

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ПОЛІСЬКИЙ НАЦІОНАЛЬНИ УНІВЕРСИТЕТ

Факультет інформаційних технологій, обліку та фінансів  
Кафедра комп'ютерних технологій  
і моделювання систем

Кваліфікаційна робота  
на правах рукопису

Третяк Владислав Сергійович  
(прізвище, ім'я, по батькові здобувача освіти)

УДК 004.81:378

## **КВАЛІФІКАЦІЙНА РОБОТА**

Інформаційна система вивчення основ штучного інтелекту

(тема роботи)

126 «Інформаційні системи та технології»

(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

---

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи  
Николюк Ольга Миколаївна,  
д.е.н., професор

Житомир – 2024

Висновок кафедри \_\_\_\_\_

за результатами попереднього захисту: \_\_\_\_\_

Протокол засідання кафедри \_\_\_\_\_

№ \_\_\_\_\_ від «\_\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_ р.

Завідувач кафедри \_\_\_\_\_

\_\_\_\_\_

(науковий ступінь, вчене звання)

\_\_\_\_\_

(підпис)

\_\_\_\_\_

(прізвище, ім'я, по батькові)

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_\_\_ р.

### Результати захисту кваліфікаційної роботи

Здобувач вищої освіти \_\_\_\_\_ захистив (ла)

(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою \_\_\_\_\_

за шкалою ECTS \_\_\_\_\_

за національною шкалою \_\_\_\_\_

Секретар ЕК

\_\_\_\_\_

(науковий ступінь, вчене звання)

\_\_\_\_\_

(підпис)

\_\_\_\_\_

(прізвище, ім'я, по батькові)

## АНОТАЦІЯ

Третяк В.С. Інформаційна система вивчення основ штучного інтелекту – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота присвячена розробленню інформаційної системи для підвищення рівня поінформованості щодо основ штучного інтелекту, основних принципах та методах його роботи. Окрім розробки безпосередньо навчального контексту системи у кваліфікаційній роботі передбачено математичне забезпечення процесу оцінювання успішності користувача.

У рамках даного дослідження розроблено інформаційну систему, що складається з таких компонентів:

- модуль теоретичних знань, який містить систематизовану інформацію про основні поняття штучного інтелекту, такі як машинне навчання;

- модуль практичних завдань, що містить низку практичних завдань, які дозволяють користувачам застосувати здобуті теоретичні знання на практиці та закріпити вивчений матеріал;

- модуль оцінювання, який включає тести, вікторини та інші інструменти для оцінки знань користувача та моніторингу його прогресу.

Науковою новизною даного дослідження є розробка інформаційної системи, яка інтегрує теоретичні знання про штучний інтелект з практичними завданнями та інструментами оцінювання. Інформаційна система створює сприятливі умови для вивчення основ штучного інтелекту користувачами різного рівня підготовки та гарантує ефективність і гнучкість навчання.

Розроблена інформаційна система може стати ефективним інструментом для викладачів, студентів та всіх, хто хоче вивчати основи штучного інтелекту. Вона може бути використана як частина навчального курсу або для самостійного вивчення чи підготовки до професійної діяльності, пов'язаної зі штучним інтелектом.

Ключові слова: штучний інтелект, інформаційна система, машинне навчання, комп'ютерний зір, нейронна мережа, освіта.

## SUMMARY

Tretiak V.S. Information system for studying the basics of artificial intelligence - Qualification work on the rights of the manuscript.

The qualification work is devoted to the development of a model of an information system for raising awareness of the basics of artificial intelligence, the basic principles and methods of its work. In addition to the development of the educational context of the system, the qualification work provides mathematical support for the process of assessing user performance.

As part of this study, a model of an information system was developed, consisting of the following components

- a module of theoretical knowledge containing systematized information about the basic concepts of artificial intelligence, such as machine learning;
- a module of practical tasks, which contains a number of practical tasks that allow users to apply the acquired theoretical knowledge in practice and consolidate the material studied;
- an assessment module that includes tests, quizzes, and other tools for assessing user knowledge and monitoring their progress.

The scientific novelty of this study is the development of an information system that integrates theoretical knowledge about artificial intelligence with practical tasks and assessment tools. The information system creates favorable conditions for studying the basics of artificial intelligence by users of different levels of training and guarantees the effectiveness and flexibility of training.

The developed information system can become an effective tool for teachers, students, and anyone who wants to learn the basics of artificial intelligence. It can be used as part of a training course or for self-study or preparation for professional activities related to artificial intelligence.

Keywords: artificial intelligence, information system, machine learning, computer vision, education.

## З М І С Т

АНОТАЦІЯ	3
SUMMARY	4
ЗМІСТ	5
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП	7
Розділ 1.	7
АНАЛІЗ ОСОБЛИВОСТЕЙ ВИВЧЕННЯ ОСНОВ ШТУЧНОГО ІНТЕЛЕКТУ	8
1.1 Роль, інформаційні потреби і визначення процесу вивчення основ штучного інтелекту	8
1.2 Моделювання процесу вивчення основ штучного інтелекту	11
Висновки до першого розділу	14
Розділ 2.	15
ПРОЕКТУВАННЯ СИСТЕМИ ВИВЧЕННЯ ОСНОВ ШТУЧНОГО ІНТЕЛЕКТУ	15
2.1 Моделювання системи вивчення основ штучного інтелекту	15
2.2 Математичне забезпечення інформаційної системи вивчення основ штучного інтелекту	21
Висновки до другого розділу	24
Розділ 3.	25
РЕАЛІЗАЦІЯ МОДЕЛІ СИСТЕМИ ПРОЦЕСУ ВИВЧЕННЯ ОСНОВ ШТУЧНОГО ІНТЕЛЕКТУ	25
3.1 Реалізація системи вивчення основ штучного інтелекту	25
3.2 Реалізація математичної моделі оцінювання успішності вивчення основ штучного інтелекту	34
3.3 Інтерфейс інформаційної системи вивчення основ штучного інтелекту	33
3.4 Керівництво користувачу інформаційної системи вивчення основ штучного інтелекту	39
Висновки до третього розділу	40
ВИСНОВКИ	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ШІ – штучний інтелект;

IDEF0 – (англ. Icam DEFinition for Function Modeling) – методологія функціонального моделювання і графічного опису процесів;

IDEF3 – (англ. Integrated DEFinition for Process Description Capture Method) – методологія моделювання та стандарт документування процесів, що відбуваються в системі;

UML (англ. Unified Modeling Language) – уніфікована мова моделювання;

HTML – (англ. HyperText Markup Language) – мова гіпертекстової розмітки;

CSS – (англ. Cascading Style Sheets) – каскадні таблиці стилів;

KNN – (англ. k-nearest neighbours algorithm) – метод К-найближчих сусідів.

## ВСТУП

Технології штучного інтелекту стрімко набирають обертів і наразі є потужним інструментом для вирішення складних завдань щодо спрощення бізнес-процесів та повсякденного життя людей. Розуміння принципів роботи штучного інтелекту стає важливим фактором його ефективного використання в різних сферах людської діяльності. З огляду на складність відповідної технології розробка інформаційних систем, що дозволяють вивчати основи штучного інтелекту, допоможе людям з різним освітнім рівнем ознайомитися з нею та навчитися раціонально її використовувати. З огляду на зазначене, обрана тема кваліфікаційної роботи є актуальною та такою, що сприятиме формуванню базових знань та компетентностей роботи зі штучним інтелектом.

Метою кваліфікаційної роботи є підвищення рівня поінформованості в основах штучного інтелекту, його основних принципах роботи, методах. Для досягнення зазначеної мети необхідно виконати такі завдання:

- аналіз та моделювання предметної області дослідження;
- проектування інформаційної системи вивчення основ штучного інтелекту;
- формалізація процесу оцінювання успішності користувачів системи;
- реалізація інформаційної системи вивчення основ штучного інтелекту.

Науковою новизною дослідження є розробка завдань у рамках інформаційної системи вивчення основ штучного інтелекту, у яких інтегровано технологію штучного інтелекту, виконання яких дає можливість роз'яснення принципів роботи штучного інтелекту користувачм будь-якого базового рівня підготовки.

Предметом дослідження є методи, засоби, інструментів, що використовуються у процесі проектування, моделювання та реалізації інформаційної системи вивчення основ штучного інтелекту.

Об'єкт дослідження – процес застосування методів, засобів та інструментів проектування, моделювання та реалізації інформаційної системи вивчення основ штучного інтелекту.

## **Розділ 1. АНАЛІЗ ОСОБЛИВОСТЕЙ ВИВЧЕННЯ ОСНОВ ШТУЧНОГО ІНТЕЛЕКТУ**

### **1.1 Роль, інформаційні потреби і визначення вивчення основ штучного інтелекту**

Наприкінці ХХ – на початку ХХІ ст процес розвитку технологій значно активізувався. Людство за цей проміжок часу почало реалізовувати все те, що неможливо було реалізувати ще п'ятсот або більше років тому. Одним з викликів періоду стрімкого технологічного розвитку було розроблення штучного мозку, який міг би так само, як людина, розмовляти, розуміти оточуюче середовище, навчатися спочатку вирішувати прості проблеми, а вже після цього розв'язувати складні задачі. Безпосередньо штучний інтелект (ШІ) – це галузь комп'ютерних наук, яка фокусується на розробці машин і систем, здатних виконувати завдання, що зазвичай вимагають залучення людського інтелекту, зокрема, навчання, вирішення суспільних проблем, прийняття рішень. В основі ШІ лежить ідея про створення машин, здатних мислити і міркувати як люди, вчитися на власному досвіді і з часом покращувати власну продуктивність. Це означає, що системи штучного інтелекту можуть аналізувати дані, вчитися на них, робити прогнози та приймати рішення на основі результатів навчання, дозволяючи машинам виконувати завдання з більшою точністю, швидкістю та ефективністю, ніж люди [6].

