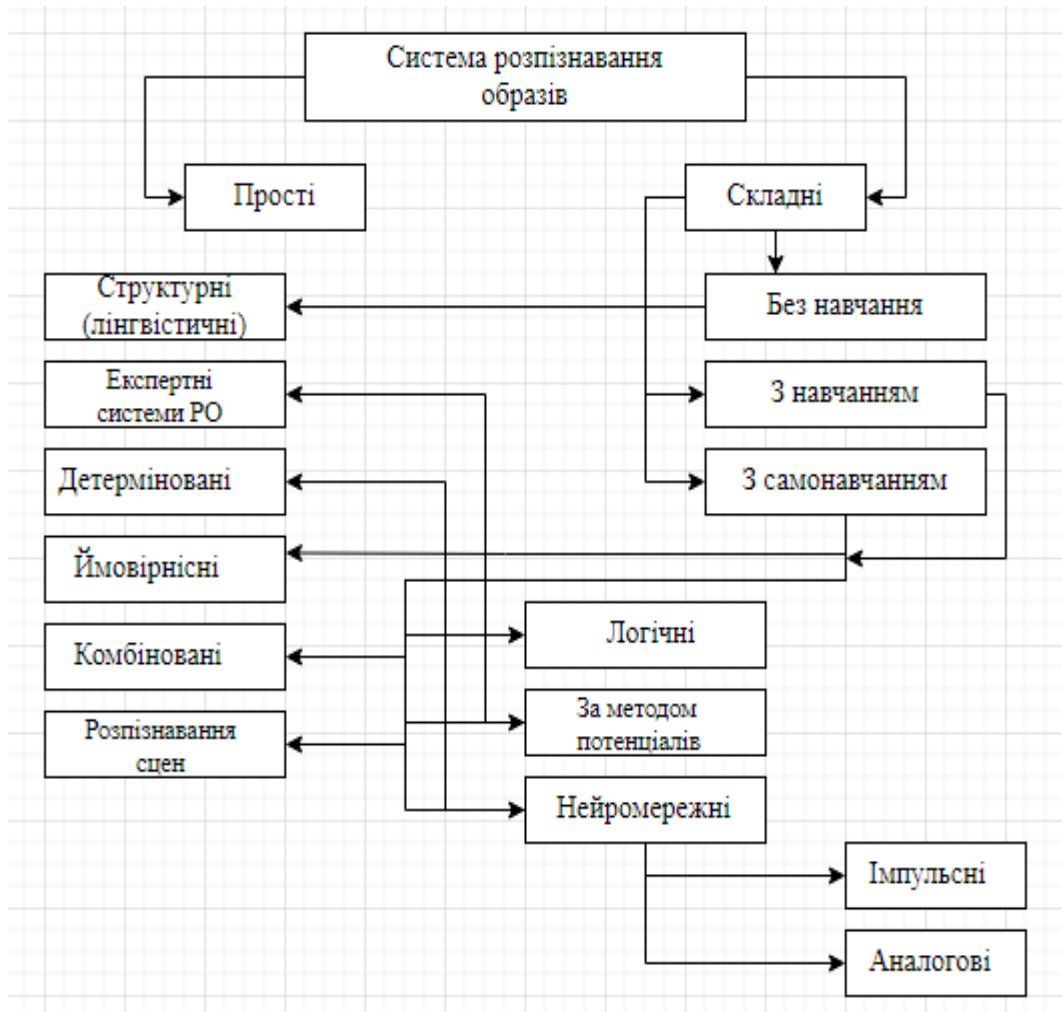


Рисунок 1.4 – Розподіл системи розпізнавання

Самонавчальна система використовує набір правил, за якими сама виробляє класифікацію, яка може відрізнитися від природної [13].

Приклад застосування експертних систем – є там, де існують евристичні або інтуїтивні методи рішень і немає точних алгоритмів або розрахунків. Проста експертна система містить накопичені знання багатьох експертів у певній

предметній області і здатна в рамках цієї області класифікувати об'єкти, давати відповіді, рекомендації, поради, запитуючи при необхідності додаткову інформацію. Експертна система найчастіше використовує правила, які мають форму: «if, then, else» [14]. До експертних систем можна віднести медичні системи (по ознаках розпізнають хвороби), ситуаційні системи (розпізнають ситуацію і приймають рішення стосовно наступних дій).

Детерміновані системи. У цих системах для побудови алгоритмів розпізнавання використовують методи, засновані на вимірюванні відстаней між розпізнаваним об'єктом і еталонами класів.

Ймовірнісні системи використовують ймовірнісний метод розпізнавання, заснований на теорії статистичних рішень.

Логічні системи використовують логічні методи розпізнавання, засновані на наявності логічних зв'язків, в якій змінні – логічні ознаки розпізнаваних об'єктів, а невідомі величини – класи, до яких ці об'єкти відносяться.

Системи РО з використанням метода потенціалів, за яким ознака об'єкту розглядається як його електричний потенціал, який зменшується із зростанням відстані до об'єкту [15].

Нейромережні системи використовують моделі нейронів для розпізнавання образів і розділяються на імпульсні та аналогові системи: імпульсні складаються з нейронів, які отримують на вході кортежі імпульсів (спайків) та генерують на виходах імпульси; аналогові складаються з нейронів, які аналізують безперервний сигнал.

Комбіновані системи РО використовують комбінації методів та систем. Приклади комбінованих систем – інтелектуальний аналізатор даних DataMining.

1.4 Принцип дії комунікативних та асоціативних процесів

Наслідувальне мислення являє собою комунікативні та асоціативні процеси зі словами, словосполученнями, реченнями і судженнями. На

комп'ютерному рівні наслідувальне мислення імітується комунікативними і асоціативними процесами з символічними мовними елементами знань. Імітація наслідувального мислення на комп'ютерному рівні здійснюється на основі символічної мовної комунікативної логіки і мережевого комунікативного та асоціативного уявлення символічних мовних елементів знань. Дію асоціативного процесу представлена на рисунку 1.5.

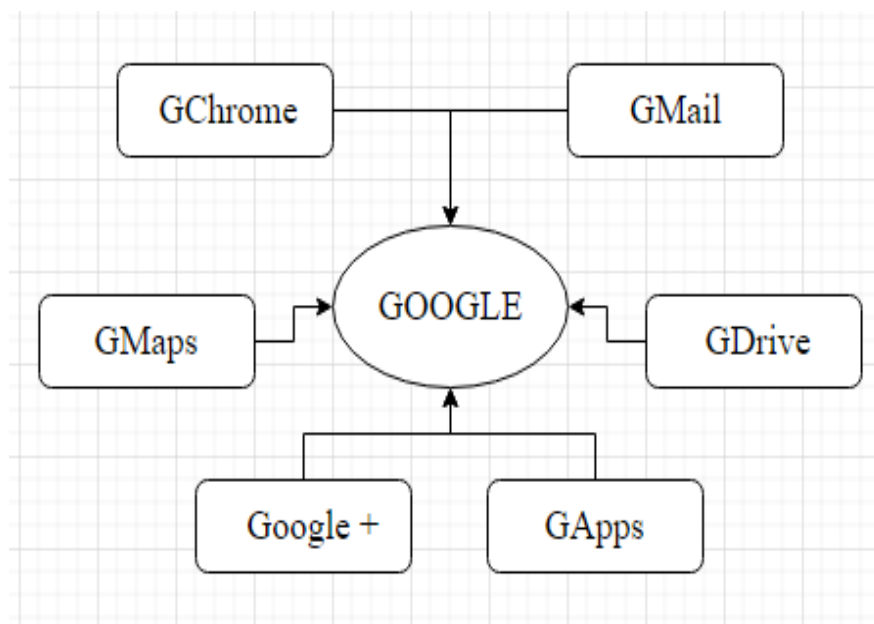


Рисунок 1.5 – Приклад асоціативного процесу [16]

Інформаційне середовище людини являє собою пов'язану середу першого сигнального рівня і мовну середу другого сигнального рівня. На першому сигнальному рівні зберігається інформація про навколишню реальність у вигляді ознак. На другому сигнальному рівні зберігаються символічні елементи знань: слів, словосполучень, пропозицій, суджень.

Тому імітувати символічно-мовне мислення людини можна на рівні розумових актів, зберігаючи відповідність між змістом і мовними символічними елементами знань, підміняючи навколишню або віртуальну (абстрактну) реальність мовним символічним описом з функціональною синтактикою, виділяє

акти розумового процесу, що зберігає відносини між елементами знань і представниками реалії та виробляє досвід розуміння і передачі сенсу [17].

Людина пов'язує смислові акти через просторово-часові, причинно-наслідкові зв'язки. Символічні структури елементів знань шикуються в пам'яті людини згідно комунікативним зв'язкам першого сигнального рівня між відповідними ознаками представників реалії або абстракції. Символічні елементи знань можна взаємно однозначно відобразити в кодовані патерну електричної активності пам'яті робота [18].

У такого робота смислові акти людини підміняються семантичним синтезом відповідних кодованих патернів, зберігаючи взаємно однозначна відповідність структур і комунікативних мовних зв'язків з відповідними елементами знань, можна здійснювати імітацію символічного мовного мислення за допомогою робота. В основі імітації символічного мовного мислення лежить символічно-мовна комунікативна логіка людини.

Імітація символічного мовного мислення здійснюється по комунікативним і асоціативним зв'язкам елементів знань. Тому важливо мати комунікативні зв'язки на стадії подання знань. Для цього потрібно вказувати предметну область знань, ситуацію для суджень, ситуативний момент для пропозицій. Ситуації пов'язують ситуативні моменти в смисловому орієнтованому поданні знань предметної області.

1.5 Аналіз функціональної структури робота

Робот з символічним мовним мисленням має систему розпізнавання співрозмовника, систему мовного введення інформаційних потреб, систему реалізації інформаційних потреб (систему імітації наслідувального мислення), нейромережевому систему синтезу мовлення за текстом реалізації інформаційної потреби.

Нейромережева система – це система широко взаємодіючих елементарних перцептронів. У будь-який момент часу кожен перцептрон має певний рівень активності, який визначається рівнями активності інших перцептронів, які передають сигнали даному перцептроні [19].

Навчання нейромережевої системи задається правилами. Правила навчання задають початкове безліч асоціативних ваг і визначають, в якому порядку повинні бути модифіковані ваги, щоб поліпшити взаємодію перцептронів (рис. 1.6).

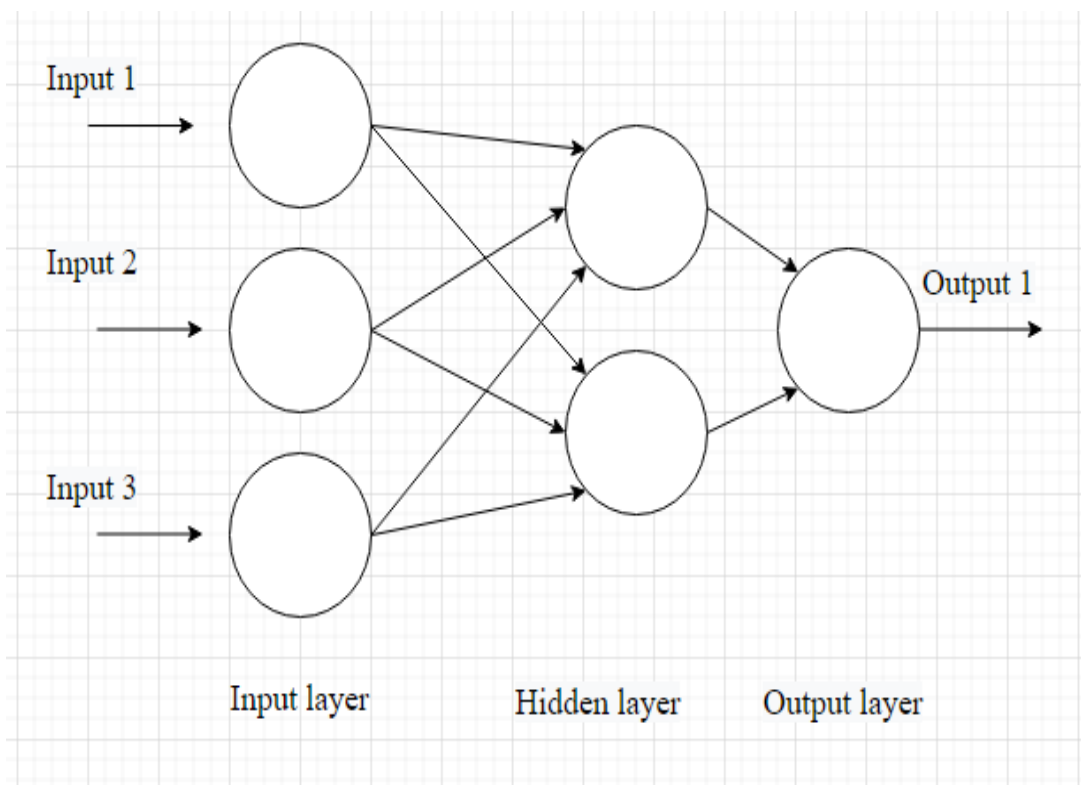


Рисунок 1.6 – Проста нейромережева система

Для розпізнавання осіб за допомогою багаторівневою нейромережевою системою задається критерій навчання наступною нормою (1.1):

$$E(w) = \sum_{i=1}^n (F(x_1) - F_w(x_i))^2, \quad (1.1)$$

де X_1, X_n – входи навчальної множини перцептронів;

$F(X_1), F(X_n)$ – відомі бажані виходи;

$F_w(X_1), F_w(X_n)$ – виходи, отримані рівневими мережами F_w , із заданими параметрами ваги і порогів.

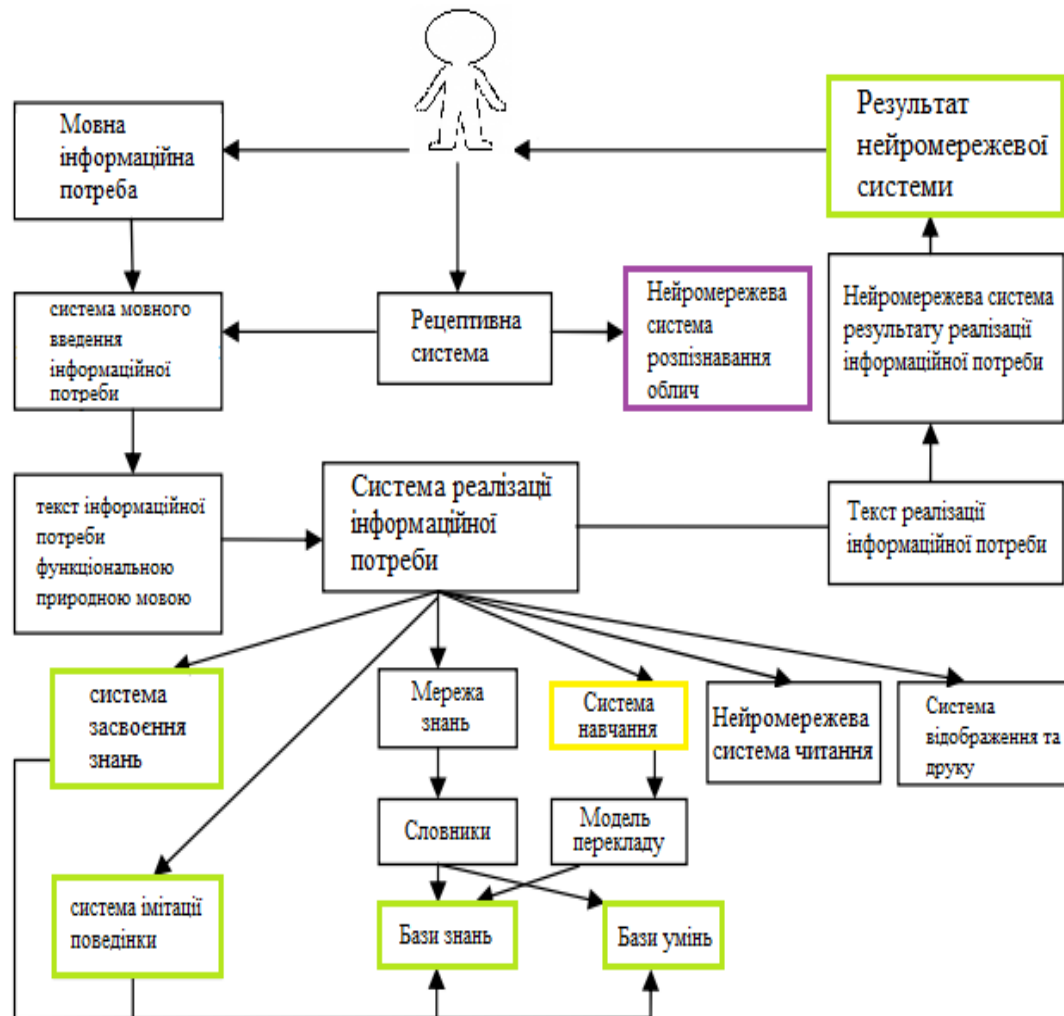


Рисунок 1.7 – Блочна схема системи роботи

Багаторівнева нейромережева система розпізнавання осіб вирішує наступні завдання: розпізнавання різних виразів обличчя, різних масштабів, присутності деталей (вуха, очі, тощо), різних перспектив та орієнтацій, шумових образів.