Наразі ШІ став частиною об'єктивної реальності сучасного суспільства. Переважна більшість сфер економічної діяльності – від пошуку даних про ринок до виявлення ризиків для пацієнтів та самих розробників моделей штучного інтелекту – користуються цією технологією. ШІ дозволяє спростити якщо не всю, то, принаймні, частину роботи, за яку відповідальна людина. З огляду на невід'ємність ШІ від сучасного світу, попит на фахівців з розробки відповідних технологій стрімко зростає. За результатами опитування проведеного у вересні-жовтні 2023 р., у якому приймали участь вчителі, які викладають в школі, та учні, які на момент опитування навчались у 8-11 класі встановлено наступне [7]:



- більшість вчителів та учнів мають досвід користування сервісами ІІІ.
- У більшості випадків респонденти задоволені отриманим досвідом;
- вчителі зазначають, що використовували сервіси ІІІ в своїй діяльності з метою підготовки до занять, створення тестів для домашнього завдання, при проведенні занять, а також для перевірки знань учнів і, навіть, у позакласній роботі. Частина вчителів залучала учнів до використання ІІІ;
  - учні, говорячи про досвід користування ІІІ, найчастіше зазначали його допомогу при підготовці домашнього завдання;
  - респонденти відмітили ключову характеристику роботи з ІІІ – здатність робити навчальний процес цікавим та захоплюючим [7].

Зважаючи на зазначене, вагома кількість інтернет-ресурсів спеціалізується на навчанні у відповідній області знань. Ці ресурси роз'яснюють, як саме працює штучний інтелект, якими методами він тренується та як його можна використовувати в майбутньому. У табл. 1.1 відображено деякі ресурси спрямовані на навчання основам ІІІ; компетентності, які вони формують; методи навчання, що використовуються для вивчення ІІІ; недоліки відповідних курсів.

Таблиця 1.1 – Особливості продуктів з вивчення основ штучного інтелекту

Особливість	Ресурси			
	Hour of code	Дія.Освіта	Grow with Google	SOFTSERVE
Методи навчання	Відеоматеріали, інтерактивні ігри, можливість програмування, яке охоплює ІІІ	Освітній серіал, фінальний тест	3 години матеріалу у форматі 5-ти хвилинних відео	14 занять, що складаються з теоретичної бази та практичних прикладів; практичних завдань
Сформовані знання та компетентності	Знання як працює штучний інтелект, вплив штучного інтелекту на світ, потенціал штучного інтелекту	Робота з алгоритмами ІІІ, машинний аналіз тексту, робота з даними, моделювання поведінки клієнтів на основі ІІІ,	Як користуватись ІІІ без технічного досвіду	Розуміння базових підходів до методів ІІІ, практичний досвід роботи з бібліотеками мови Python, які забезпечують програмну реалізацію алгоритмів ІІІ

		робота з чат-ботами		
Ціна	Безкоштовно	Безкоштовно	Безкоштовно	8599 грн, термін проходження 2 місяці
Недоліки	На сторінці курсу відбувається інформаційний хаос /все знаходиться на одній сторінці веб-сайту	Тільки один курс, який пропонує короткий вступ у тему штучного інтелекту	Один курс, навчання тільки за відеоматеріалами	Один курс, висока ціна

Більшість освітніх ресурсів пропонують навчити користуватись технологією ШІ вже за готовими методиками, майже не демонструють, як саме працює ця технологія та передбачають ознайомлення з основними концепціями і поняттями ШІ на основі великої кількості відео-лекцій, вступного курсу про штучний інтелект та знаходження теоретичного матеріалу разом з завданнями на одній сторінці. Велика кількість ресурсів не впроваджує гейміфікацію в навчальний процес, переважно обмежуючись лише відеоматеріалами. Проте, важливою є оптимізація розподілу викладацького матеріалу: спочатку слід формування розуміння основ ШІ, а вже потім поступово давати завдання для закріплення пройденого уроку. Крім того, доцільним вбачається зменшення кількості відео-матеріалів та скорочення їх тривалості; зробити курс цікавим та зрозумілим для сприйняття, стимулюючи користувачів продовжувати вивчення і розуміти свій прогрес на уроці. Тому, вивчення ШІ повинно бути більш різноманітним, зрозумілим та вдосконаленим в плані розуміння безпосередньо роботи інтелекту.

Таким чином, ефективна система навчання основами ШІ має передбачати надання цікавого, зрозумілого матеріалу та використання сучасних методів викладання. Навчання повинно бути покроковим, не орієнтуватись на швидкість надання користувачеві необхідної інформації, безпосередньо матеріал не повинен бути громіздкий, а подаватися невеликими обсягами. За результатами аналізу аналогів встановлено, що необхідною функцією системи навчання ШІ є розробка курсу в цікавій ігровій формі, яка заохочуватиме користувача проходити

завдання, читати невеликі довідки або розв'язувати тести. Такий підхід забезпечить розуміння, як саме працює ШІ, та розвіє міф про складність цієї технології.

## 1.2 Моделювання системи процесу вивчення основ штучного інтелекту

Особливості бизнес-процесів у рамках досліджуваної предметної області (навчання основам штучного інтелекту) можна продемонструвати на моделі IDEF0 – методології функціонального моделювання, яка використовується для створення функціональної моделі, що відображає структуру і функції системи, а також потоки інформації і матеріальних об'єктів, що зв'язують ці функції [9].

На рис. 1.1 представлено функціональну модель системи навчання основам штучного інтелекту. Вхідними потоками системи визначено відео-лекції, практичні завдання, теоретичний матеріал; механізмами реалізації – програмно-апаратне забезпечення та безпосередньо викладач; інструментами управління – силабуси та відповідні законодавчо-нормативні акти, ДСТУ; вихідними потоками – знання ШІ, практичні навички, результати проходження завдань та тестів.

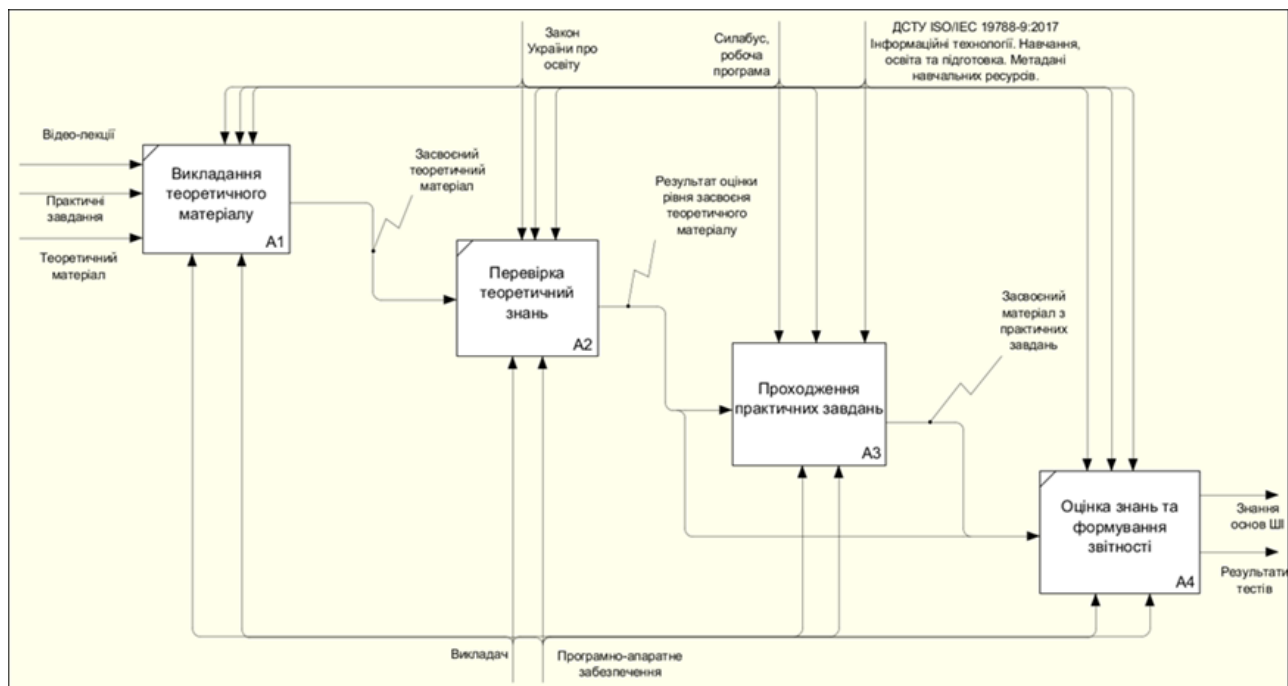


Рисунок 1.1 – IDEF0-модель системи навчання основам штучного інтелекту

На діаграмі відображено чотири функціональні блоки, які відповідають за різні елементи процесу вивчення ІІІ та описують його. Перший блок – «Викладання теоретичного матеріалу», у рамках якого користувач ознайомлюється з теоретичним матеріалом, який може надаватись у вигляді відео-лекцій, теоретичними матеріалами та порадами викладача. Другий блок – «Перевірка теоретичних знань», у межах якого відбувається проходження невеликого опитування щодо засвоєного матеріалу; формується оцінка, яка визначить рівень підготовки до виконання практичних завдань. Третій блок передбачає проходження практичних завдань, які будуть мати гейміфіковані ігрові практики та механізми. З попереднього блоку береться оцінка, що дозволить користувачу отримати певний рівень доступу. Четвертий блок – «Оцінка знань та формування звітності», де формується остаточна оцінка за допомогою оцінювання рівня засвоєння та кількості проходження практичних завдань.

Для більшого розуміння в предметній області проведено декомпозицію третього функціонального блоку «Проходження практичних завдань» (рис. 1.2), який включає передбачає організацію інтерактивних взаємодій для того, щоб користувач мав змогу взаємодіяти з задачею та вирішувати її.

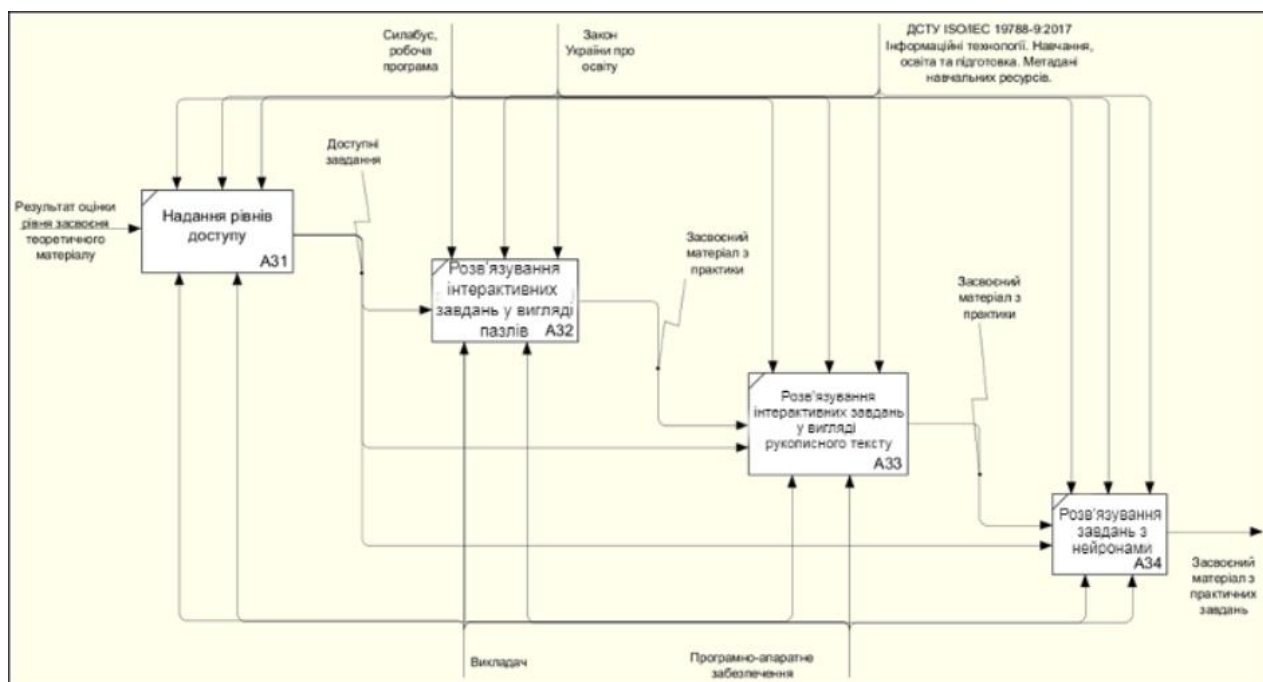


Рисунок 1.2 – Декомпозиція функціонального блоку «Проходження практичних завдань»

Зазначений функціональний блок складається з чотирьох функцій, а саме:

- «Надання рівнів доступу». Здобутий результат проходження тестів, дає змогу отримати індивідуальний набір практичних завдань;
- «Розв'язування інтерактивних завдань у вигляді пазлів»;
- «Розв'язування інтерактивних завдань у вигляді рукописного тексту»;
- «Розв'язування завдань з нейронами».

Завершення попередніх задач дає змогу працювати з нейронами. На виході отримується фінальна оцінка з проходження курсу.

Крім діаграми IDEF0, на основі якої проведено функціональне моделювання предметної області, важливим є аналіз відповідних бізнес-процесів на базі методології процесного моделювання створювання та опис діаграми IDEF3. IDEF3 – це методологія опису процесів, що розглядає послідовність виконання і причинно-наслідкові зв'язки між ситуаціями і подіями для структурного представлення знань про систему [12]. На рис. 1.3 наведено IDEF3-діаграму, яка демонструє процес вивчення основ штучного інтелекту.

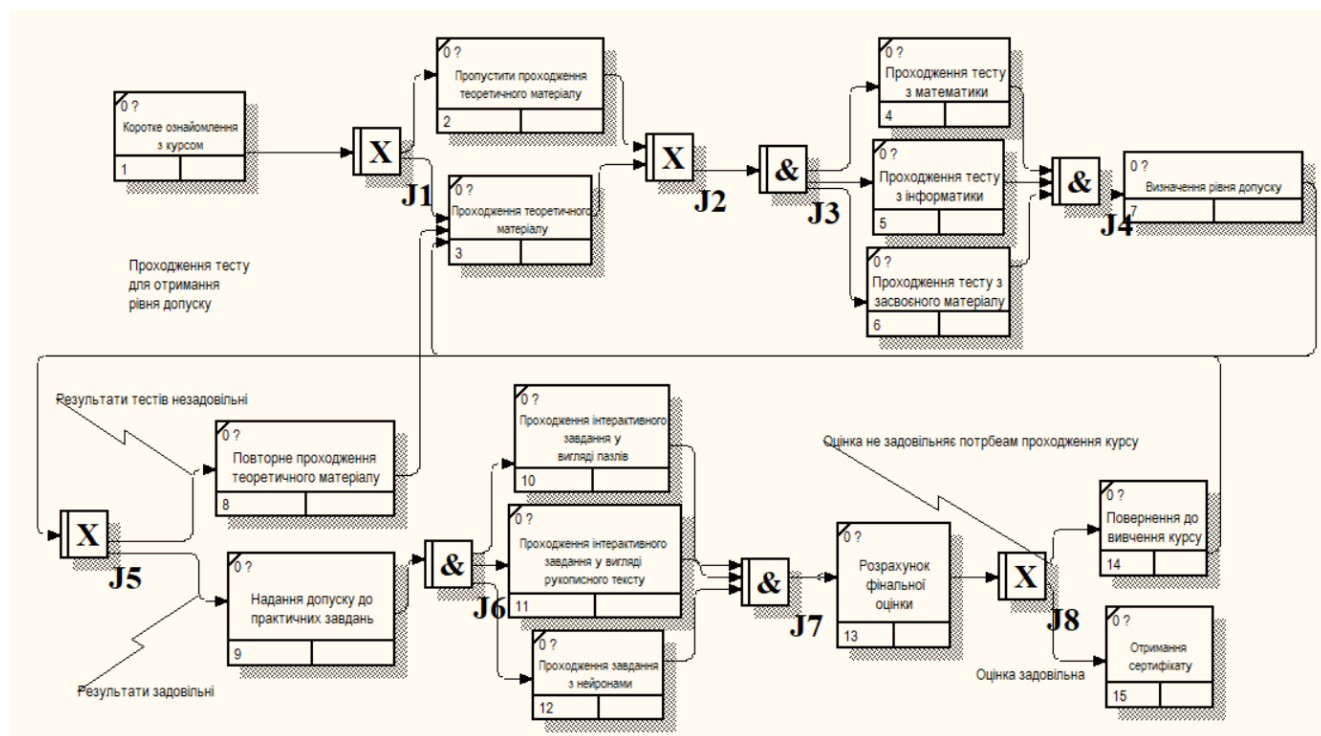


Рисунок 1.3 – Діаграма IDEF3 процесу вивчення основ штучного інтелекту

Процес вивчення основ штучного інтелекту починається, з короткого ознайомлення з курсом, зокрема, його описом, цілями та структурою. Далі користувач може або пропустити вивчення теоретичного матеріалу (у разі, якщо він має відповідні знання), або ознайомитись із теоретичним матеріалом, який необхідний для подальшого навчання. Після засвоєного теоретичного матеріалу, користувач повинен пройти три тести, а саме: з математики, інформатики та тест з засвоєного матеріалу у рамках вивчення теоретичних основ ШІ. На основі результатів тестів, якщо оцінка задовільна, йому надається доступ до проходження практичних завдань, і іншому випадку – здобувач освіти направляється на повторне проходження теоретичного матеріалу. Далі користувач виконує три практичні завдання, які відповідають навчальному матеріалу курсу. Під час формування фінальної оцінки враховуються оцінки і тестів і практичних завдань. Якщо оцінка задовільна, учень отримує сертифікат про завершення курсу; якщо оцінка не задовольняє потрібним вимогам, користувачу надається можливість повторного проходження курсу.

### **Висновки до першого розділу**

В першому розділі проаналізовано інформаційні потреби предметної області дослідження та визначено плюси та мінуси конкурентів. Проведено функціональне моделювання предметної області: сформульовано контекстну діаграму IDEF0. Для розглядання послідовності виконання і причинно-наслідкових зв'язків між ситуаціями і подіями в системі, було створено та описано діаграму IDEF3.