Інформаційною одиницею спілкування між роботом та співрозмовником є інформаційна потреба. Співрозмовник використовує інформаційні потреби, які у основі знань робота. Він спілкується з роботом за допомогою комбінацій інформаційних потреб, збагачуючи ними робота. Робот отримує нові базові інформаційні потреби, елементи знань та реалізації часу свого навчання (рис. 1.7).

Знайомство робота з людиною здійснюється через нейромережну систему розпізнавання осіб. Якщо людина невідома роботу, то рецептивна система запам'ятовує його мовний словник та обличчя. Якщо він відомий роботу, то ця система налаштує систему мовного введення інформаційної потреби на мовний словник співрозмовника. Після цього починається інформаційний контакт між роботом та людиною. Система мовного введення перетворює мовну інформаційну потреба в текст функціональною природною мовою.

Система мовного введення включає систему розпізнавання мови. Система розпізнавання мовлення перетворює мовну інформаційну потреба учня в коди і записує на згадку. Слова інформаційної потреби людина вимовляє за складами. Вимова за складами слова дозволяє здійснити послогове розпізнавання слів. Послогове розпізнавання слів не вимагає обчислювальної потужності та словникового запасу. Процес розпізнавання мовної інформаційної потреби є послідовністю процедур класифікації елементарних слухових образів. Слуховими образами є склади мовлення. У кожному нормально вимовляється склад стандарт, що забезпечує розуміння мови, що вимовляється різними людьми. Стандарт виділяється шляхом речеслухового нормування мовних складів. Як стандарт вибираються ознаки складів і відносини основних сигналів формантних структур букв складів шляхом факторного аналізу незалежних сигналів мовного стилю [19].

Система реалізації інформаційної потреби налаштовується на засвоєння знань чи реалізацію, чи на навчання співрозмовника, чи переклад, чи читання тексту, чи печатку, чи графічне відображення. Система реалізації інформаційної

потреби використовує словники сутностей предметної галузі бази знань, типові процедури реалізації з бази умінь, і навіть мережу поелементної реалізації інформаційних потреб.

1.6 Нейромережеві моделі асоціативного процесора та пам'яті

Нейромережева модель асоціативного процесора. У режимі запису така модель запам'ятовує таблицю відповідності пар (1.2):

$$(x_i, y_j), i = 1 - L, \quad (1.2)$$

де L – довжина таблиці;

x_i належить X_n ;

y_i належить Y_n ;

X_n та Y_n – простори векторів розмірності.

У режимі відтворення вона повинна у відповідь на пред'явлення на вхід сигналу x_i із записаної таблиці видати парний щодо нього сигнал y_i , а у відповідь пред'явлення сигналу x , що у таблиці, видати деякий стандартний сигнал y_0 . Сигнали x_i y_i , що входять до таблиць, – свої, а не вхідні – чужі. Чужим точкам ставиться у відповідність точка y_0 . Крапка y_i ставиться у відповідність деякому околиці точки x_i . Порушення кожного вихідного еталонного вектора y_i є реакцією моделі на специфічний вектор вхідний z_i – стимул. При кодуванні на вхід подається L пари векторів (x_i, z_i) . В результаті модифікації пластичних елементів формується зв'язок між еталоном x_i та парами (z_i, y_i) .

У режимі запису така модель запам'ятовує набір H еталонних векторів x_i , що належать X_n , $i = 1 - L$. У режимі відтворення при подачі на вхід x , що

належить X_n , вона повинна відтворити вектор X_j , що належить X_n . Якщо вектор x належить деякому околиці q_i вектор X_i , то вектор X_j належить околиці q_i вектор X_i .

Асоціативна пам'ять відтворює послідовність символів за її фрагментом. У режимі запису кожному її фрагменту довжини K ставиться у відповідність наступний та попередній символи послідовності. Під час подачі на вхід фрагмента на виході відтворюються попередній і наступний символи. Ці символи додаються до попереднього вхідного фрагменту та цикл роботи повторюється до того часу, доки відновиться послідовність (рис. 1.8) [20].

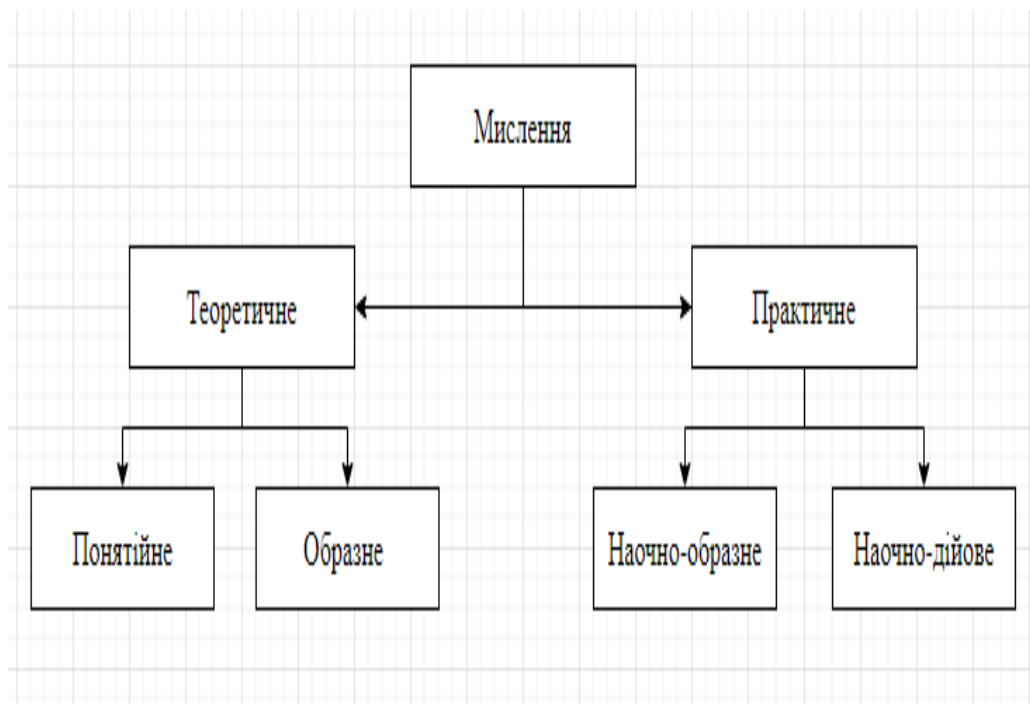


Рисунок 1.8 – Блочна схема роботи асоціативної пам'яті на простому рівні

За процедурою навчання на вхід нейромережевої моделі асоціативної пам'яті послідовно подаються L еталонних незалежних векторів x_i . Внаслідок пред'явлення кожного еталонного вектора і утворюється пам'ятний слід.

1.7 Інтелектуальні системи та різні методи представлення знань

З аналізу можемо зробити важливий висновок, що розвиток інтелектуальних систем тісно пов'язаний з роботами та дослідженнями у напрямку штучного інтелекту, спробами різними шляхами реалізувати аналог мислення людини.

Зараз при будові інтелектуальних систем (ІС) і виборі методу представлення знань використовуються різні підходи або їх комбінації, а саме:

- структурний;
- логічний;
- еволюційний;
- імітаційний.

Структурний підхід. Назва даного способу пов'язана із спробами побудови ІС шляхом моделювання на комп'ютері структури людського мозку, що включає моделі нейронів мозку і нейронних мереж. Однією з перших таких реалізацій структурного підходу став простий перцептрон Розенблатта, запропонований ним і досліджений у 1957-1959 рр. для розпізнавання зорових образів. Пізніше цей напрям став теорією розпізнавання образів і штучні нейронні мережі, і вважається одним з перспективнішим напрямів створення інтелектуальних систем.

Але потрібно пам'ятати, що в кожна теорія має свої проблеми та недоліки. Відповідно до сучасних уявлень мозок людини складається приблизно з 10 мільярдів нервових клітин – нейронів. Для моделювання одного нейрона потрібні досить складні електронні схеми, а для прояву більш-менш цікавих для практики інтелектуальних властивостей потрібно створювати ансамблі з вельми значної кількості нейронів. Вченими японських і німецьких наукових центрів була створена модельна мережа, що складається з 1,73 млрд. нервових клітин і 10,4 трлн синапсів (засобів з з'єднання клітин). Імітація однієї секунди активності навіть на такій моделі, приблизно еквівалентній 1 % реального мозку

людини, потребувала 40 хвилин машинного часу одного з самих швидкодіючих супер комп'ютерів.

Потрібно розуміти, що структурування моделі мозку до рівня нейронів, як виявляється, ще не гарантує успіху. Наприклад, Говард Гарднер – американський психолог, автор теорії множинного інтелекту, що стала класичною, вважає, що людина має не єдиний інтелект (головний «загальний інтелект»), а низку інших відносно незалежних здібностей. Моделювання цих інтелектуальних можливостей, що взаємодіють, на структурному рівні є задачею вельми нелегкою.

Логічний підхід. Цей підхід є найбільш поширеним. Його виникнення пов'язане безпосередньо із здібностями людини до розвиненого логічного мислення, що відрізняє його принципово від тварини. Основи логічного підходу почали розвиватися від логіки Аристотеля, числення висловлювань і булевої алгебри. Фактично усі побудовані на логічному принципі ІС реалізують так звані «машини доказу теорем». Кожна така машина має блок генерації мети і систему висновку, що намагається довести цю мету як теорему.

Еволюційний підхід використовується також досить широко при побудові ІС. У цьому випадку основна увага приділяється побудові початкової моделі і правилам, за якими вона може змінюватися. При цьому моделі може бути складеною з використанням найрізноманітніших методів, включаючи нейронні мережі, набори логічних правил і будь-які інші моделі. Окрім правил в програмі визначаються також критерії оцінки якості кожного варіанту. Зараз цей напрям отримав розвиток під назвою «генетичні алгоритми». Це алгоритм, що оперує з популяцією індивідів: $P(t) = \{x_i(t), x_n(t)\}$, де $t = 1, 2$ – номер ітерації. Розроблено чимало різних модифікацій генетичних алгоритмів, що утворюють цілий клас методів.

Комбінаційний підхід. На верхніх рівнях управління для прийняття рішень частіше використовуються логічні підходи. Завданням систем нижніх рівнів управління є забезпечення їх взаємодії із зовнішнім середовищем отримання і

первинна обробка інформації і формування реакцій системи залежно від рішень, прийнятих на верхніх рівнях. Виходячи з цього можемо сказати, що в системах нижнього рівня, де відбувається обробка первинної інформації, частіше використовуються комбінація структурного, еволюційного і імітаційного підходів.

1.8 Приклад дії асоціацій з урахуванням кластеризації

Кластерний аналіз полягає у поділі заданого набору об'єктів на кластери – групи об'єктів, схожі між собою за тим чи іншим критерієм подібності. Близькими по змісту до терміну "кластер" є поняття "клас" та "таксон". Тому кластеризацію часто називають таксономією або класифікацією без учителя, оскільки, на відміну від задачі "звичайної" класифікації, класи апіорно не задані.

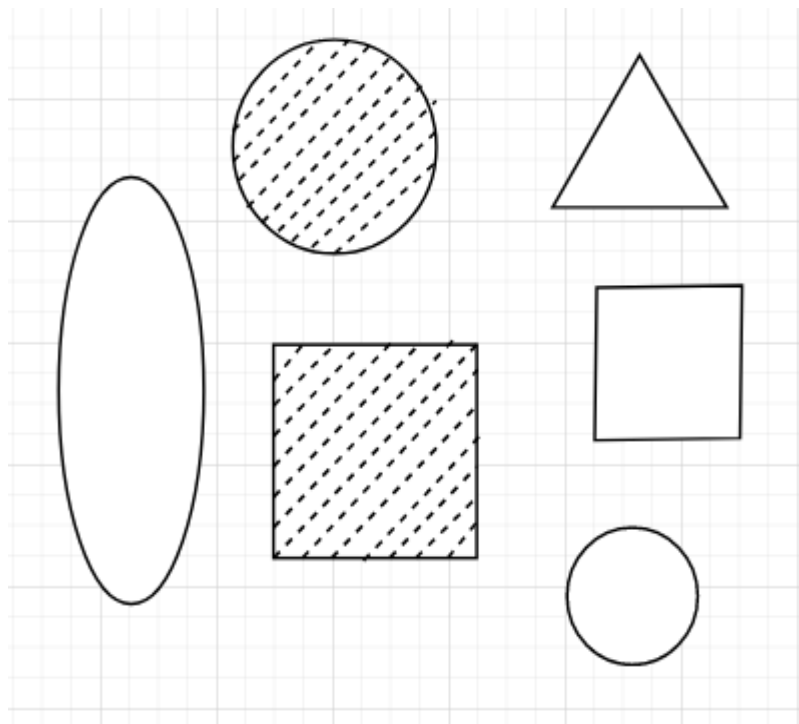


Рисунок 1.9 – Фігури з різним критерієм кластеризації

Слід зазначити, що у кластеризації присутні як об'єктивні, так і суб'єктивні моменти. Це добре ілюструється прикладом Богнарда. Фігури, зображені на рисунку 1.9 можна розбити на різні класи у залежності від критерію кластеризації: колір, кількість кутів та розмір. Тому результат кластеризації визначається багато у чому завданнями та засобами, які використовують у її процесі міри близькості. Для виділення кластерів потрібно у першу чергу ввести функцію подібності об'єктів. Розглянемо міри близькості у випадку кластеризації об'єктів, заданих у векторній формі. Найбільш часто використовується евклідова відстань. Для булевих векторів використовують відстань Хеммінга. Також часто використовують відстань Махаланобіса (1.3):

$$d(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)}, \quad (1.3)$$

де S – коваріаційна матриця;

x, y – об'єкти-змінні.

Можна також використовувати неметричні функції близькості (1.4):

$$s(x, z) = \frac{x'z}{\|x\| \|z\|}, \quad (1.4)$$

Яка рівна косинусу кута між векторами x та z (у останній формулі через x' позначається вектор-рядок транспонований, до вектора-стовпчика x). У випадку двійкових векторів величина $x'z$ рівна кількості співпадаючих ознак для об'єктів x та z . Іншим варіантом формули (1.4) є міра Танімото (1.5):

$$s(x, z) = \frac{x'z}{\|x\|^2 + \|z\|^2 - x'z}. \quad (1.5)$$

Слід зазначити, що у зв'язку із неоднорідністю шкал, у яких вимірюються ознаки об'єктів, при розрахунках відстаней може виявитися, що деякі ознаки

набагато сильніше впливають на близькість об'єктів, ніж інші. Тому бажано попередньо провести нормалізацію змінних [21].

Основні підходи до побудови методів кластеризації. Методи кластеризації поділяють на ієрархічні та неієрархічні. У ієрархічних методах будується дерево вкладених кластерів (дендограма). Приклад зображення дендограм наведено на рисунку 1.10.

Ієрархічні методи поділяються на агломеративні методи та методи поділу. Агломеративні методи полягають у послідовному об'єднанні початкових кластерів (побудова кластерів знизу вгору). Методи поділу – у розщепленні існуючих кластерів (побудова кластерів зверху вниз).

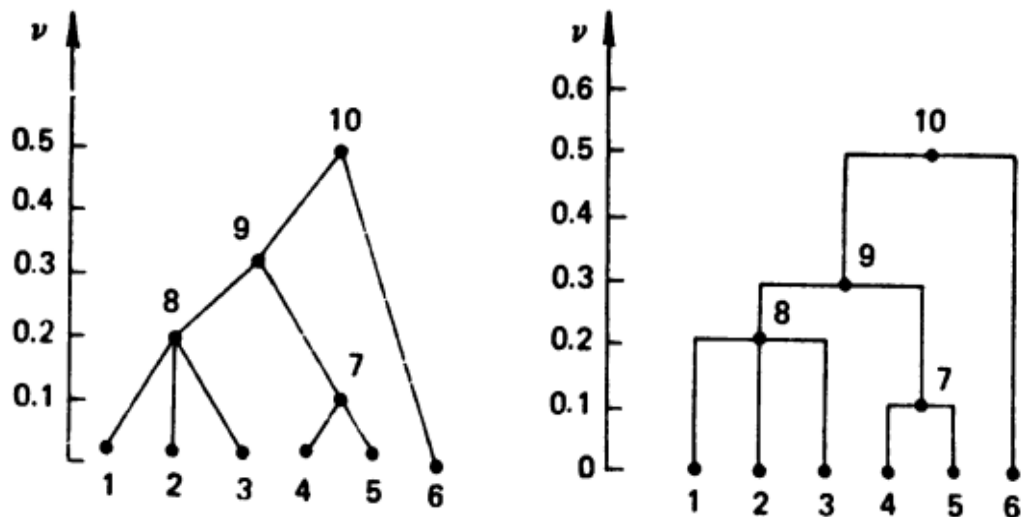


Рисунок 1.10 – Різні способи зображення дендограм [22]

1.9 Порівняння різних алгоритмів кластеризації

Агломеративні алгоритми кластеризації. Нехай x – скінченна множина, яку потрібно кластеризувати. На першому кроці кожному елементу множини відповідає одноелементний кластер. На кожному наступному кроці вибираються два найближчі між собою кластери C_i та C_j , які об'єднуються у новий кластер

$C_j \cup C_i$. Після кожного такого кроку кількість кластерів зменшується. У кінці процедури отримується кластер, який складається із одного об'єкта.

Для оцінки відстані $d(C_j, C_i)$ між двома множинами використовують наступні функції:

а) відстань між найближчими об'єктами множин (1.6)

$$d(C_i C_j) = \min\{d(x, y) | x \in C_i, y \in C_j\}; \quad (1.6)$$

б) відстань між найбільш віддаленими об'єктами (1.7)

$$d(C_i C_j) = \max\{d(x, y) | x \in C_i, y \in C_j\}; \quad (1.7)$$

в) середня відстань між множинами (1.8)

$$d(C_i C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y); \quad (1.8)$$

г) відстань між центрами мас множин (1.9)

$$d(C_i, C_j) = d(z_i, z_j); \quad (1.9)$$

д) статистична відстань (1.10)

$$d(C_i C_j) = \sqrt{\frac{|C_i| \cdot |C_j|}{|C_i| + |C_j|}} \|z_i - z_j\|. \quad (1.10)$$

де z_i, z_j – центри мас відповідних множин, які обчислюються

за формулою: $z(C) = \frac{1}{|C|} \sum_{x \in C} x$.

Відстань $d(x, C)$ між об'єктом x та множиною C покладають рівною $d(\{x\}, C)$.

Алгоритм Боннера. Випадковим чином вибирається одноелементний кластер, який об'єднується з тими одноелементними кластерами, відстань до яких не перевищує заданий поріг T . Далі випадково вибирається наступний одноелементний кластер і так само проходить злиття. Процес продовжується до тих пір, поки ще є одноелементні кластери [21].

Алгоритми поділу. Алгоритм Макнаутона-Сміта. У алгоритмі багато разів застосовується наступна процедура поділу множини C на два кластери C_1 та C_2 .

На початку усі об'єкти множини заносяться у кластер C_1 , а кластер C_2 – порожній.

На кожному кроці знаходиться об'єкт $x \in C_1$, для якого величина $d_x = d(x, C_1 \setminus \{x\}) - d(C_2, x)$ приймає найбільше значення, де $d(x, C_1 \setminus \{x\})$, $d(x, C_2)$ – відстані між об'єктом x та відповідними множинами.

Якщо $d_x > 0$, то об'єкт x виключається із кластера C_1 і заноситься у кластер C_2 і відбувається перехід на наступний крок процедури поділу, інакше робота процедури завершується. У алгоритмі Макнаутона-Сміта спочатку усі об'єкти заносяться у один кластер C . Потім цей кластер за допомогою вищенаведеної процедури розбивається на два кластери C_1 та C_2 . Далі процес розбиття застосовується до одного з кластерів C_1 та C_2 . У результаті отримують три кластери і т.д. Рекомендується на кожному кроці вибирати для розбиття кластер найбільшого діаметру.

Алгоритм ISODATA (алгоритм Болла-Холла). Алгоритм ІЗОДАТА є модифікацією алгоритму групових середніх. Спочатку випадковим чином із множини x , яка підлягає кластеризації, вибираються центри k кластерів. Інші об'єкти множини x відносяться до кластера із найближчим центром. Далі знаходяться центри кластерів так само, як у методі групових середніх. Два кластери C_i та C_j зливаються, якщо статистична відстань $d(C_i, C_j)$ менша за

задане порогове значення T_1 . Якщо середньо групова дисперсія деякого кластера C_1 по деякій координаті r ($1 \leq r \leq n$) більша за порогове значення T_2 , то відповідний кластер розбивається. Далі процедура застосовується до отриманих кластерів.

1.10 Постановка завдання дослідження

Головна метою кваліфікаційної роботи є розробка методу ідентифікації об'єктів для мобільних роботів на базі асоціативної логіки та використання способів розпізнавання образів.

Асоціацій при запам'ятовуванні це те, що ми прив'язуємо нові знання до відомої нам інформації. Щоб побудувати хорошу асоціацію, потрібно знати деякі корисні критерії знаходження зв'язку між речами, а також розвивати своє асоціативне мислення та творчу уяву. Не менш важливо навчитися будувати асоціативні ряди та зв'язки для стимулювання образної пам'яті.

Мета кваліфікаційної роботи – імплементація програми ідентифікації об'єктів в промисловій зоні для мобільних роботів на базі нейронних мереж з правилами асоціативної логіки [23].

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- провести аналіз особливостей асоціативних процесів та способів розпізнавання образів;
- провести аналіз дії асоціації з урахуванням кластеризації;
- порівняти алгоритми кластеризації;
- зробити розрахунки методів розпізнавання образів з використання правил асоціативних процесів;
- проаналізувати приклади розв'язання математичних нерівностей вагової функції;
- поєднати систему YOLO з бібліотекою OpenCV для подальшої роботи;

- імплементувати програму з урахуванням методу асоціативних процесів;
- розробити програму виявлення об'єктів в реальному часі – провести експериментальне дослідження на розробленій програмі.

1.11 Висновки до першого розділу

В ході аналізу було виявлено, що асоціативна логіка – важкий механізм, який використовує метод асоціацій або метод зіставлення.

Асоціація – це зв'язок між окремими фактами, подіями, предметами чи явищами, відбитими у свідомості людини і закріпленими у його пам'яті. Асоціативне сприйняття та мислення призводить до того, що поява одного елемента в певних умовах викликає образ іншого, пов'язаного з ним.

Але виходячи з проведеного аналізу було з'ясовано, що пам'ять робота здатна використовувати вивчену інформацію тільки на короткий час, з часом інформація забувається або ж просто не вистачає пам'яті щоб запам'ятовувати цю інформацію.

Отже, розробка та дослідження даної теми з правилами асоціації – є актуальною темою в сьогоденні час.

2 МАТЕМАТИЧНІ РОЗРАХУНКИ МЕТОДІВ РОЗПІЗНАВАННЯ ОБРАЗІВ, ОСНОВА РОБОТИ ШТУЧНОГО НЕЙРОНА

2.1 Розв'язання математичної нерівності вагової функції

Пам'ятаючи, що будь-яка математична нерівність виконує логічну функцію. Дану особливість можна використати для розділу на «худих» та «товстих» множини людей з характеристиками (x_1, x_2) , де $x_1 = 0 - 150$ кг – вага людини, а $x_2 = 0 - 2,0$ м – її зріст. На рисунок 2.1 можемо знайти координати «худої» та «товстої» людини вони помічені чорними точками (рис. 2.1) [24].

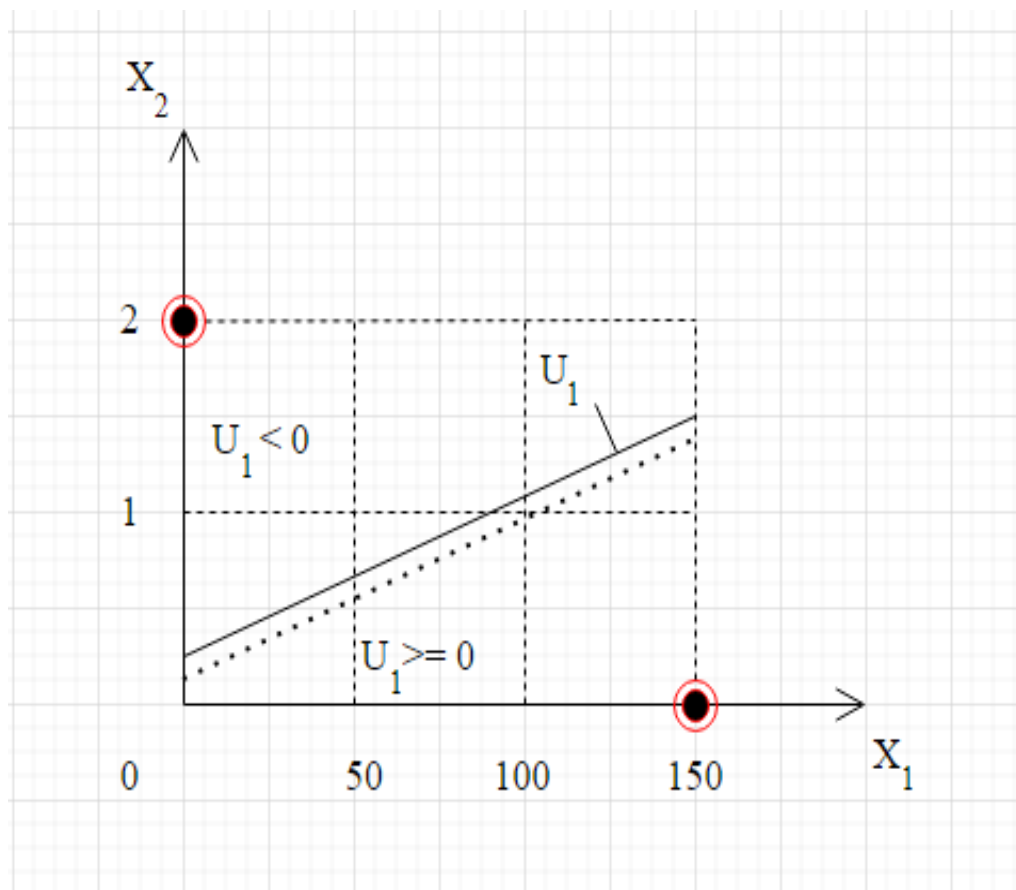


Рисунок 2.1 – Класифікація об'єктів на два класи по вазі x_1 та зросту x_2

Тоді всю множину об'єктів що розміщена у прямокутнику при $x_1 = 0 - 150$ кг, $x_2 = 0 - 2$ м можна розділити на дві частки («худих» та «товстих») за допомогою прямої лінії (вхідної вирішальної функції нейрону), яка знаходиться між вказаними чорними точками у вигляді нерівності (2.1):

$$u_1 = x_1 - 150x_2 + 45 \geq 0. \quad (2.1)$$

Нерівність (2.1) розділяє площу ознак (x_1, x_2) на дві півплощини: в одній півплощині нерівність виконується, а в іншій – ні. У цьому можна пересвідчитись, якщо підставити в нерівність (2.1) координати будь-якої довільно обраної точки. На практиці для перевірки нерівності найчастіше використовують точку $(x_1 = 0, x_2 = 0)$. В цій точці ми отримуємо відповідь, що нерівність виконується на цій півплощині. Ця півплощина на прямій u_1 помічається пунктирною лінією. На іншій півплощині нерівність не виконується для всіх її точок.

Таким чином, вхідна функція нейрону є вирішальною функцією і виконує логічну умову: якщо $u_1 \geq 0$ (нейрон видає сигнал «Істина»), то це – людина «повна», якщо ні, то «худа» (нейрон видає сигнал «Хибність»).

Приклад нейрону Adaline згідно вирішальної функції (2.1) можемо побачити на рисунку 2.2. Нелінійний вихід нейрону призначений для обмеження числових величин *NET* значеннями $y = -1 \dots +1$, які виконують логічний висновок нейрона і одночасно запобігають отриманню занадто великих вихідних цифр, які можуть призвести до аварійної зупинки електронно-обчислювальної машини (ЕОМ) по переповненню.

Таким чином, для об'єктів з кількісними ознаками за допомогою математичних нерівностей ми можемо створювати логічні вирази для їх класифікації.

Структура нейрону Adaline є на рисунку 2.2.

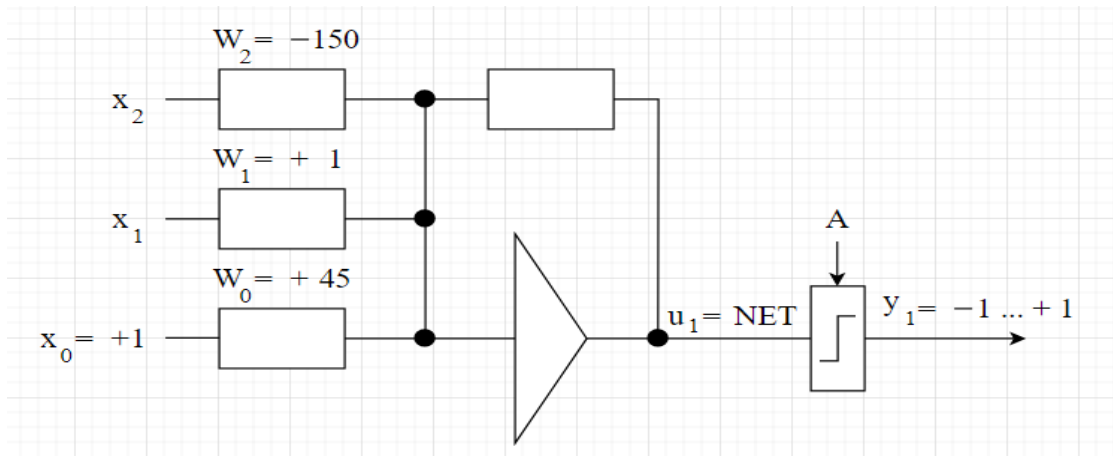


Рисунок 2.2 – Класифікація об'єктів на два класи по вазі x_1 та зросту x_2

Таким чином, для об'єктів з кількісними ознаками за допомогою математичних нерівностей ми можемо створювати логічні вирази для їх класифікації.

Даний тип завдання можемо розв'язати двома шляхами:

Перший варіант: за рахунок створення програми на будь-якій мові програмування (найчастіше так й роблять). Тоді можемо розмовляти про програмну реалізацію нейронної мережі.

Другий варіант: за рахунок створення спеціального пристрою, якій зветься штучним нейроном (наприклад, за рахунок використання операційного підсилювача та нелінійного перетворювача виходу – рисунок 2.2). Тоді можна казати про апаратну реалізацію нейронної мережі, маючи на увазі що мережа складається з окремих пристроїв – штучних нейронів.

Робота операційного підсилювача слугує тим, що перемножує вхідні постійні величини та змінні сигнали x_j на коефіцієнти передачі W_j (вагові коефіцієнти) і потім отримує їх підсумок.

Важливу роль у даному випадку виконує саме знак функції u , а не її величина. Тому додатні значення перетворюють на «+1» (маленький зріст), а від'ємні значення – на «-1» або «0» (високий зріст).