## Розділ 2. ПРОЕКТУВАННЯ СИСТЕМИ ВИВЧЕННЯ ОСНОВ ШТУЧНОГО ІНТЕЛЕКТУ

### 2.1 Моделювання системи вивчення основ штучного інтелекту

У рамках даного дослідження розроблено дві веб-сторінки, які містять в собі практичні завдання з розпізнавання малюнків та класифікація зображень. Для розрахунку фінальної оцінки, яка також являється оцінкою успішності, розроблена окрема веб-сторінка, де є можливість обраховувати фінальну оцінку за тести та завдання.

Моделюванні інформаційної системи проведено на основі методології UML, що використовується для моделювання архітектури великих проєктів та передбачає розробку моделей, які описують об'єкти [12]. На рис. 2.2 відображено діаграму прецедентів інформаційної системи вивчення основ штучного інтелекту, що описує типовий сценарій використання системи і взаємодії між користувачами та системою. На цій діаграмі відображено акторів (що взаємодіє з системою, але не належить до неї) та прецедентів (дія системи). Акторами є:

- учень: людина, яка хоче навчатися нейромережам;
- інформаційна система: програмне забезпечення, яке використовується для навчання;
- вчитель: людина, яка навчає інших нейромережам;
- адміністратор: людина, яка керує системою.

Іншим елементом діаграми прецедентів є безпосередньо прецеденти, які вказують на активності у межах інформаційної системи. У проєктованій системі вивчення основ ШІ передбачено такі прецеденти:

- 1) учні можуть:
  - проходити початкові тести, щоб оцінити свої знання з нейромереж;
  - перевіряти теоретичні знання за допомогою тестів;
  - вирішувати завдання з нейромереж, отримуючи оцінку своїх навичок;
  - розв'язувати практичні задачі, застосовуючи теоретичні знання та навички на реальних прикладах;

- досліджувати неймережі, вивчаючи їх архітектуру, параметри та результати навчання;

- отримувати сертифікат про успішне проходження курсу.

2) вчителі можуть:

- досліджувати неймережі так само, як і учні;

- додавати курси до системи, створюючи нові навчальні матеріали;

- переглядати контент, доступний в системі.

3) адміністратори можуть:

- переглядати контент, доступний в системі;

- оновлювати контент, додаючи нові матеріали та редагуючи вже існуючі.

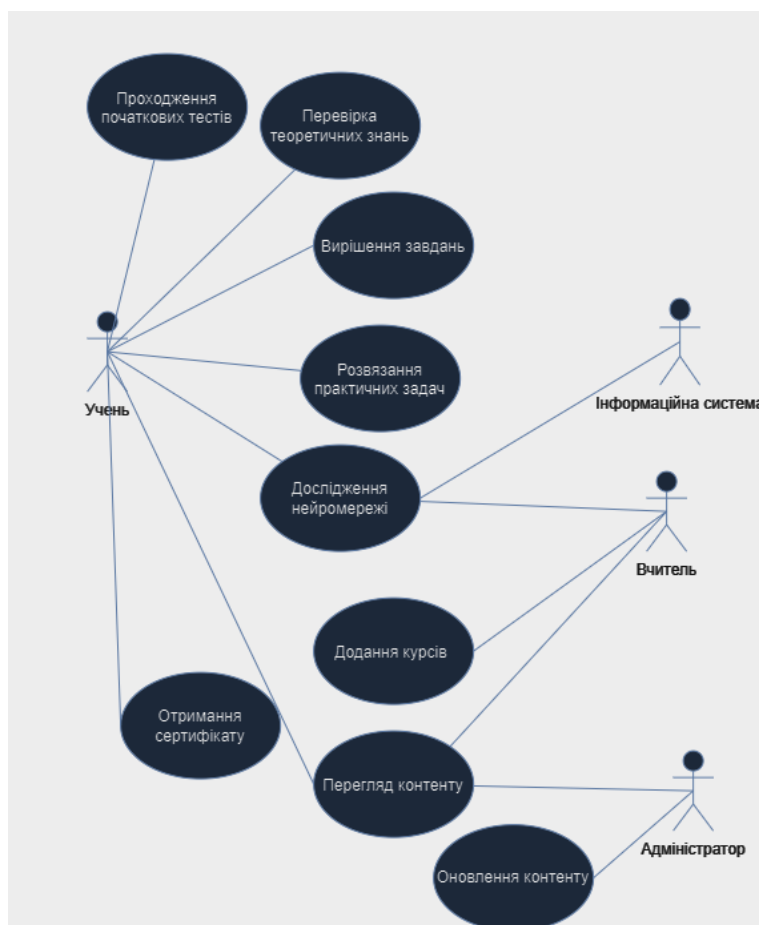


Рисунок 2.2 – UML-діаграма прецедентів інформаційної системи вивчення основ штучного інтелекту



UML-діаграма класів (рис. 2.3) відображає структуру системи шляхом моделювання її класів, операцій і зв'язків між об'єктами [12]. Верхній набір блоків – це діаграма класів системи, що вчиться запам'ятовувати зображення, які показує користувач. Нижні чотири блоки – система, яка дозволяє користувачеві захоплювати зображення з веб-камери і класифікувати їх в одну з трьох категорій.

В класі **DCanvas** присутні поля атрибутів:

- canvas: об'єкт HTML для малювання;
- ctx: контекст малювання;
- pixel: розмір пікселя;
- is\_mouse\_down: прапорець, що показує, чи натиснута кнопка миші.

Методи цього класа наступні:

- drawLine(x1, y1, x2, y2, color): малює лінію на canvas;
- drawCell(x, y, w, h): малює комірку (квадрат) на canvas з верхнім лівим кутом у точці (x, y) і розміром (w, h);
- clear(): очищує canvas;
- drawGrid(): малює сітку на canvas;
- calculate(): обчислює або обробляє.

Клас **NeuralNetwork**, в якому є атрибути:

- train\_data: дані для навчання нейромережі;
- net: об'єкт нейромережі.

Та методи:

- train(data): навчає нейромережу на даних data;
- run(input): запускає нейромережу з вхідними даними input і повертає результат.

Клас **Main** містить в собі атрибути:

- d: об'єкт типу DCanvas (інстанція класу DCanvas);
- net: об'єкт типу NeuralNetwork (інстанція класу NeuralNetwork).

Та методи:

- `toLowerCase()`: використовується для перевірки, чи відповідає подія натискання клавіші клавішам «с», «v» або «b»;
- `addEventListener()`: використовується для додавання слухача подій до елемента.

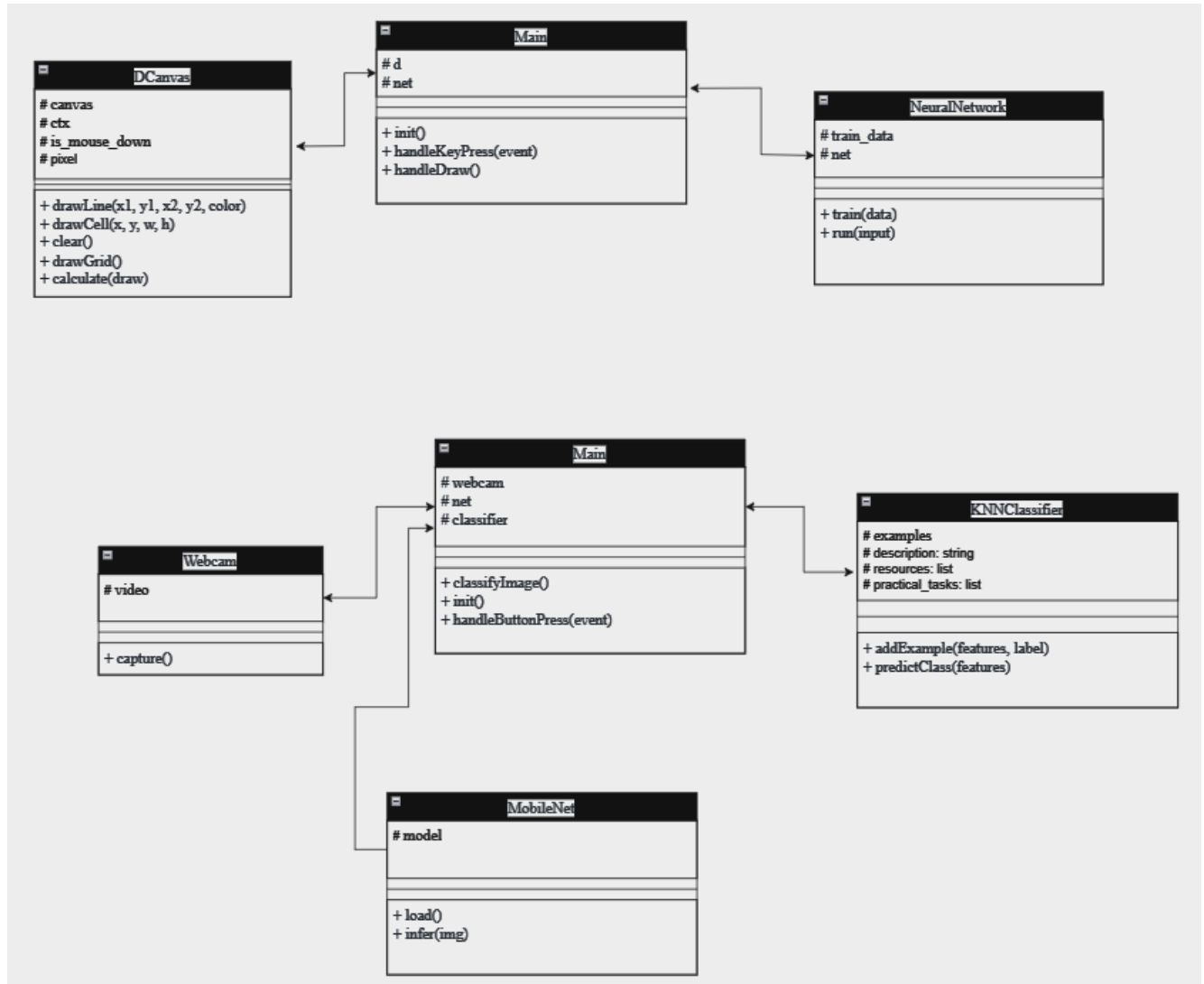


Рисунок 2.3 – UML-діаграма класів двох практичних завдань інформаційної системи вивчення основ штучного інтелекту

#### Клас **Webcam**:

- Атрибути:
  - `video`: об'єкт, що представляє відеопотік від вебкамери.
- Методи:
  - `capture()`: захоплює кадр з вебкамери і відображає його на `canvas`.