Практично ми перетворюємо математичну нерівність в логічну умову. Функція u іноді зветься NET (у матричній формі це має вигляд рівняння $U = NET = W \cdot X$), а вихідна нелінійна функція $y = f(NET)$ – зветься OUT . Замість вихідних сигналів « $y = +1 / -1$ » можуть застосовуватись сигнали « $y = +1 / 0$ ». Функція перетворення значення NET у вихідну функцію $y = f(OUT)$ – зветься активаційною (вихідною) функцією нейрона.

Можемо припустити, що відомі входні функції означають відомість вагових коефіцієнтів, тоді для класифікації об'єктів можна не застосовувати процес навчання.

Функція активації $y = F(NET) = F(u) = OUT$ може мати різний вигляд. Рисунок 2.3 демонструє яким чином приблизно відображається їх уживаність. Усі вони можуть бути зсунуті вздовж осі u .

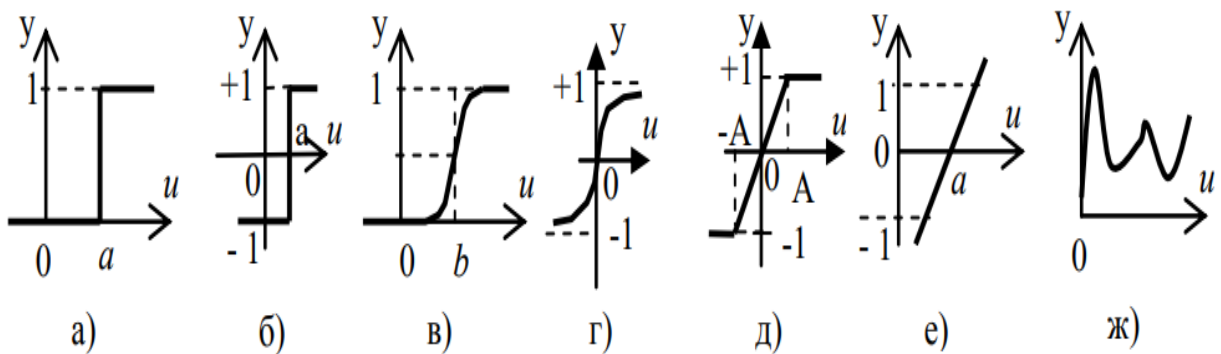


Рисунок 2.3 – Функції взбудження штучного нейрону [25]

В залежності від ситуації після активаційної функції на виході нейрона використовується додаткова вихідна функція $f_{вих}(y)$, яка призначена для перетворення та уточнення значення виходу нейрону.

Частіше всього можемо зустріти такі функції активації: уніполярна стрибкоподібна функція (рис. 2.3, а); біполярна стрибкоподібна функція (рис. 2.3, б); сігмоїда $y = 1 / [1 + e^{-a(u-b)}]$, де a, b – постійні коефіцієнти (рис. 2.3, в); тангенс гіперболічний (рис. 2.3, г); складена з відрізків прямих ліній (рис. 2.3, д); лінійна залежність (рис. 2.3, е); довільна задана функція

(рис. 2.3, ж). Виходячи з аналізу можемо сказати, що активаційна функція задається користувачем і може мати будь-який вигляд, наприклад, при бажанні вона може мати вигляд або прямої лінії, або складатись з 10-ти відрізків прямих ліній, або мати задану користувачем будь-яку нелінійну залежність. Вигляд активаційної функції обирається користувачем у залежності від вимог конкретної задачі.

Частіше всього для складної НМ, яка розв'язує конкретну задачу, всі нейрони мають однакову активаційну функцію.

Головне призначення активаційної функції – обмеження максимального сигналу виходу значенням модуля «1»: це означає, що відбувається обмеження розрахункових числових значень НМ і не дозволяє їм наблизитись до неприпустимо великих величин, які викликають зупинку роботи ЕОМ. Слід сказати, що такої переваги не має активаційна функція (рис. 2.3, е), яка є лінійною функцією $OUT = Ku$, де K – константа.

Електричні схеми штучного нейрону продемонстровані на рисунках 2.4 – 2.6.

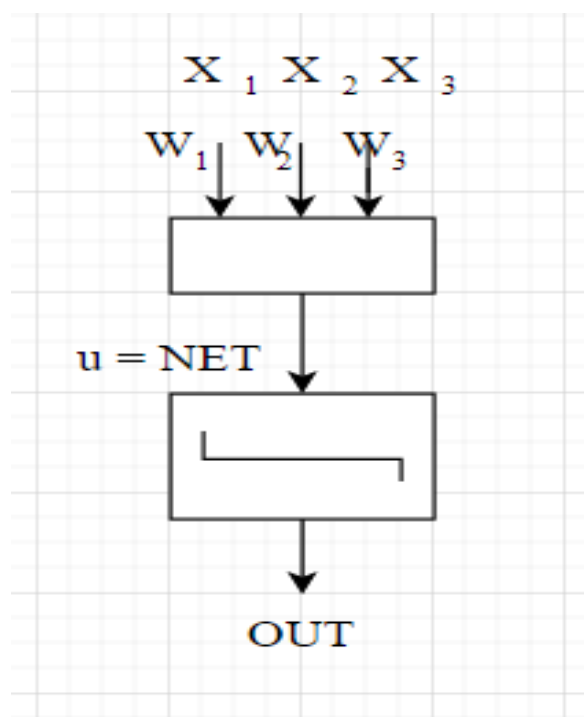


Рисунок 2.4 – Штучний нейрон У. Маккалоха і У. Піттса

У. Маккалох і У. Піттс першими опублікували дослідження про роботу штучних нейронів з пороговим виходом «1\0» у вигляді електричної схеми (рис. 2.4).

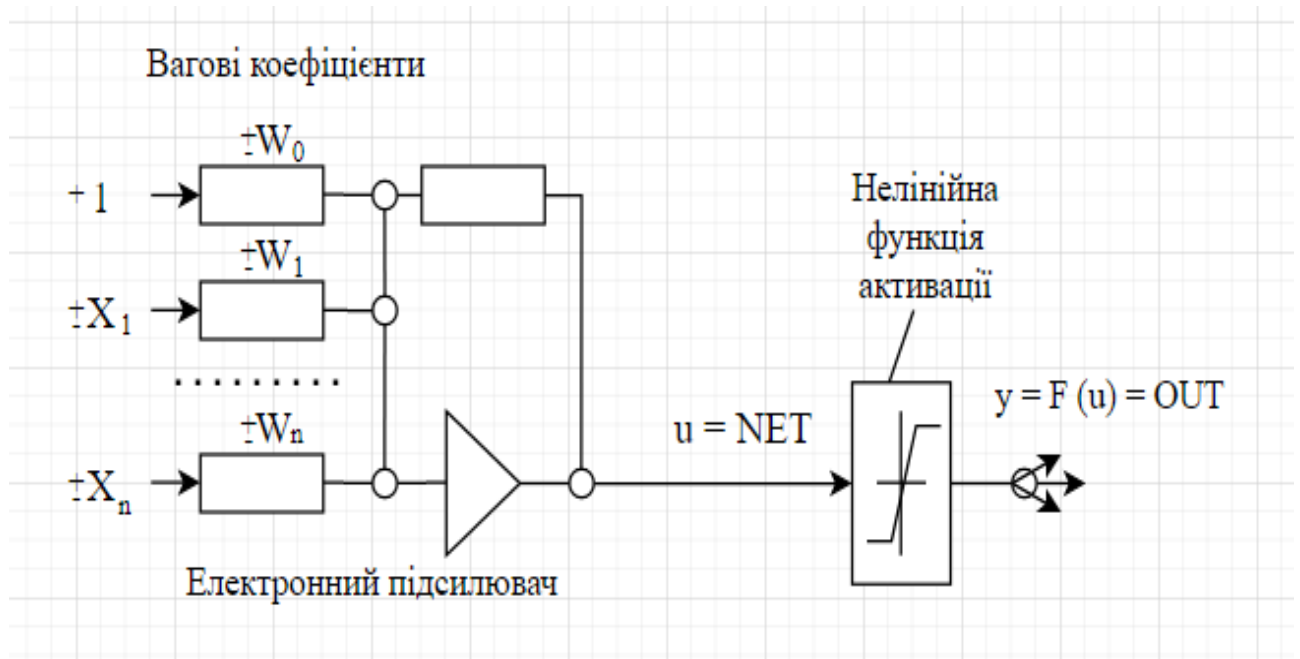


Рисунок 2.5 – Штучний нейрон ADALINE Б. Відрова та М. Хоффа

Б. Відров та М. Хофф розробили модель ADALINE (ADaptive Linear NEuron – Адаптивні Лінійні Нейрони) та мережі з ADALINE. Як можемо побачити з рисунку 2.5, на вхід нейрону була додана постійна складова. Зараз ADALINE є найбільш поширеним елементом нейронних мереж.

Особливості нейрона ADALINE:

– значення $u = NET$ проходить через нелінійну функцію активації у вигляді:

$$y = F(u) = F(NET) = OUT = \begin{cases} 1 & \text{при } NET \geq 0, \\ 0 & \text{при } NET < 0, \end{cases}$$

Сигнал NET може бути розрахованим у матричній формі (2.2):

$$NET[1,1] = X^T[1,n + 1] W [n + 1,1], \quad (2.2)$$

де $x(x_0 = 1 = const, x_1, x_n)$ – вектор вхідних сигналів (значення $x_0 = 1 = const$ вектора x задається довільно);

$W(w_0, w_1, w_n)$ – вектор вагових коефіцієнтів;

– вхідні сигнали x_j множаться на вагові коефіцієнти w_j , підсумовуються і підсилюється по потужності електронним підсилювачем, в результаті чого на його виході отримуємо значення вирішальної функції $u = NET$, яке, у залежності від вхідних координат x_j , може мати додатне або від’ємне значення

$$u = \sum_{j=0}^n w_j x_j = NET \geq 0. \quad (2.3)$$

Математично доведено, що будь-яка нерівність, в тому числі і у вигляді (2.3), виконує логічні функції: якщо отримана сума « $NET = u$ » більше заданого порогового значення, то вихід дорівнює 1, інакше – 0.

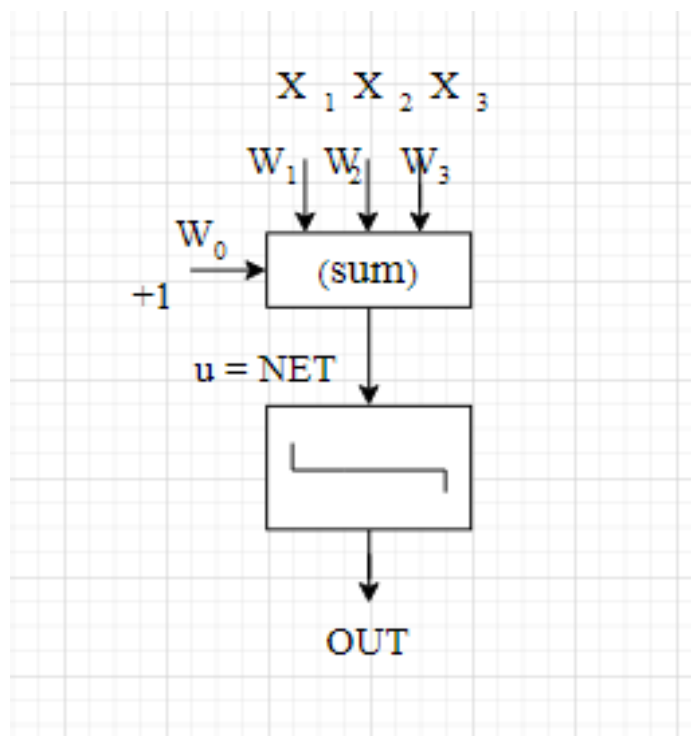


Рисунок 2.6 – Персептрон Розенблата

Розенблатт виконав моделювання перцептронів – багатозарової НМ з ваговими коефіцієнтами прямих зв'язків між шарами. Кожний шар має довільну кількість нейронів. У 1960 р. перцептрон був реалізований на ЕОМ «Марк-1» для розпізнавання літер алфавіту (рис. 2.6) [26].

Переваги моделі перцептрона:

- перцептрони складаються з одного або кількох шарів штучних нейронів з прямим розповсюдженням дискретних сигналів. В кожному шарі розміщується довільна кількість нейронів. Зі збільшенням кількості нейронів НМ може розв'язати більш складні задачі;
- входи та виходи нейронів мають значення або «1» або «0»;
- перцептрони не мають зворотних зв'язків;
- перцептрони можуть виконати класифікацію лише для лінійно розділювальної сукупності векторів (якщо пряма лінія або площина може розділити введені вектори на класи, то вважається, що вектори є лінійно розділювальними).

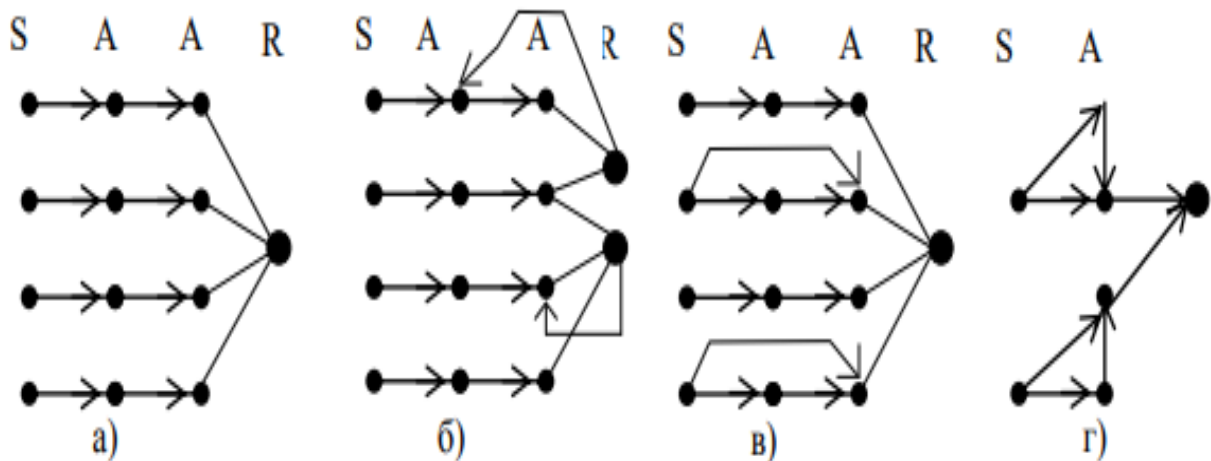


Рисунок 2.7 – Різні види зв'язків між шарами нейронів [25]:

а) прямі; б,в) зворотні; г) перехресні

На рисунку 2.7 продемонстровані прості системи зв'язків між шарами нейронів.

2.2 Висновки до другого розділу

Аналіз математичних розрахунків показав яким чином в електричній схемі штучного нейрона побудована система зв'язків між кожним елементом. Було розібрано яким чином працює штучний нейрон, було з'ясовано недоліки та переваги кожного з них. Також можемо зробити висновок, що вихідне значення цілком залежить від чіткості вхідних значень та їх кількості.

3 ІМПЛЕМЕНТАЦІЯ ПРОГРАМИ З УРАХУВАННЯМ МЕТОДУ АСОЦІАТИВНИХ ПРОЦЕСІВ

3.1 Впровадження в YOLO. Виявлення об'єктів за допомогою YOLO

Остатній розділ передбачає написання програми з урахуванням асоціативних процесів та розпізнавання образів в реальному часі. На цьому етапі буде представлено новий підхід до виявлення об'єктів, а саме за допомогою YOLO. Попередня робота з ініціалізацією та ідентифікацією обличчя перепрофілює класифікатори та використовує спосіб так званих каскадів для виконання виявлення. Замість цього ми розглядаємо виявлення об'єктів як проблему регресії на просторово розділені обмежувальні рамки і пов'язані класові ймовірності. Одна нейронна мережа прогнозує обмежувальні квадрати та ймовірності класів безпосередньо з повні зображення в одній оцінці. З моменту всього виявлення конвеєр – це єдина мережа, його можна оптимізувати наскрізь безпосередньо на продуктивність виявлення.