**Клас MobileNet:**

- Атрибути:
  - model: об'єкт моделі MobileNet для класифікації зображень.
- Методи:
  - load(): завантажує модель MobileNet для використання;
  - infer(img): класифікує зображення img за допомогою завантаженої

моделі MobileNet.

**Клас KNNClassifier:**

- Атрибути:
  - examples: масив прикладів для навчання класифікатора k-найближчих сусідів.
- Методи:
  - addExample(features, label): додає приклад з ознаками features і міткою `label` до навчального набору;
  - predictClass(features): прогнозує клас для нових ознак features за допомогою класифікатора k-найближчих сусідів.

**Клас Main:**

- Атрибути:
  - webcam: об'єкт типу Webcam (інстанція класу Webcam);
  - net: об'єкт типу MobileNet (інстанція класу MobileNet);
  - classifier: об'єкт типу KNNClassifier (інстанція класу KNNClassifier);
- Методи:
  - app(): ініціалізує модель MobileNet та вебкамеру.

UML-діаграма послідовностей відображає взаємодію об'єктів впорядкованих за часом. На рис. 2.4, представлена діаграма послідовності реалізації першого практичного завдання з розпізнавання малюнків, які створює користувач. Учасниками діаграми послідовностей з розпізнаванням малюнків є користувач, модуль «Практичне завдання» та нейронна мережа.

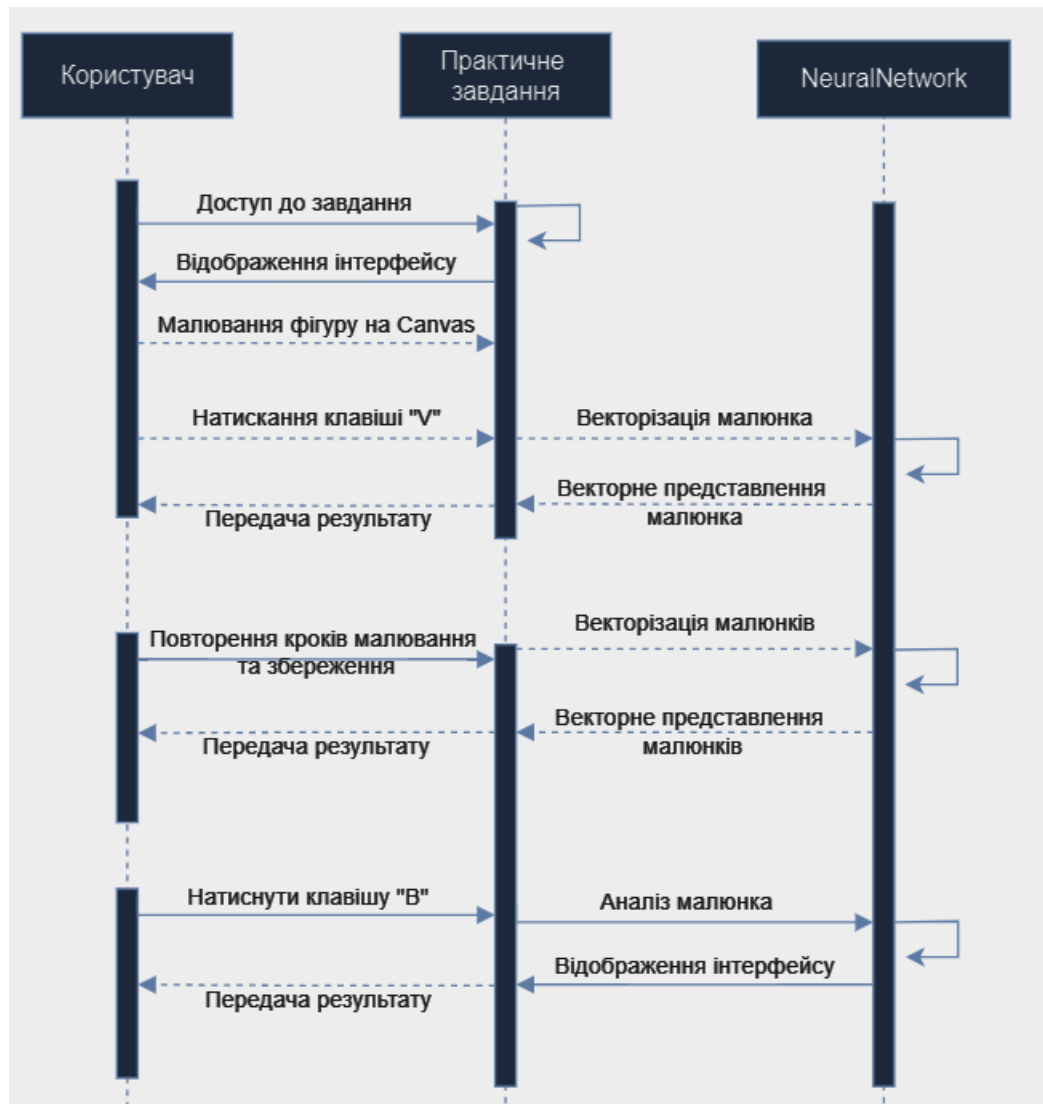


Рисунок 2.4 – UML-діаграма послідовностей ралізації практичного завдання з розпізнання малюнків

Здобувач освіти отримує доступ до практичного завдання, що представляє собою, веб-сторінку, на якій є поле для малювання об'єктів та інструкція. Користувач може малювати фігуру, та натискаючи клавішу «v», після чого веб-сторінка звертається з запитом до нейронної мережі з векторизувати малюнок. Після векторизації, результат передається на веб-сторінку і користувачу. Після цього користувач декілька разів повторює малювання фігури та її запис. Коли учень хоче перевірити, наскільки точно нейромережа може оприділити намальований рисунок, необхідно натиснути англійську клавішу «b», що призведе до

аналізу малюнка нейромережею. Результат надається веб-сторінці, яка в свою чергу передає результат користувачу.

## **2.2 Математичне забезпечення інформаційної системи вивчення основ штучного інтелекту**

Успішне вивчення основ штучного інтелекту є ключовим фактором для його ефективного застосування в різних сферах господарської діяльності. Для оцінки рівня набутих знань і навичок учнів необхідно використовувати чіткі та об'єктивні методи оцінювання. Розробка курсів, які будуть демонструвати сутність роботи штучного інтелекту, необхідна, насамперед, для двох категорій людей, а саме:

- користувачі без попереднього досвіду роботи з нейронними мережами. Ця група може не володіти базовими знаннями про ШІ або вважати цю сферу складною та неактуальною. Однак, їм може бути цікавий процес здобуття нових знань, якщо їм буде запропоновано доступний та захоплюючий курс;
- люди які вже володіють навичками роботи з технологіями штучного інтелекту і їм може бути потрібно оновити власні знання, ознайомитися з новими розробками або поглибити розуміння фундаментальних основ ШІ.

Навчальні курси зі ШІ повинні бути розроблені в такий спосіб, щоб зацікавити та залучити обидві групи. Такий підхід дозволить випускникам курсів оволодіти широким спектром знань та навичок, що може бути цінним для роботодавців. Початківці можуть отримати базові знання та навички, необхідні для початку роботи у сфері ШІ, а досвідчені фахівці – поглибити знання та отримати нові навички, що дозволить їм зайняти більш престижні та високооплачувані посади. Крім того, охоплення обох груп дозволить збільшити кількість кваліфікованих фахівців у сфері ШІ, що сприятиме розвитку цієї галузі, створенню нових інноваційних продуктів і послуг та розвитку країни цілому.

Як зазначалось у п. 1.2, одним із функціональних блоків інформаційних систем з освітньою складовою є оцінювання успішності користувачів, яка

представлена як «Розрахунок фінальної оцінки», що передбачає визначення загального рівня набутих ними знань і навичок та уможлиблює застосування знань зі ШІ для вирішення практичних завдань, розвиток навичок роботи з інструментами та програмним забезпеченням ШІ. Висока успішність у навчанні вимагає від учнів розв'язування складних задач, що, в тому числі, сприятиме розвитку їх вмінь вирішувати складні та нетривіальні задачі.

Проте, оцінювання успішності важлива не лише для здобувача освіти, а й для адміністраторів та розробників веб-платформи. Кількісне вимірювання успішності навчання дозволяє оцінити якість контенту загалом: якщо учні справляються з поставленими завданнями, то можна зробити висновок про високу якість контенту, а це в свою чергу, покращує репутацію в освітньому середовищі серед інших навчальних ресурсів.

У дослідженні формалізовано процес оцінювання успішності учнів. Припустимо, що кожен з тестів та завдань має максимальну оцінку 100 балів. Остаточна оцінка ( $Y$ ) може бути обчислена як середнє з усіх оцінок, з урахуванням їх ваги ( $w$ ). Ваговий коефіцієнт визначає важливість кожного компонента в загальній оцінці. Остаточна оцінка визначається за формулою:

$$Y = \frac{\sum_{i=1}^n w_i \cdot x_i}{\sum_{i=1}^n w_i}, \quad (2.1)$$

де  $Y$  – результативна оцінка;  $n$  – кількість тестів та завдань;  $w_i$  – вага  $i$ -го тесту та завдання;  $x_i$  – оцінка за  $i$ -й тест або завдання.

У рамках процесу вивчення основ ШІ пропонується виділяти три типи тестів, а саме: з інформатики, математики та тест по засвоєного матеріалу, а також три типи завдань – завдання з пазлами, завдання з рукописного тексту, завдання з нейронами. Перші два тести потрібні для попередньої перевірки володіння учнем базовими знаннями з предметів, які необхідні для того, щоб приступити до вивчення основ ШІ.

Для кожного завдання та тесту встановлюється вага – параметр визначення важливості кожного компонента. Сума вагових коефіцієнтів повинна

дорівнювати 1. На рис. 2.5 відображено функціонально модель математичного апарату забезпечення процесу оцінювання успішності навчання, що базується на виразі (1).

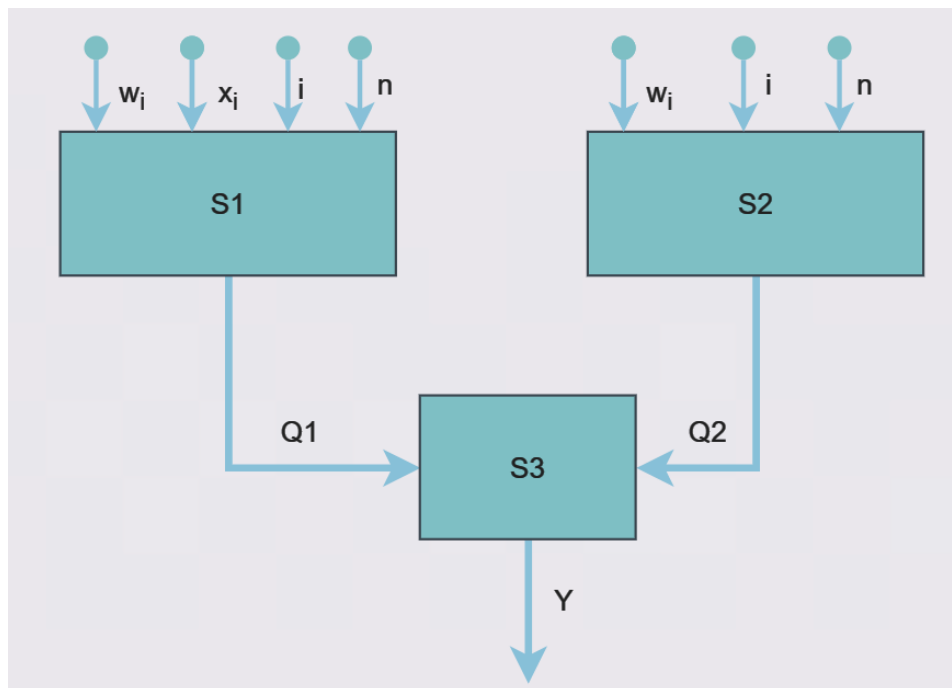


Рис. 2.5 – Схема формалізованої моделі системи оцінки успішності

Формулу (2.1) можна розписати таким чином:

$$Y = \frac{w_1 x_1 + w_2 x_2 + \dots + w_6 x_6}{w_1 + w_2 + \dots + w_6}, \quad (2.2)$$

де  $Y$  – результативна оцінка;  $w_i$  – вага тестів та завдань;  $x_i$  – оцінки за тести та завдання.

Вирази наведені вище, не враховують прогрес учня у динаміці. Тому, доцільно додати ще одну змінну (рис. 2.5):

$$Y = \frac{\sum_{i=1}^n w_i \cdot x_i \cdot e^{-\lambda t_i}}{\sum_{i=1}^n w_i}, \quad (2.3)$$

де  $Y$  – результативна оцінка;  $n$  – кількість тестів та завдань;  $w_i$  – вага  $i$ -го тесту та завдання;  $x_i$  – оцінка за  $i$ -й тест або завдання;  $t_i$  – час, що минув з моменту отримання  $i$ -ї оцінки;  $\lambda$  – параметр згасання;  $e^{-\lambda t_i}$  – експоненційний часовий

коефіцієнт, який зменшує вплив оцінки з часом (чим більше часу минуло з моменту отримання оцінки, тим менший її вплив на загальну оцінку).

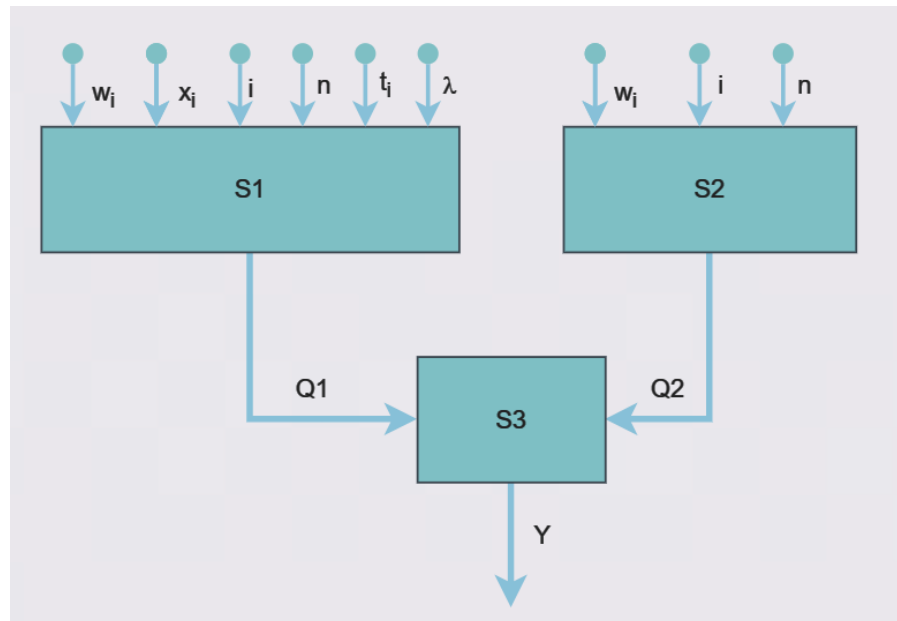


Рис. 2.6 – Схема формалізованої моделі оцінки успішності з врахуванням прогресу учня

Таким чином, розробка курсів штучного інтелекту і включення окресленого підходу для математичне забезпечення інформаційної системи вивчення основ штучного інтелекту допоможе створити ефективну навчальну систему, корисну для людей з різним освітнім рівнем, і сприятиме розвитку знань і навичок в області ШІ. Застосування чіткої та зрозумілої мови, використання практичних прикладів та завдань, а також ретельне оцінювання успішності навчання може допомогти створити ефективні курси, які сприятимуть розвитку знань та навичок у сфері ШІ.

### Висновки до другого розділу

За допомогою моделювання інформаційної системи вивчення основ штучного інтелекту було розроблено три UML-діаграми: прецедентів, класів, послідовностей. Сформовано математичне забезпечення інформаційної системи вивчення основ штучного інтелекту, в якій зазначено підхід для обрахування оцінки успішності здобувача освіти.



## Розділ 3. РЕАЛІЗАЦІЯ СИСТЕМИ ВИВЧЕННЯ ОСНОВ ШТУЧНОГО ІНТЕЛЕКТУ

### 3.1 Програмна реалізація практичних завдань вивчення основ штучного інтелекту

В інформаційній системі вивчення основ штучного інтелекту реалізовано два типи практичних завдань, а саме:

– розпізнавання малюнків – модуль, який розпізнає фігури, намальовані користувачем, і класифікує їх за двома категоріями;

– класифікація зображень – модуль, який класифікує зображення, зроблені веб-камерою, за однією з трьох категорій.

Для розроблення завдань використовувалась мова програмування JavaScript, яка застосовується для веб-розробки. JavaScript є динамічною мовою, що означає, що її можна використовувати для створення динамічних веб-сторінок, які змінюються у процесі взаємодії користувача з веб-сторінкою. Це робить його оптимальним вибором для розробки систем, які вимагають взаємодії з користувачем, зокрема, системи розпізнавання малюнків і класифікації зображень.

Перше практичне завдання з розпізнавання малюнків на основі нейронних мереж:

HTML-файл має базову структуру з розділом **<head>**, який включає в себе **<title>**, тег **<meta>** для встановлення кодування символів і тег **<script>**, який завантажує бібліотеку Brain.js. Розділ **<body>** містить елемент **<canvas>** з ідентифікатором «canv», елемент **<div>** з ідентифікатором «legend» і два теги **<script>**, які містять код JavaScript.

Елемент **<canvas>** використовується для створення області малювання, де користувач може малювати об'єкти за допомогою миші. Полотно має ширину і висоту 500 пікселів, а також білий колір фону з сірою рамкою. Текст «Оновіть браузер!!!» відображається, якщо браузер не підтримує полотно.