Наша уніфікована архітектура надзвичайно швидка. Модель YOLO обробляє зображення в режимі реального часу з частотою 45 кадрів за секунду. Менша версія мережі, Fast YOLO, обробляє приголомшливі 155 кадрів в секунду.

Люди дивляться на зображення і миттєво знають, які об'єкти на зображенні, де вони знаходяться і як вони взаємодіють. Зорова система людини швидка й точна, що дозволяє нам виконувати складні завдання, наприклад, водіння, не замислюючись. Швидкі та точні алгоритми виявлення об'єктів дозволяють комп'ютерам керувати автомобілями без спеціалізованих датчиків, а допоміжні пристрої передають інформацію в режимі реального часу [27].

Сучасні системи виявлення змінюють призначення класифікаторів для виконання виявлення. Щоб виявити об'єкт, ці системи беруть класифікатор для цього об'єкта та оцінюють його в різних місцях і масштабують на тестовому

зображенні. Такі системи, як деформовані деталі моделі (DPM) використовують підхід з розсувним вікном, де класифікатор виконується в рівномірно розташованих місцях по всьому зображенню (рис. 3.1).

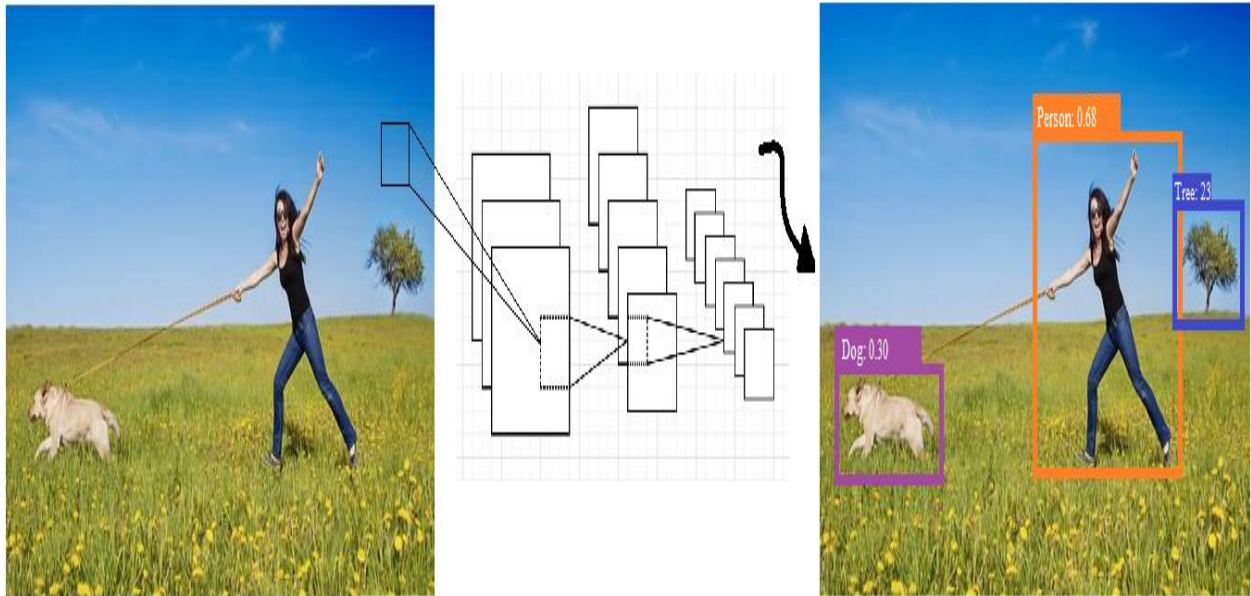


Рисунок 3.1 – Приклад системи виявлення YOLO

Наша система змінює розміри вхідне зображення до 448×448 , потім запускає одну згорткову мережу на зображенні і обмежує результати виявлення за допомогою впевненістю моделі.

YOLO надзвичайно простий: дивіться рисунок 3.1. Одна згортка мережі одночасно прогнозує кілька обмежувальних прямокутників і ймовірності класів для цих блоків. Тренування відбувається на повних зображеннях і безпосередньо оптимізується продуктивність виявлення. Ця уніфікована модель має ряд переваг над традиційними методами виявлення об'єктів.

По-перше, YOLO надзвичайно швидкий. Оскільки ми визначаємо кадр як проблема регресії, нам не потрібен складний конвеєр.

По-друге, YOLO міркує глобально про зображення, коли робити прогнози. На відміну від розсувного вікна та регіону методи, засновані на пропозиції,

YOLO бачить ціле зображення під час навчання та тестування, тому він неявно кодує контекстну інформацію про класи, а також їх зовнішній вигляд.

По-третє, YOLO вивчає узагальнювані уявлення об'єктів. Під час навчання на природних зображеннях та тестування на ілюстраціях YOLO перевершує найкращі методи виявлення, як-от DPM і R-CNN з великим відривом. Оскільки YOLO дуже узагальнений, він має меншу ймовірність руйнування при застосуванні нових доменів або неочікуваних вхідних даних.

3.2 Уніфікація окремих компонентів виявлення об'єктів та поєднання в єдину нейронну мережу

Наша мережа використовує функції з усього зображення, щоб передбачити кожен обмежувальну рамку. Це також прогнозує всі обмежувальні рамки для всіх класів для зображення одночасно. Це означає, що наша мережа глобально міркує про повне зображення та всі об'єкти на зображенні. Конструкція YOLO забезпечує наскрізне навчання та швидкість в реальному часі, зберігаючи високу середню точність [16].

Можемо сказати, що система розбиває вхідне зображення на сітку $S \times S$. Якщо центр об'єкта потрапляє в клітинку сітки, ця клітинка сітки відповідає за виявлення цього об'єкта. Даний спосіб продемонстровано на рисунку 3.2.

Кожна обмежувальна рамка складається з 5 передбачень: x , y , w , h , і так звана довіра. Координати (x, y) представляють центр квадрата відносно меж клітинки сітки. Ширина і висота прогнозуються відносно всього зображення. Прогноз довіри представляє собою IOU між прогнозованим боксом і будь-яким наземним боксом істини [28].

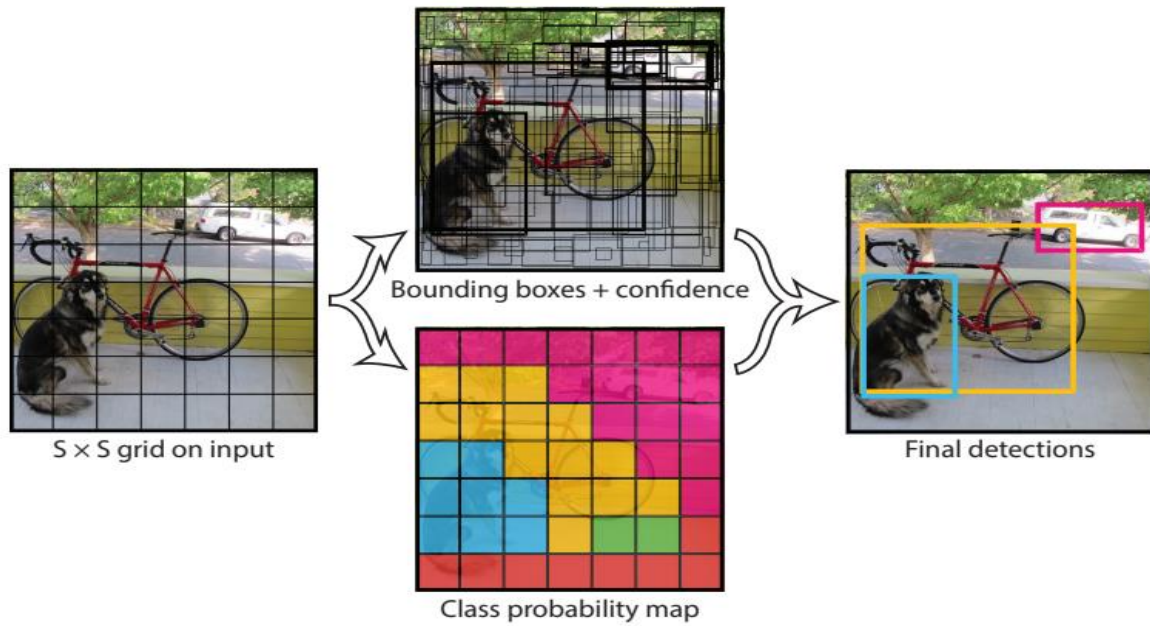


Рисунок 3.2 – Поділ зображення на сітку $S \times S$ і для кожної клітинки сітки передбачає «В» обмежувальні рамки

Наша мережева архітектура приймає GoogLeNet модель для класифікації зображень. Є 24 згорткові шари, за якими слідує 2 повністю пов'язаних шари.

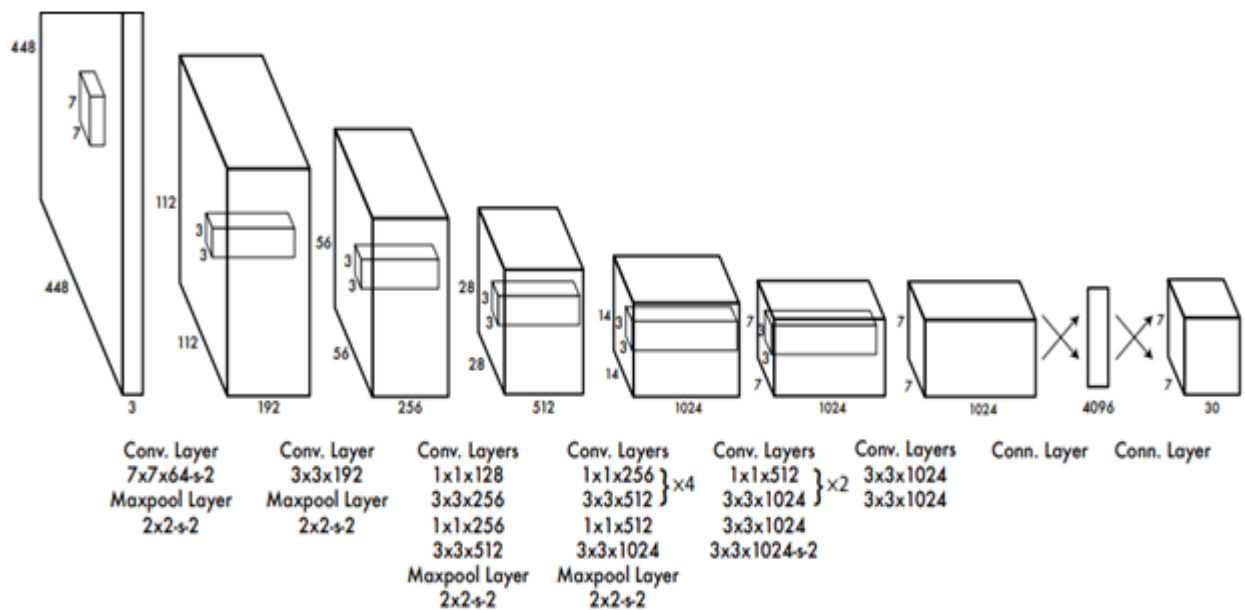


Рисунок 3.3 – Архітектура нейронної мережі, принцип навчання та виявлення об'єкта

Даний метод використовує функцію лінійної активації для кінцевого шару і всі інші шари використовують наступну лінійну активацію (рис. 3.3):

$$\varphi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0,1x, & \text{otherwise} \end{cases} \quad (3.1)$$

Відбувається оптимізація похибки суму-квадратів у результатах досліджуваної моделі, формула (3.1).

YOLO передбачає декілька обмежувальних прямокутників на клітинку сітки. Під час навчання нам потрібен лише один обмежувальний прямокутник, котрий буде відповідати за кожен об'єкт. Призначаємо один прямокутник бути «відповідальним» за передбачення об'єкта, на основі якого прогноз має найвищий поточний IOU з землею правда. Це призводить до спеціалізації між обмежувальною рамкою прямокутника. Кожен прямокутник стає кращим у передбаченні певного розміру, співвідношення сторін або класи об'єктів, покращуючи в цілому згодність.

3.3 Порівняння YOLO з іншими системами виявлення об'єктів та його обмеження

Виявлення об'єктів є основною проблемою комп'ютерного зору. Даний розділ передбачає порівняння YOLO з кількома найкращими системами виявлення, висвітлюючи ключові схожості та відмінності:

- Deformable parts models (DPM) – моделі деталей, що деформуються. Даний спосіб використовує підхід ковзного вікна для виявлення об'єктів (рис. 3.4);

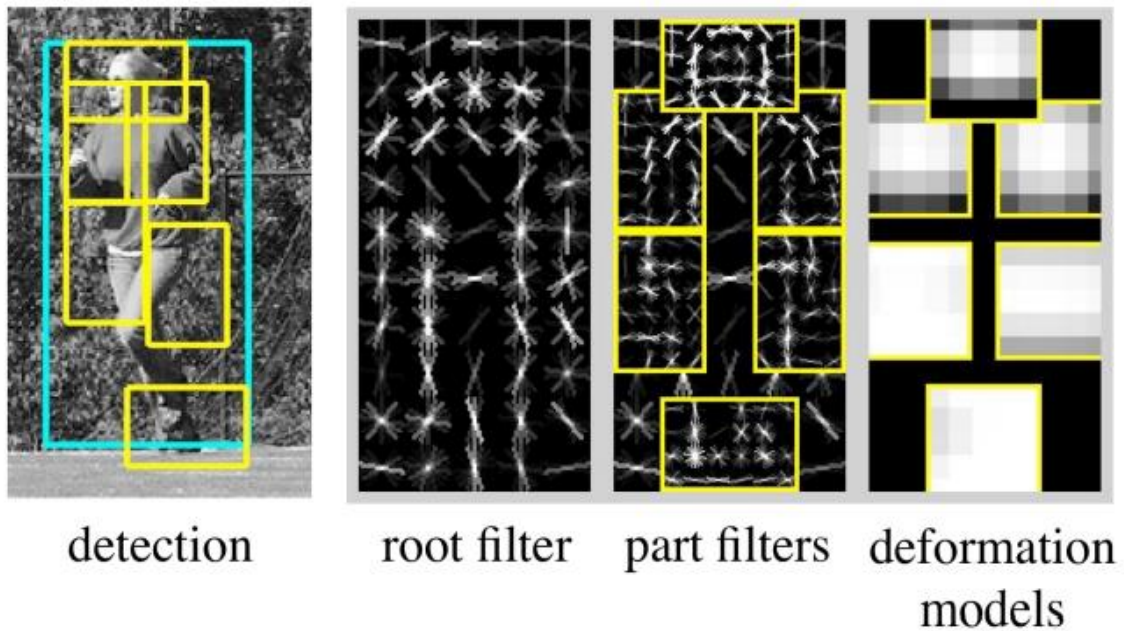


Рисунок 3.4 – Приклад розбиття зображення на рівні (DPM)

DPM використовує роз'єднаний конвеєр для вилучення статичних функцій, класифікувати регіони, передбачити обмежувальні рамки для високої оцінки регіону тощо. Наша система замінює всі ці розрізнені частини з єдиною згортковою нейронною мережею. Мережа одночасно виконує виділення ознак, передбачення обмежувальної рамки, неадекватне придушення та контекстне міркування. Замість статичних функцій мережа тренує функції і оптимізує їх для завдання виявлення. Наші уніфікована архітектура веде до швидшої та більш точної моделі ніж DPM;

– R-CNN. Модель даного типу використовує пропозиції регіонів замість розсувних вікон для пошуку об'єктів на зображеннях. Вибірковий пошук генерує потенційні обмежувальні рамки, згортка мережи витягує ознаки, SVM оцінює рамки, лінійна модель налаштовує обмежувальні рамки, а неадекватне придушення усуває повторні виявлення. Кожен етап цього процесу – це складний конвеєр, він повинен бути точно налаштований самостійно і отримана система дуже повільна, займаючи більше 40 секунд на зображення під час тестування (рис. 3.5);

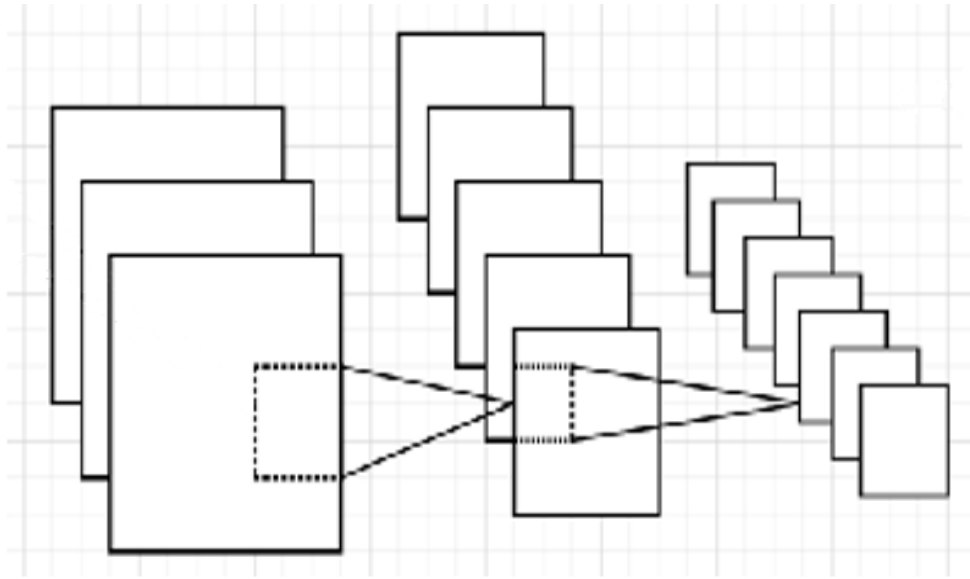


Рисунок 3.5 – Приклад розбиття зображення на рівні (R-CNN)

YOLO має певну схожість з R-CNN. Кожна клітинка пропонує потенційні обмежувальні рамки та оцінює їх бокси з використанням згорткових ознак. Проте наша система накладає просторові обмеження на пропозиції осередку сітки, які допомагають пом'якшити багаторазове виявлення одного і того ж об'єкта. Наші система також пропонує набагато менше обмежувальних квадратів, лише 98 за зображення в порівнянні з приблизно 2000 із вибіркового пошуку. І останнє, наша система поєднує ці окремі компоненти в єдину, спільно оптимізовану модель;

– MultiGrasp. Принцип YOLO за дизайном схожий на роботу виявлення захоплення за Redmon. Сітковий підхід YOLO подібний до прогнозу обмежувальної рамки заснований на системі MultiGrasp для регресії до захоплень. Однак виявлення захоплення – це багато простіше завдання, ніж виявлення об'єктів. Потрібно лише MultiGrasp щоб передбачити єдину область для охоплення для зображення, що містить один об'єкт. Для цього не потрібно оцінювати розмір, розташування, або межі об'єкта або передбачити його клас, знайдіть лише регіон, придатний для його захоплення. YOLO передбачає обидва

обмеження бокса та ймовірності класів для кількох об'єктів, кількох класів на зображенні (рис. 3.6).

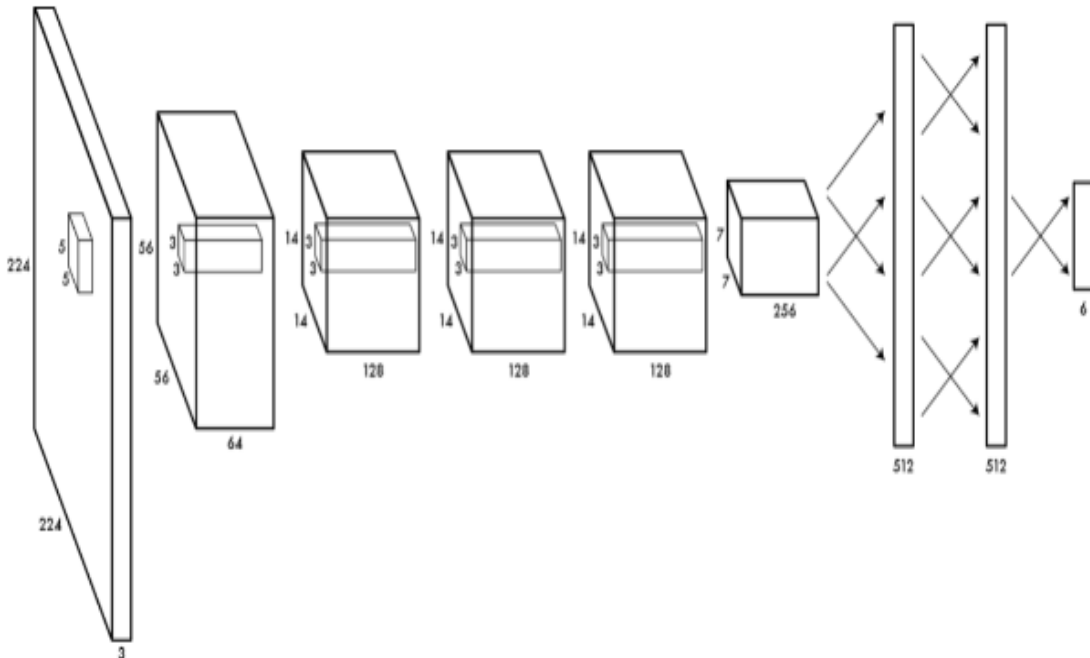


Рисунок 3.6 – Приклад розбиття зображення на рівні (MultiGrasp) [29]

3.4 Розробка програми ідентифікації об'єктів мовою програмування Python за допомогою бібліотеки OpenCV з урахуванням всіх правил асоціативної логіки

Початковий рівень. Даний етап містить в собі розуміння яким чином працює система YOLO, написання програми ідентифікації предметів які знаходяться на зображенні.

Одразу можемо зауважити, що на початковому етапі вже враховується те, що середовище для розробки – є налаштоване, а саме:

- налаштування середовища для написання програми (PyCharm, Brackets або Notepad++);
- створено документ з розширенням .names;
- побудовано файл конфігурації YOLO, розширення .cfg;
- побудовано системний файл YOLO, розширення .weights;

- камера NexiGo N960E 1080P 60FPS підготовлена до роботи;
- завантаження бібліотеки OpenCV в систему.

Зробимо простий вивід з екрану нашої камери (рис. 3.7).



Рисунок 3.7 – Вивід інформації на екран з камери NexiGo

Цікава функція, яка присутня в розробці програми – це так звани прийом «краплі» (або «blob»):

```
blob = cv2.dnn.blobFromImage(frame, 0.00392, (320, 320), (0, 0, 0),
True, crop=False)
net.setInput(blob)
outs = net.forward(output_layers)
```

Дана функція дозволяє полегшити попередню обробку зображень для класифікації глибокого навчання [8].

Наступний крок – зробимо тест нашого зображення, а саме поділимо його на декілька рівнів, збільшимо крок вибірки. Рисунок 3.8, демонструє тільки поділ нашого зображення без урахування системи YOLO.

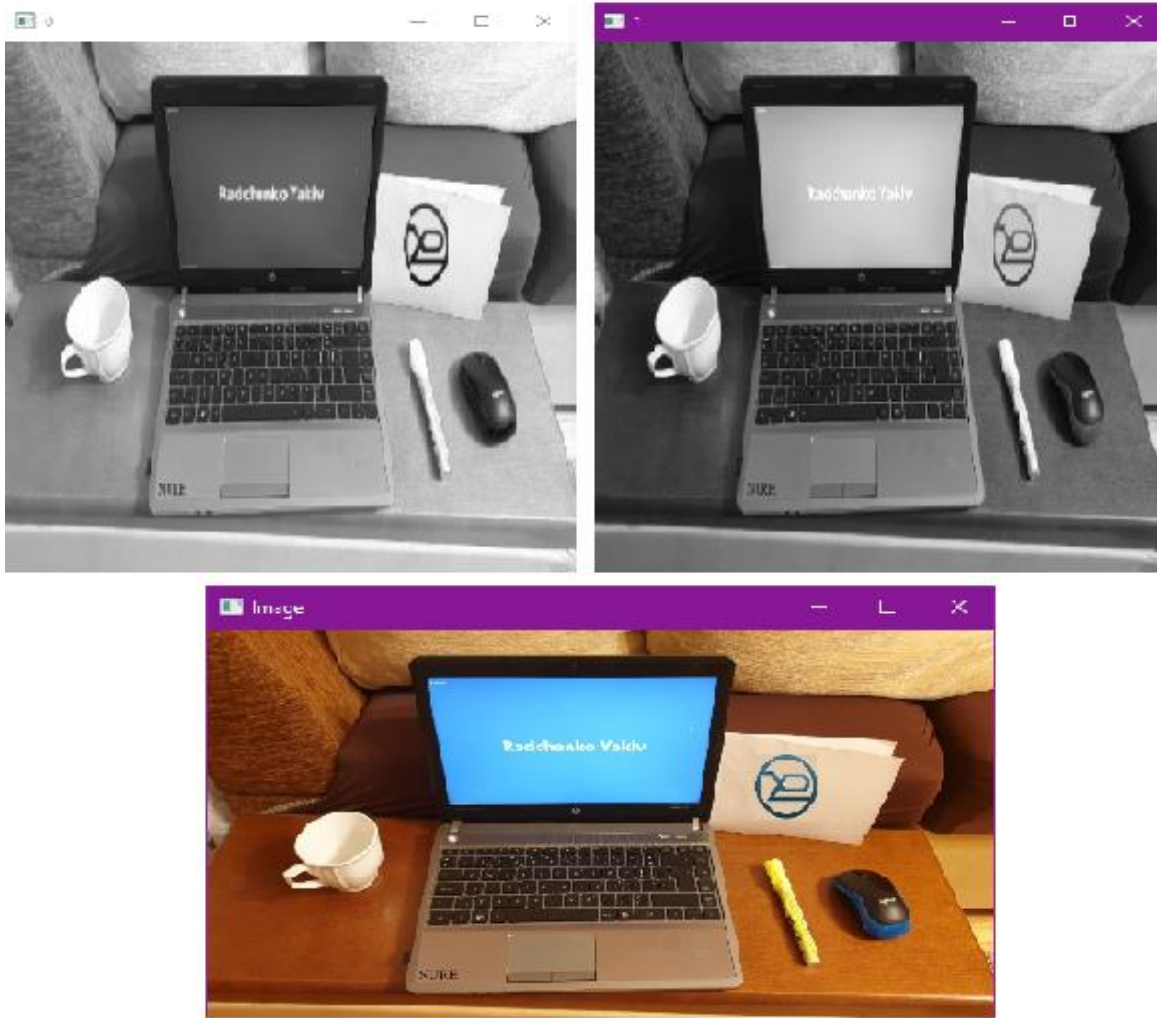


Рисунок 3.8 – Поділ зображення на декілька рівнів

Етапом для досягнення цілі служить підключення файлів конфігурації YOLO та пошук всіх схожих елементів які існують в нашій базі. Відбувається спосіб зіставлення елементів, т.з. асоціація. На рисунку 3.9 було продемонстровано код нашої програми та отриманий результат з урахуванням пошуку елементів.

Важливо зауважити, в даній роботі використовується третя версія YOLO(3). Можемо зробити аналіз рисунка 3.9, що наша програма тільки показала де ці об'єкти знаходяться, але не показала який вони мають розмір, система працює швидко і одразу показує, що на зображенні є ініціалізація трох елементів з чотирьох (філіжанка, ноутбук та мишка до нього) [5].

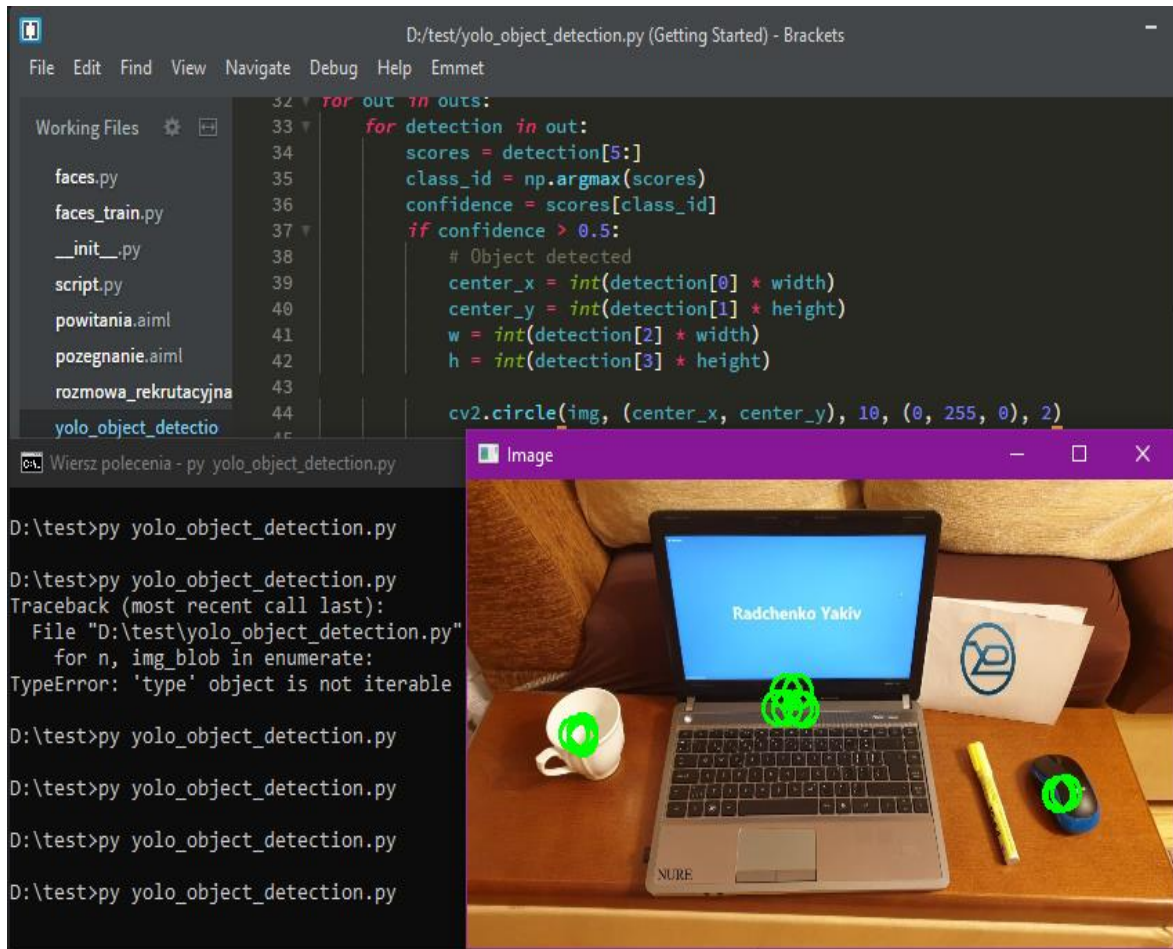


Рисунок 3.9 – Підключення файлів конфігурації YOLO, знаходження об'єктів на зображенні

Пропонуємо вдосконалити нашу програму і додати рубіж цих об'єктів (рис. 3.10). Внизу вихідного зображення продемонстровано рядок коду, яким чином відбувається зв'язок OpenCV з побудовою цих рубежів на кожному елементі, їх розмірність та вибраний колір, який ініціалізує ці об'єкти.