Елемент `<div>` з ідентифікатором «`legend`» використовується для відображення легенди або підказки, яка надає інформацію про систему. Легенда має кнопку перемикача, яка показує або приховує вміст легенди, що включає в себе список інструкцій і кнопку для згортання легенди (рис. 3.1).

```
<canvas id="canv" style="display: block; border: 10px solid #ddd; box-shadow: 0px 0px 10px 15px rgb(15, 1, 1);">Updat
<div id="legend" style="position: absolute; right: 0; top: 0; background-color: #E0FFFF; padding: 1px; border: 1px soli
  <h4 id="legend-toggle">Натисніть сюди для підказки</h4>
  <div id="legend-content" style="display: none;">
    <ul>
      <li>C - Стерти/Видалити малюнок (Clear)</li>
      <li>V - Векторизувати малюнок (Vectorize)</li>
      <li>B - Нехай навчена модель вгадає, що я намалював (Guess)</li>
    </ul>
    <p style="font-size: 12px; color: #666;">Примітка: Розкладка клавіатури/вибрана мова має бути англійською!!!</p>
    <button id="collapse-legend" style="background-color: #00FFFF;">Згорнути</button>
  </div>
</div>
```

Рисунок 3.1 – Елементи `canvas` та `div`

Тег `<script>` (рис. 3.2) визначає клас `DCanvas`, який представляє елемент `canvas`. Клас має декілька методів:

- **drawLine(x1, y1, x2, y2, color)**: малює лінію на полотні.
- **drawCell(x, y, w, h)**: малює комірку на полотні.
- **clear()**: очищає полотно.
- **drawGrid()**: малює сітку на полотні.
- **calculate(draw)**: обчислює значення пікселів полотна і повертає векторне представлення малюнка. Якщо **draw** має значення `true`, він також перемальовує полотно з обчисленими значеннями пікселів.

```

function DCanvas(el)
{
  const ctx = el.getContext('2d');
  const pixel = 20;

  let is_mouse_down = false;

  canv.width = 500;
  canv.height = 500;

  this.drawLine = function(x1, y1, x2, y2, color = 'gray') {
    ctx.beginPath();
    ctx.strokeStyle = color;
    ctx.lineJoin = 'miter';
    ctx.lineWidth = 1;
    ctx.moveTo(x1, y1);
    ctx.lineTo(x2, y2);
    ctx.stroke();
  }

  this.drawCell = function(x, y, w, h) {
    ctx.fillStyle = 'blue';
    ctx.strokeStyle = 'blue';
    ctx.lineJoin = 'miter';
    ctx.lineWidth = 1;
    ctx.rect(x, y, w, h);
    ctx.fill();
  }

  this.clear = function() {
    ctx.clearRect(0, 0, canv.width, canv.height);
  }

  this.drawGrid = function(){
    const w = canv.width;
    const h = canv.height;
    const p = w / pixel;

    const xStep = w / p;
    const yStep = h / p;
  }
}

```

Рисунок 3.2 – Елемент script з описом класу DCanvas

Клас **DCanvas** також визначає слухачів подій для елемента canvas (рис. 3.3):

- **mousedown**: встановлює прапорець, який вказує на те, що миша опущена.
- **mouseup**: встановлює прапорець, який вказує на те, що миша піднята.
- **mousemove**: малює лінію на полотні, якщо миша опущена.

```

e1.addEventListener('mousedown', function(e) {
  is_mouse_down = true;
  ctx.beginPath();
})

e1.addEventListener('mouseup', function(e) {
  is_mouse_down = false;
})

e1.addEventListener('mousemove', function(e) {
  if( is_mouse_down )
  {
    ctx.fillStyle = 'red';
    ctx.strokeStyle = 'red';
    ctx.lineWidth = pixel;

    ctx.lineTo(e.offsetX, e.offsetY);
    ctx.stroke();

    ctx.beginPath();
    ctx.arc(e.offsetX, e.offsetY, pixel / 2, 0, Math.PI * 2);
    ctx.fill();

    ctx.beginPath();
    ctx.moveTo(e.offsetX, e.offsetY);
  }
})

```

Рисунок 3.3 – Елемент script з слухачами подій

Наступним елементом тега `<script>` є визначення основної логіки роботи системи: він створює екземпляр класу **DCanvas** і присвоює його змінній `d`. Також він визначає слухача події натискання клавіш, який прослуховує наступні англійські клавіші:

- **c**: очищає полотно;
- **v**: обчислює векторне представлення малюнка і пропонує користувачеві зберегти його на одну з двох кнопок;
- **b**: навчає нейронну мережу на основі збережених даних і робить прогноз на основі поточного малюнка.

Коли користувач натискає клавішу `v`, створюється спеціальне діалогове вікно, в якому користувачеві пропонується зберегти малюнок на одну з двох кнопок. Коли користувач натискає на одну з кнопок, малюнок зберігається у відповідному виході.

Коли користувач натискає клавішу **b**, нейронна мережа навчається на основі збережених даних, і робиться прогноз на основі поточного малюнка. Результат відображається у спливаючому діалоговому вікні разом з точністю прогнозу (рис. 3.4).

```

    if (e.key.toLowerCase() == 'b') { //У6р**
net = new brain.NeuralNetwork();
net.train(train_data, { log: true });

const userVector = d.calculate();
const result = brain.likely(userVector, net);

// Порівняти розпізнаний результат із навчальними даними та обчислити точність
const accuracy = net.run(userVector);

// Поп-ап для відображення результату і точності
const dialog = document.createElement('div');
dialog.innerHTML = `
    <div style="position: fixed; top: 50%; left: 50%; transform: translate(-50%, -50%);
      <h3>Результат:</h3>
      <p>Модель передбачає, що це: ${result}</p>
      <p>Точність: ${accuracy[result]}</p>
      <button id="close-button" style="background-color: red; color: white;">Закрити</bu
    </div>
`;
document.body.appendChild(dialog);

// Обробник подій для кнопки закриття
document.getElementById('close-button').addEventListener('click', function() {
    document.body.removeChild(dialog);
});

```

Рисунок і3.4 – Логіка при натисканні англійської клавіші «b»

Нейронна мережа в цьому коді створена за допомогою конструктора **brain.NeuralNetwork()**, бібліотеки Brain.js, який створює нову нейронну мережу з налаштуваннями за замовчуванням. Нейронна мережа має один прихований шар з 10 нейронів і використовує сигмоїдну функцію активації.

Бібліотека Brain.js використовується для створення та навчання нейронної мережі. Бібліотека надає простий API для створення нейронних мереж та їх навчання на основі даних. У цьому коді нейронна мережа навчається на основі збережених даних і робить прогнози на основі вхідних даних. Функція **brain.likely()** використовується для обчислення ймовірності належності вхідних

даних до кожної з вихідних категорій. Функція **brain.run()** використовується для того, щоб зробити прогноз на основі вхідних даних.

Нейронна мережа навчається за допомогою методу **net.train()**, який отримує на вхід масив навчальних даних. Навчальні дані – це масив об'єктів, де кожен об'єкт має вхідну та вихідну властивості. Вхідна властивість – це масив значень пікселів, які представляють малюнок користувача, а вихідна властивість – це об'єкт, який представляє категорію малюнка.

Наприклад, якщо користувач намалює коло і збереже його на першій кнопці, навчальні дані можуть виглядати так:

```
[{
  вхідні дані: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...],
  вихід: {позитивне: 1}
}]
```

У цьому прикладі вхідною властивістю є масив значень пікселів, які представляють коло, а вихідною властивістю є об'єкт, який вказує на належність малюнка до категорії «позитивних».

Метод **net.train()** використовує алгоритм зворотного розповсюдження для налаштування ваг та зсувів нейронної мережі на основі навчальних даних. Алгоритм ітеративно підлаштовує ваги та зсуви, щоб мінімізувати різницю між прогнозованим та фактичним виходом.

Після того, як нейронна мережа навчена, її можна використовувати для прогнозування на основі нових вхідних даних. Функція **brain.likely()** використовується для обчислення ймовірності належності вхідних даних до кожної з вихідних категорій. Функція **brain.run()** використовується для того, щоб зробити прогноз на основі вхідних даних. Наприклад, якщо користувач намалює нове коло і натисне клавішу **b**, функція **brain.likely()** повертає такий результат:

```
{
  позитивних: 0,95,
  негативні: 0,05
}
```

У цьому прикладі позитивна властивість має значення 0.95, що означає, що нейромережа на 95% впевнена, що малюнок належить до категорії «Це перший малюнок». Негативна властивість має значення 0,05, що означає, що нейромережа лише на 5% впевнена, що малюнок належить до категорії «Це другий малюнок».

Функція **brain.run()** може бути використана для створення прогнозу на основі вхідних даних. Функція повертає об'єкт, який містить категорію прогнозованого результату та відповідний рівень достовірності.

Наприклад, якщо користувач намалює нове коло і натисне клавішу **b**, функція **brain.run()** може повернути наступний результат:

```
{  
  позитивні: 0,95  
}
```

У цьому прикладі позитивна властивість має значення 0.95, що означає, що нейромережа на 95% впевнена, що малюнок належить до категорії «Це перший малюнок».

Друге практичне завдання класифікує зображення, зроблені веб-камерою, за однією з трьох категорій.