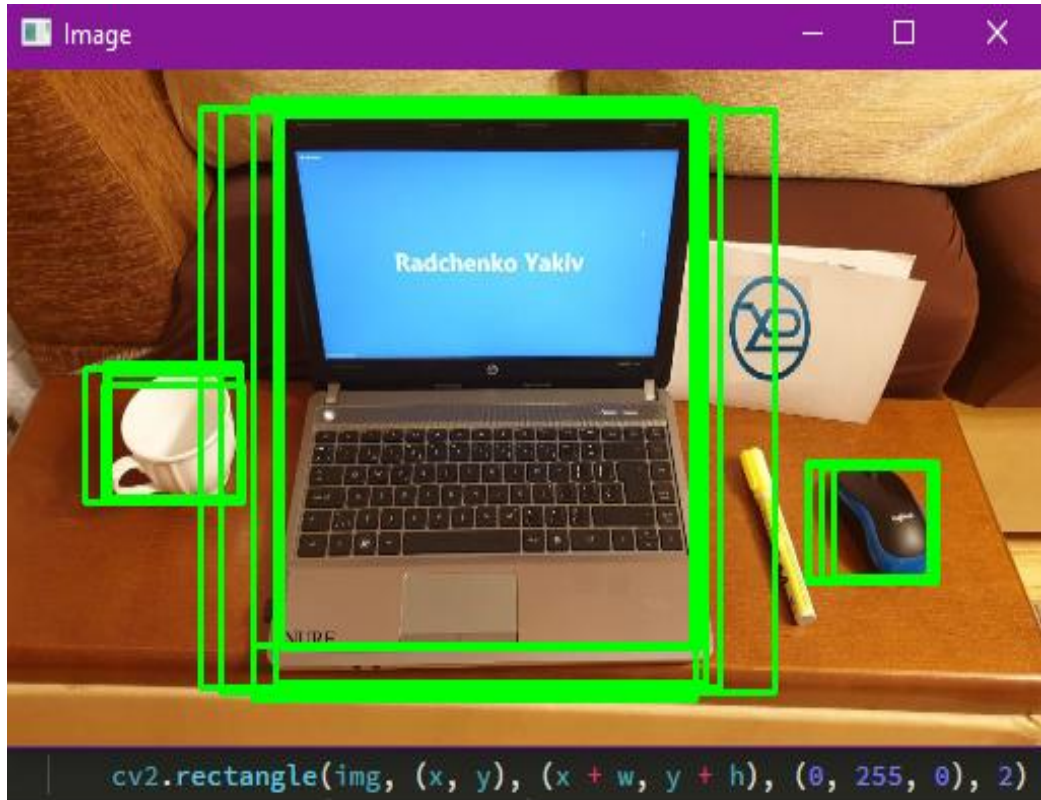


Рисунок 3.10 – Конфігурація параметрів для знаходження всього контуру об'єкта

В який спосіб з'являються обрамлення на об'єктах можемо побачити в кодї програми:

```
x = int(center_x - w / 2)
y = int(center_y - h / 2)
boxes.append([x, y, w, h])
confidences.append(float(confidence))
class_ids.append(class_id)
```

Де, x та y беруть на себе позиціонування об'єкта відносно довжини та широти. Наступним чином показано, що елемент буде в середині так званого «бокса» і має свій власний ID, який дає ініціалізацію об'єкту.

Остатнім кроком в даній роботі – є правильна ідентифікація об’єктів і саме на цьому етапі використовується правила асоціацій та пошук всіх зв’язаних елементів на зображенні (рис. 3.11).

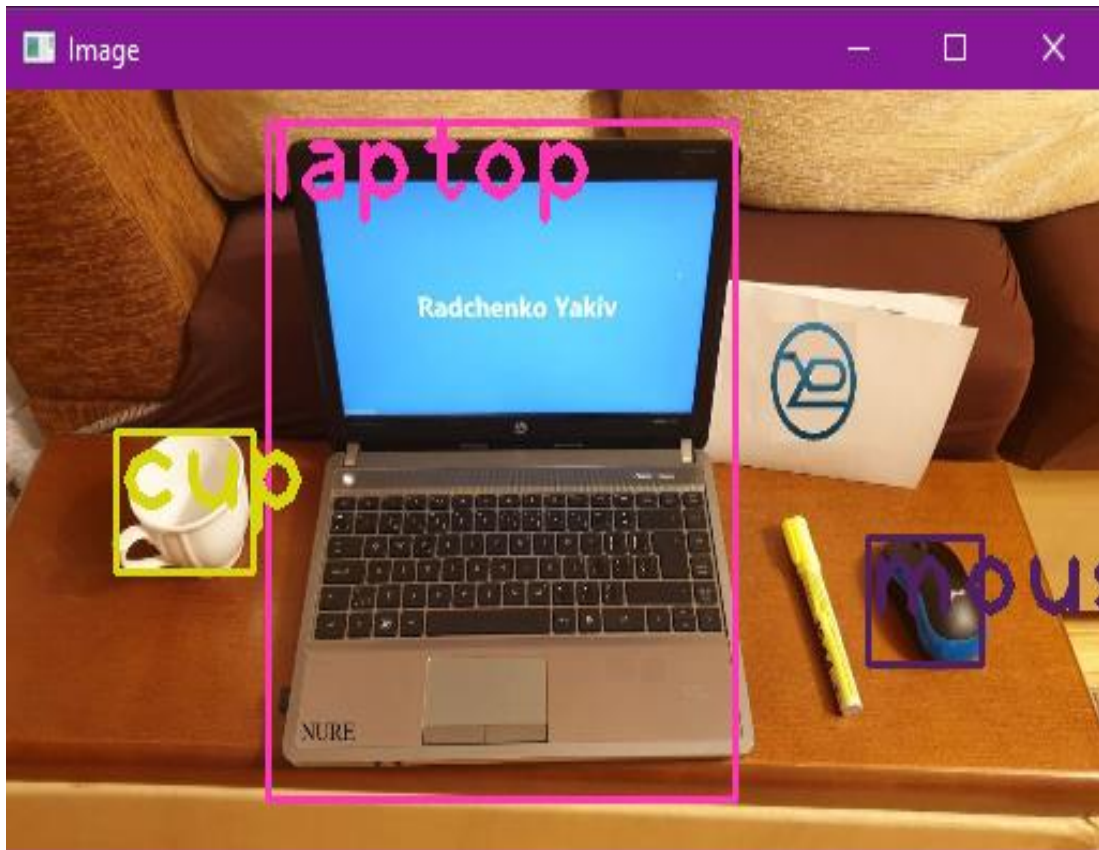


Рисунок 3.11 – Ідентифікація об’єктів на зображенні з використанням YOLO

Прив’язування до вже ініціалізованих об’єктів відбувається автоматично, приклад коду є нижче:

```
label = str(classes[class_ids[i]])
confidence = confidences[i]
color = colors[class_ids[i]]
cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
cv2.putText(frame, label + " " + str(round(confidence, 2)), (x, y + 30),
font, 3, color, 3)
```

З отриманного результату можемо зробити висновок, а саме: система YOLO накладає сильні просторові ліміти на обмеження передбачення блоків, оскільки кожна клітинка сітки передбачає лише два блоки і може мати лише один клас. Це просторове локалізування обмежує кількість найближчих об'єктів, яких наша модель може передбачити. Іншими словами, YOLO бореться з дрібними об'єктами, які об'єднуються у групи, наприклад зграї птахів.

Важлива частина недоліку даної системи – поки ми тренуємося на функції втрат, яка наближається до ефективності виявлення, наша функція втрат обробляє помилки те саме в малих обмежувальних прямокутниках порівняно з великими боксами. Невелика помилка у великому боксі як правило, є доброякісною, але як невелика помилка у малому боксі має набагато більший вплив на IOU. Наше головне джерело помилок – неправильна локалізація.

3.5 Експериментальне дослідження

Рівень адмін. Даний етап містить в собі експериментальну частину вдосконалення нашої програми. В ході всього часу ми робили дослід від отриманого нами зображення, яке просто подається в систему. Зараз ми модифікуємо нашу програму і зробимо, щоб вона працювала в реальному часі.

На рисунку 3.12 продемонстровано набір різних кухонних приборів, камера робить зйомку зверху (приблизно $\angle 90^\circ$). Це без урахування нашої програми.

Рисунок 3.13 демонструє результат роботи нашої програми та показує похибку в пошуку об'єктів та правильність ідентифікації. Програма розуміє, що на зображенні знаходяться такі об'єкти:

- ніж;
- виделка;
- філіжанка;
- ложка десертна / ложка звичайна.



Рисунок 3.12 – Вивід отриманного зображення з камери (вид зверху $\angle 90^\circ$)



Рисунок 3.13 – Отриманий результат програми

Даний дослід виявить залежність між часом яким потрібно, щоб знайти потрібні нам об'єкти та їх поправну ідентифікацію [14].

На рисунку 3.14 продемонстровано такий самий набір кухонних приборів, камера робить зйомку збоку (приблизно $\angle 45^\circ$). Це без урахування нашої програми.



Рисунок 3.14 – Вивід отриманного зображення з камери (вид збоку $\angle 45^\circ$)



Рисунок 3.15 – Знаходження всіх об'єктів та поправність ідентифікації

Рисунок 3.15 демонструє результат роботи програми і показує вже інший результат в пошуку об'єктів та правильності ідентифікації.

Можемо побачити, що під кутом у $\angle 45^\circ$ програма працює як треба і робить пошук всіх елементів, які тільки можуть знаходитися на отриманому зображенні.

На рисунку 3.15 було додатково знайдено: ніж та обідній стіл.

Останній етап дослідження покаже залежність від кута зйомки. На рисунку 3.16 продемонстровано такий самий набір кухонних приборів, камера робить зйомку збоку (приблизно $\angle 15^\circ$). Це без урахування нашої програми.



Рисунок 3.16 – Вивід отриманного зображення з камери (вид збоку $\angle 15^\circ$)

Рисунок 3.17 демонструє результат роботи програми та показує похибку зйомки від кута, пошук об'єктів та правильність ідентифікації.

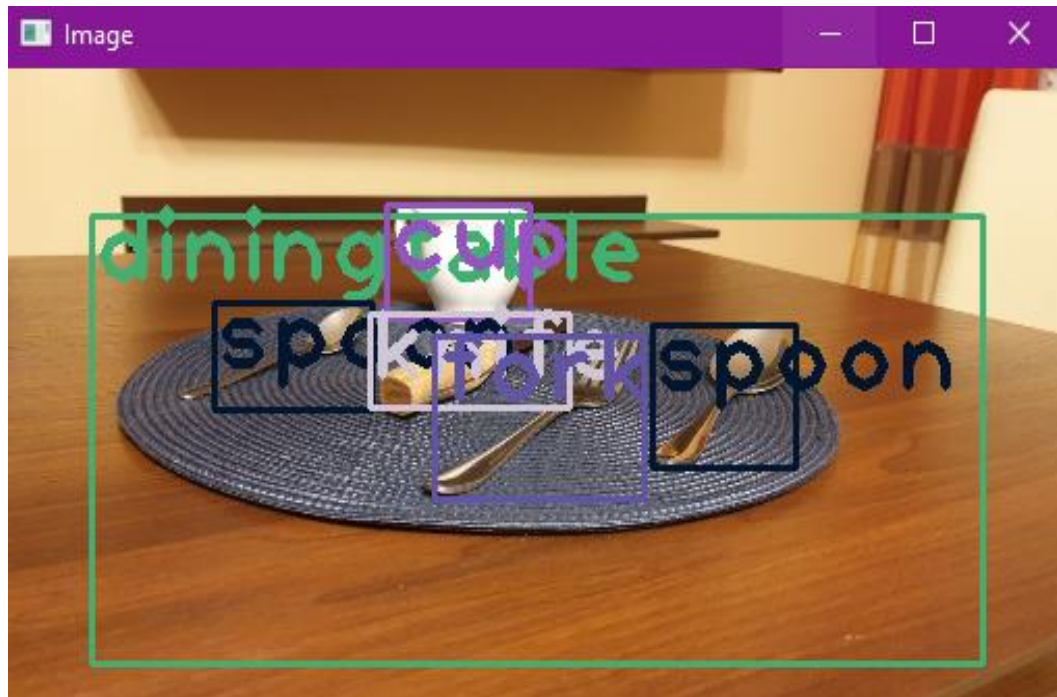


Рисунок 3.17 – Результат програми все ідентифікував вірно

На наступних рисунках (рис. 3.18 та рис. 3.19) буде продемонстровано звичайне знаходження об'єктів та подальша їх ідентифікація [5].

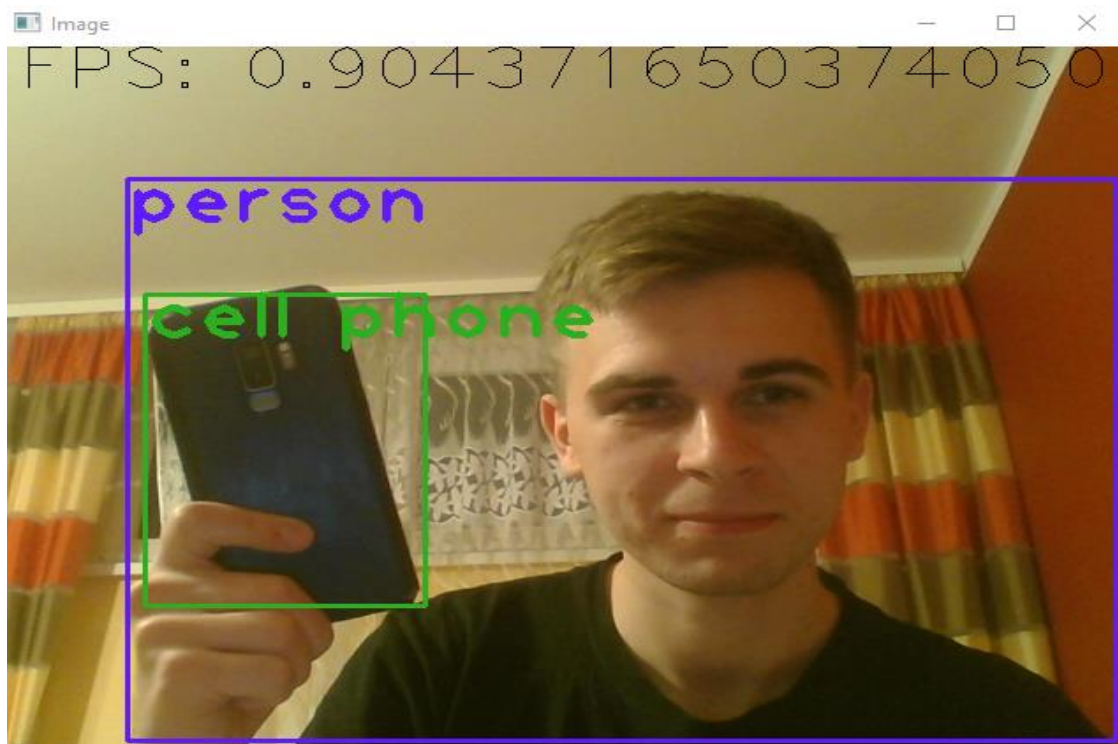


Рисунок 3.18 – Ініціалізація людини та ідентифікація телефону

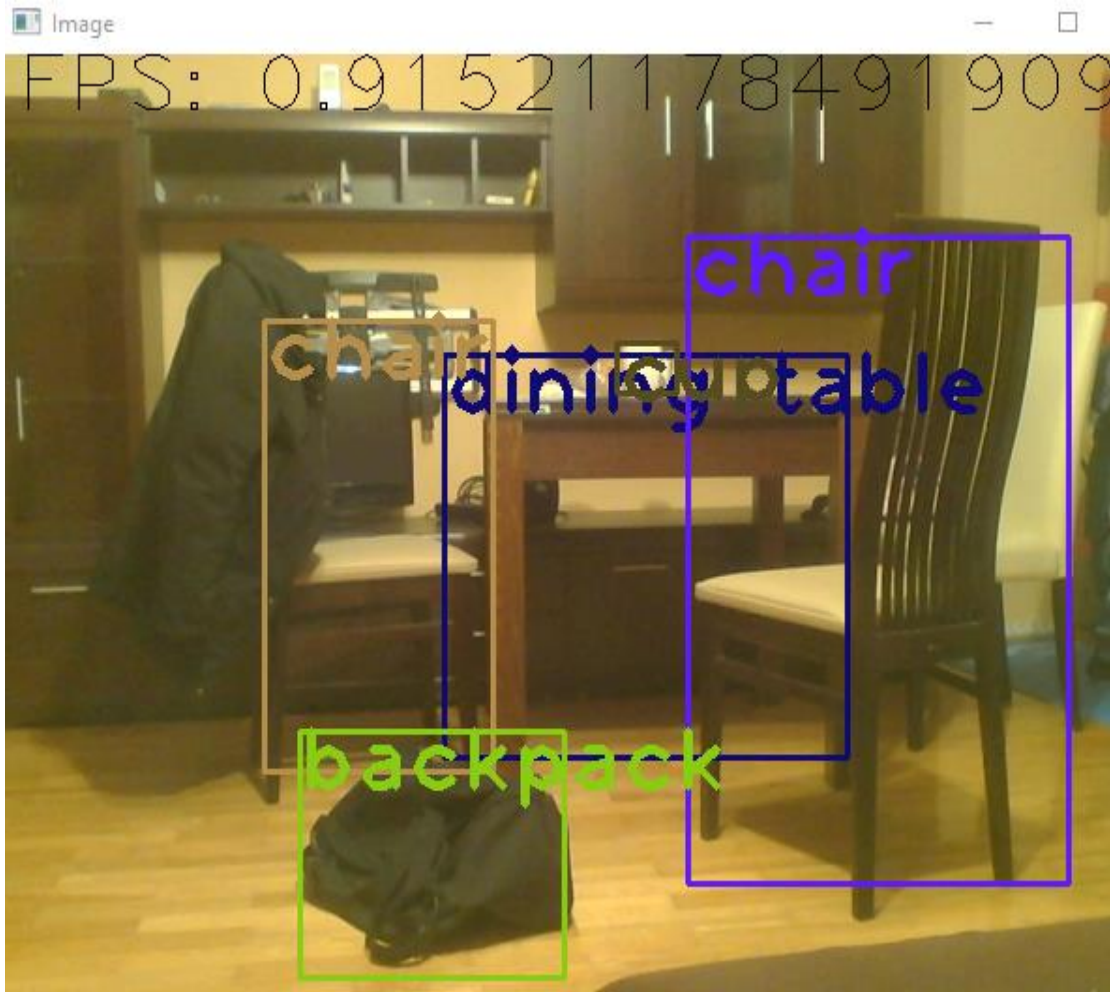


Рисунок 3.19 – Знаходження об'єктів в кімнаті та поправка їх ідентифікація

3.6 Охорона праці

Середовище для розробки та тестування програми знаходиться в лабораторії. Для комфортної і безпечної роботи з написання програми з точки зору охорони праці потрібно застосувати правильне штучне освітлення.

Згідно ДСН 3.3.6 – 042-99 робота в лабораторії з персональними комп'ютерами по категорії робіт відноситься до легких (категорія Ia – легкі фізичні роботи з енерговитратами до 139 Вт або 120 ккал/ч). Робота ведеться сидячи, не вимагає систематичного фізичного напруження і переміщення важких предметів [30].

У робочій зоні приміщення згідно з ДСН 3.3.6.042 – 99 повинні бути встановлені оптимальні значення параметрів мікроклімату. Для категорії робіт Іа в приміщенні повинні підтримуватися наступні параметри мікроклімату:

- температура повітря 22-24 °С в теплий період і 23-25 °С в холодний період;
- відносна вологість повинна бути в межах 40-60 %;
- швидкість руху повітря не повинна перевищувати 0,1 м/с.

Для підтримки потрібної температури повітря в лабораторії рекомендується встановити кондиціонер.

Згідно ДБН В.2.5 – 28 – 2006 в лабораторії застосовується сумісний тип освітлення, рекомендована освітленість від загального рівномірного штучного освітлення при роботі з монітором становить 300-500 лк при використанні газорозрядних ламп білого кольору. Коефіцієнт природного освітлення – 1,5 %. Рівень загального штучного освітлення приміщення можна перевірити за допомогою методу питомої потужності по формулі:

$$W = \frac{W_{\Sigma}}{S}, \quad (3.1)$$

де W – питома потужність, Вт/м²;

S – площа приміщення, $S = 48$ м²;

W_{Σ} – загальна потужність освітлювальної установки Вт, яка розраховується за формулою:

$$W_{\Sigma} = W_{cv} \cdot n_{cv}, \quad (3.2)$$

де W_{cv} – потужність одного світильника, $W_{cv} = 80$ Вт;

n_{cv} – кількість світильників в приміщенні, $n_{cv} = 6$ світильників.

$$W_{\Sigma} = 80 \cdot 6 = 480 \text{ Вт}, \quad (3.3)$$

$$W = \frac{480}{24} = 20 \text{ Вт/м}^2. \quad (3.4)$$

Взаємозв'язок питомої потужності (W) і освітленості (E) для приміщення площею 24 м^2 наведені в таблиці 3.1 [30].

Таблиця 3.1 – Взаємна зв'язок питомої потужності і освітленості

Площа приміщення $S, \text{ м}^2$	Споживана потужність при освітленості E , лк						
	200	300	400	450	500	550	600
17 – 25	12,2	16,3	24	32	49	65	81

Згідно таблиці 3.1 питома потужності 20 Вт/м^2 відповідає освітленість 350 лк , що задовольняє вимоги ДБН В. 2.5. – 28 – 2006. Обрана кількість світильників (6 штук) задовольняє вимоги освітленості приміщення площею 24 м^2 .

3.7 Висновки до третього розділу

Останній розділ показав залежність роботи програми від положення камери, це впливає на пошук елементів та знаходження інших, які можуть бути другорядними. Програма виконана належним чином з усіма нормами та правилами пошуку вже відомих нам об'єктів.

Розроблена програма є актуальною і може бути вдосконалена для більш великого функціонал

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було проведено аналіз особливостей асоціативних процесів та способів розпізнавання образів, зроблено аналіз дії асоціації з урахуванням кластеризації та порівняно різні алгоритми кластеризації.

Перший розділ: проведено аналіз методу асоціативних процесів та приклади застосування в роботизованих системах. Розглянуто класифікації задач розпізнавання образів на базі навчання та самонавчання нейронних мереж, зроблено порівняння різних алгоритмів кластеризації та її впливи.

Другий розділ: зроблено аналіз математичних розрахунків, з'ясовано яким чином працює штучний нейрон, знайдено недоліки та переваги різного типу штучних нейронів та розібрана схема електрична кожного з них.

Третій розділ: зроблено імплементацію програми з урахування методу асоціативних процесів, програма написана базуючись методом YOLO за використанням бібліотеки OpenCV, зроблено зіставлення архітектури нейронної мережі з різними типами моделей, пошук сильних та слабих сторін розробки різних моделей нейронних мереж, написання програми для пошуку різних об'єктів на зображенні в реальному часі та їх ідентифікація, мова програмування є Python в середовищі PyCharm. Щоб з'ясувати або ж знайти недоліки розробленої програми були зроблені експериментальні дослідження.

Дослід містить в собі такі дані: на столі маємо набір кухонних предметів (виделка, філіжанка, ніж, ложка десертна та ложка звичайна), задача розробленої програми показати недолік виявлення всіх існуючих елементів під різним кутом нахилу, звідси можемо проаналізувати, що:

– перший дослід має кут нахилу $\angle 90^\circ$. Можемо спостерігати, що програми повинна працювати належним чином, бо всі об'єкти знаходяться навпроти камери і немає жодних перешкод, щоб ініціалізувати та ідентифікувати

всі об'єкти, але ж ні. Маємо похибку або в швидкості ініціалізації та пошуку елементів на зображенні, або повільність дії зіставлення з потрібним нам об'єктом;

– другий дослід має кут нахилу $\angle 45^\circ$. Другий дослід має відмінний результат бо зміг знайти навіть те, що не було вказано (стіл, де знаходяться все предмети). Можемо побачити, що програма показала добрий результат в пошуку та поправну ідентифікацію всіх об'єктів;

– третій дослід має кут нахилу $\angle 15^\circ$. Результат є відмінний, навіть коли кут нахилу є досить малим, програма все ж таки робить максимальне зіставлення з усіма елементами які є на зображенні і показує правильну ідентифікацію всіх об'єктів.

Висновки які були отримані з експериментального дослідження показали, що: не дивлячись на метод, який використовуємо (YOLO), його швидкість та архітектуру нейронної мережі все одно буде похибка того, під яким кутом камера намагається зробити ініціалізацію об'єктів та поправно їх потім зіставити, щоб результат відподів дійсності.

В подальшій роботі можна буде вдосконалити алгоритм дії пошуку об'єктів та розробити систему зчитування інформації в декількох режимах, а саме розкладання отриманного зображення на більшу кількість рівнів, а потім зіставлення всіх отриманих результатів та їх обробка під сумою квадратів впливів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення [Текст]. – Введ. 2015-06-22. – К.: Держстандарт України, 2017. – 29 С.

2. Невлюдов, І.Ш. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології» [Текст]: навч. посіб. / І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, Г.В. Пономарьова. – Київ-58, пр. Космонавта Комарова, 1, 2016. – 320 С.

3. Методичні вказівки до випускної кваліфікаційної роботи рівня «Магістр» для студентів усіх форм навчання напряму 6.050202-Автоматизація та комп'ютерно-інтегровані технології / Упоряд. І.Ш.Невлюдов, О.В.Токарева.:– Харків: ХНУРЕ. – 2011. – 60 С.

4. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні [Електронний ресурс] : Наказ ХНУРЕ від 06 травня 2021 р. No 143. – Режим доступу : https://nure.ua/wpcontent/uploads/Main_Docs_NURE/143-vid-06.05.2021-pro-vvedennja-v-dijurishennja-vchenoi-radi-universitetu.pdf.

5. Radczenko J., Metoda porównywania uzyskanej informacji, przykład logiki asocjacyjnej / J. Radczenko // Manufacturing & Mechatronic Systems 2021 // [Text] / J. Radczenko, B.B. Євсєєв. – Wrocław, 2021. – Issue 5. – 141-144 p.

6. Теорія розпізнавання образів – основні поняття теорії розпізнавання образів [Електронний ресурс] / Режим доступу : www / URL: <https://dspace.uzhnu.edu.ua> – Загл. с екрана.

7. Дуньюэ, Ц. Управление мобильными роботами на основе нечетких моделей [Текст] / Цюй Дуньюэ // Журн. Современные проблемы науки и образования – 2007. – № 6. ч. 2 – С. 115 – 121.

8. Евсеев В.В., Разработка 3D системы технического зрения и идентификаций объектов для мобильных роботов [Текст] / В.В. Евсеев, Г.В. Фоменко И.Н. Малая // Журн. Технология приборостроения. – 2016. – №. 3, – С. 25 – 29.
9. Логика мышления – Ассоциативная память [Электронный ресурс] / Режим доступа : [www / URL: https://habr.com/ru/post/216301/](http://www.habr.com/ru/post/216301/) – Загл. с экрана.
10. Тімінський, О. Г. Технології адаптивного управління як механізм забезпечення ефективності організаційно-управлінських систем [Текст] / О.Г. Тімінський // Управління розвитком складних систем. – 2016. – № 27. – С. – 122 – 131.
11. Ashby, W.R. Design for a brain; the origin of adaptive behavior [Text] / W.R. Ashby. – New York, Wiley, 1960. – 308 p.
12. Інтелектуальні системи і технології – Навчальний посібник [Електронний ресурс] / Режим доступа : [www / URL: https://www.nam.kiev.ua/files/publications/nester-kovt-fal-2-ostanna.pdf](http://www.nam.kiev.ua/files/publications/nester-kovt-fal-2-ostanna.pdf) – Загл. с экрана.
13. Жиров, М. В. Адаптивное управление технологическими процессами с нестационарными параметрами [Текст]: Дис. д-ра техн. наук: 05.13.06 / М. В. Жиров. – М., 2004. – 511 С.
14. Радченко, Я.О. Макет з'єднання і програмна реалізація контролю зору для мобільних роботів на базі Raspberry Pi та мови програмування Python [Текст] / І.Ш., Невлюдов, В.В. Євсєєв, Я.О. Радченко // Журн. Технология приборостроения. – 2019. – №. 2, – С. 18 – 20.
15. Vaish D. Python Robotics Project [Text] / Prof. Diwakar Vaish – UK. : Published by Packt Publishing Ltd., 2018. – 328 С.
16. Радченко, Я.О. Связь нейронных сетей в автоматизированной системе управления [Текст] : тез. допв. науч.-практ. конф. (квіт. 2019) / конф.: Автоматизовані системи та комп'ютеризовані технології радіоелектронного приладобудовання / від. ред. Я. О. Радченко. – Х : 2019. – С. 23 – 24.

17. Крилов В. С. Комп'ютерний зір: інноваційний віртуальний лабораторний практикум / В. С. Крилов – Науковий часопис НПУ імені М. П. Драгоманова. Серія 2 : Комп'ютерно-орієнтовані системи навчання. – 2012. – № 13. – С. 181 – 185.

18. Сравнение библиотек компьютерного зрения для разработки систем охранного видеонаблюдения с возможностью видеоаналитики [Электронный ресурс] / Режим доступа : [www / URL: <https://cyberleninka.ru/article/n/sravnenie-bibliotek-kompyuternogo-zreniya-dlya-razrabotki-sistem-ohrannogo-videonablyudeniya-s-vozmozhnostyu-videoanalitiki/viewer>](http://www.cyberleninka.ru/article/n/sravnenie-bibliotek-kompyuternogo-zreniya-dlya-razrabotki-sistem-ohrannogo-videonablyudeniya-s-vozmozhnostyu-videoanalitiki/viewer) – Загл. с экрана.

19. Кручинін А. Розпізнавання образів з використанням OpenCV / А. Коручинін – 2011. – 171 С.

20. Working with Images - Emgu CV: OpenCV in .NET (C#, VB, C++ and more) [Електронний ресурс] / Working with Images – Режим доступу: http://www.emgu.com/wiki/index.php/Working_with_Images

21. Невлюдов, І.Ш. Математичне та алгоритмічне забезпечення автоматизації проектування модульних конструкцій робототехнічних засобів [Текст] / І.Ш. Невлюдов, В.В. Євсєєв, А.О. Функендорф, М.О. Голіков // Журн. Технология приборостроения. – 2016. – №. 3, – С. 10 – 15.

22. Асоціативні процесори з паралельно-послідовною обробкою даних – Навчальний посібник [Электронный ресурс] / Режим доступа : [www / URL: <http://ir.lib.vntu.edu.ua/handle/123456789/30513>](http://ir.lib.vntu.edu.ua/handle/123456789/30513) – Загл. с экрана.

23. Евсеев, В.В. 3D проектирование и моделирование конечностей зооморфного робота [Текст] / В.В. Евсеев, В.И. Роменский, Н.В. Замирец В.Э. Салиева, // Журн. Технология приборостроения. – 2017. – №. 2, – С. 10 – 14.

24. Как работает Object Tracking на YOLO и DeepSort – habr.com [Электронный ресурс] / Режим доступа : [www / URL: <https://habr.com/ru/post/514450/>](https://habr.com/ru/post/514450/) – Загл. с экрана.

25. Євсєєв, В.В., Модели автоматизированного проектирования технологического процесса сборки [Текст] / В.В. Євсєєв, И.В. Филипенко, С.С. Милютин // Журн. Технологический аудит и резервы производства. – 2015. – №. 1/2 (21). – С. 4 – 8.
26. Управляем шаговыми движками и DC моторами, L298 и Raspberry Pi [Электронный ресурс] / Режим доступа : [www / URL: https://ph0en1x.net/87-l298-and-raspberry-pi-stepper-and-dc-motor.html](http://www.ph0en1x.net/87-l298-and-raspberry-pi-stepper-and-dc-motor.html) – Загл. с экрана.
27. Євсєєв В.В, Проектування мобільних роботів на базі одноплатних комп'ютерів Raspberry Pi і мови Python [Текст] : підр. посібник / В.В. Євсєєв, І.Ш. Невлюдов, А.О. Андрусевич; НАУ. – 1-е вид., – Київ. : Фактор, 2020. – 257 С. – (Наукова книга).
28. Nachour O. Pathplanning of Autonomous Mobile robot / O. Nachour // International Journal of Systems Applications, Engineering and Development. – 2009. – Issue 4. – P. 178 – 190.
29. Пишем свою нейросеть: пошаговое руководство [Электронный ресурс] / Режим доступа : [www / URL: https://proglib.io/p/neural-nets-guide](http://www.proglib.io/p/neural-nets-guide) – Загл. с экрана.
30. Методичні вказівки до виконання розділу «Охорона праці» у випускних роботах ОКР «бакалавр» усіх форм навчання / Упоряд.: Б.В.Дзюндзюк, В.А.Айвазов, Т.Є.Стиценко. – Харків: ХНУРЕ, 2012. – 28 С.