Код другого завдання використовує TensorFlow.js та модель MobileNet для створення простої системи класифікації зображень. Система дозволяє захоплювати зображення з веб-камери та класифікувати їх у одну з трьох категорій. HTML-файл містить елемент **<video>**, який відображає відеопотік з веб-камери, та три кнопки, які дозволяють користувачу захоплювати зображення та класифікувати їх (рис.3.5). Файл JavaScript містить основну логіку системи.

```
<div class="work_section">  
  <video  
    autoplay  
    playsinline  
    muted  
    id="webcam"  
    width="1000px"  
    height="400px"  
  ></video>  
</div>
```

Рисунок 3.5 – Елемент **<video>**

Функція **app()** є головною точкою входу в систему. Вона завантажує модель MobileNet та створює об'єкт веб-камери, який захоплює зображення з веб-камери користувача. Вона також створює класифікатор KNN, який буде використовуватися для класифікації зображень (рис.3.6).

Функція **addExample()** викликається, коли користувач натискає одну з кнопок. Вона захоплює зображення з веб-камери, витягує активаційні функції з моделі MobileNet та додає приклад до класифікатора KNN (рис.3.6).

Цикл **while** безперервно захоплює зображення з веб-камери та класифікує їх за допомогою класифікатора KNN. Результат класифікації відображається в блоку **console**.

```

async function app() {
  console.log("Loading mobilnet...")

  net = await mobilenet.load()

  console.log("Loaded model")

  const webcam = await tf.data.webcam(webcamElement)

  const addExample = async (classId) => {
    const img = await webcam.capture()

    const activation = net.infer(img, true)

    classifier.addExample(activation, classId)

    img.dispose()
  }
}

```

Рисунок 3.6 – Функції **app()** та **addExample()**

Функція **predictClass()** використовується для класифікації зображення. Вона приймає активаційні ознаки як вхідні дані і повертає об'єкт, який містить передбачену мітку класу та відповідний рівень впевненості (рис.3.7).

Функція **getNumClasses()** використовується для перевірки наявності прикладів у класифікаторі KNN. Якщо прикладів немає, система не буде намагатися класифікувати зображення (рис.3.7).

Функція **dispose()** використовується для звільнення пам'яті шляхом видалення об'єктів тензорів (рис.3.7).



```

while (true) {
  if (classifier.getNumClasses() > 0) {
    const img = await webcam.capture()

    const activation = net.infer(img, "conv_preds")

    const result = await classifier.predictClass(activation)

    const classes = ["Це перша картинка", "Це друга картинка", "Це третя картинка"]

    document.getElementById("console").innerText = `
      Це: ${classes[result.label]}\n
      Вірогідність: ${result.confidences[result.label]}
    `

    img.dispose()
  }
}

```

Рисунок 3.7 – Функції `getNumClasses()`, `predictClass()` та `dispose()`

У цьому коді частина нейронної мережі базується на моделі MobileNet, яка є попередньо навченою згортковою нейронною мережею (CNN), що використовується для класифікації зображень. Модель MobileNet завантажується за допомогою функції `mobilenet.load()`, яка повертає об'єкт моделі TensorFlow.js.

Модель MobileNet використовується для вилучення ознак з вхідних зображень. Функція `infer()` використовується для проходження вхідного зображення через модель MobileNet та вилучення активаційних ознак зі згорткових шарів. Отримані активаційні ознаки використовуються як вхідні дані для класифікатора KNN. Класифікатор KNN – це простий класифікатор, який використовує алгоритм k-найближчих сусідів для класифікації вхідних зображень. Функція `knnClassifier.create()` створює новий об'єкт класифікатора KNN, а функція `addExample()` додає новий приклад до класифікатора. Функція `predictClass()` використовується для класифікації нового вхідного зображення за допомогою класифікатора KNN.

Нижче наведено загальний огляд того, як працює частина з нейронною мережею:

- Вхідне зображення захоплюється з веб-камери за допомогою функції `tf.data.webcam()`.

- Вхідне зображення проходить через модель MobileNet за допомогою функції **infer()**, яка витягує активаційні ознаки зі згорткових шарів.
- Активаційні ознаки потім передаються класифікатору KNN за допомогою функції **addExample()**.
- Класифікатор KNN використовує активаційні ознаки для класифікації вхідного зображення за допомогою алгоритму k-найближчих сусідів.
- Результат класифікації потім відображається в div-елементі консолі за допомогою функції **predictClass()**.

Нейромережева частина коду використовується для вилучення ознак з вхідних зображень та їх класифікації за допомогою класифікатора KNN. Модель MobileNet використовується як екстрактор ознак, а класифікатор KNN використовується для прийняття остаточного класифікаційного рішення.

### 3.2 Реалізація математичної моделі оцінювання успішності вивчення основ штучного інтелекту

Математичну модель обчислення узагальненої оцінки успішності учні щодо вивчення основ штучного інтелекту наведену у п. 2.2 реалізовано на базі JavaScript у формі окремої веб-сторінки. Зазначена модель враховує ваги оцінок та час, витрачений на виконання завдання для обчислення остаточної результату. Вираз (2.3), що використовується для відповідних розрахунків, трансформовано для подальшої програмної реалізації:

$$\text{finalScore} = (\sum(\text{weights}[i] * \text{scores}[i] * \exp(-\text{decay} * \text{times}[i])) / \text{totalWeight},$$

де:

- $\text{weights}[i]$  – вага  $i$ -ої оцінки;
- $\text{scores}[i]$  –  $i$ -та оцінка;
- $\text{times}[i]$  – період проходження, пов'язаний з  $i$ -ою оцінкою;
- $\text{decay}$  – константа (встановлена на 0.1), яка представляє швидкість занепаду оцінки з часом;
- $\text{totalWeight}$  – сума всіх ваг.

Для її реалізації моделі оцінювання успішності процесу вивчення основ штучного інтелекту використано мови HTML, CSS та JavaScript. Калькулятор враховує бали, вагу та час, необхідний для виконання завдання, пов'язаний з кожним балом. HTML-код створює таблицю з полями для введення балів, ваг та часу. Кожен рядок представляє один бал. В коді HTML можна зазначити декілька функцій: функція **calculateScore** ітераційно переглядає поля введення, витягує значення і обчислює остаточний бал за формулою, вона також перевіряє, чи загальна вага не перевищує 1, і попереджає користувача, якщо це так; функція **updateMissing** викликається щоразу, коли змінюється поле введення ваги, і оновлює відображення відсутньої ваги.

А код JavaScript, в свою чергу, визначає кілька функцій:

- **incrementScore** та **decrementScore** збільшують або зменшують значення поля введення балів або ваги.
- **adjustValue** гарантує, що вхідне значення знаходиться в допустимому діапазоні (наприклад, бали від 1 до 100, ваги від 0 до 1).
- **calculateScore** обчислює остаточний результат за формулою вище.
- **updateMissing** обчислює загальну вагу і виводить на екран відсутню вагу (тобто, 1 - загальна вага).

Формула обчислює підсумковий бал шляхом підсумовування добутків кожного балу, ваги та експоненціального коефіцієнта розпаду. Коефіцієнт розпаду використовується для зменшення впливу попередніх оцінок на підсумковий бал.

Покрокова розбивка формули має вигляд:

- для кожної оцінки обчислити добуток оцінки, ваги та експоненціального коефіцієнта розпаду:  $weights[i] * scores[i] * \exp(-decay * times[i])$ ;
- підсумок цих добутків за всіма оцінками:  $\sum(weights[i] * scores[i] * \exp(-decay * times[i]))$ ;
- ділення суми на загальну вагу:  $(\sum(weights[i] * scores[i] * \exp(-decay * times[i]))) / totalWeight$ .

На рисунку 3.8 показано інтерфейс веб-сторінки для обчислення остаточної оцінки успішності вивчення основ штучного інтелекту. Інтерфейс містить таблицю з полями для введення оцінок, ваг та час, необхідний для виконання завдання, а також кнопку «Обчислити остаточно оцінку» для розрахунку підсумкової оцінки за математичною моделлю. У таблиці присутні кнопки «+» та «-» для збільшення або зменшення відповідних значень оцінок. Внизу інтерфейсу відображається остаточно розрахована оцінка.

### Оцінки за тести та завдання

#	Оцінка	Вага	Період проходження	
1	<input type="text" value="85"/>	<input type="text" value="0,1"/>	<input type="text" value="0"/>	<input type="button" value="+"/> <input type="button" value="-"/>
2	<input type="text" value="90"/>	<input type="text" value="0,25"/>	<input type="text" value="0"/>	<input type="button" value="+"/> <input type="button" value="-"/>
3	<input type="text" value="80"/>	<input type="text" value="0,15"/>	<input type="text" value="0"/>	<input type="button" value="+"/> <input type="button" value="-"/>
4	<input type="text" value="95"/>	<input type="text" value="0,15"/>	<input type="text" value="0"/>	<input type="button" value="+"/> <input type="button" value="-"/>
5	<input type="text" value="100"/>	<input type="text" value="0,20"/>	<input type="text" value="0"/>	<input type="button" value="+"/> <input type="button" value="-"/>
6	<input type="text" value="90"/>	<input type="text" value="0,15"/>	<input type="text" value="0"/>	<input type="button" value="+"/> <input type="button" value="-"/>

Остаточна оцінка: 90.75

Рисунок 3.8 – Математична модель оцінювання успішності вивчення основ штучного інтелекту

Представлена веб-сторінка з обчисленням оцінки успішності вивчення основ штучного інтелекту пропонує зручний інструмент для оцінювання знань та навичок учнів. Інтерфейс веб-сторінки простий та зрозумілий, що робить його

доступним для широкого кола користувачів. Цей калькулятор може бути корисним як для викладачів, які прагнуть оцінювати знання своїх учнів, так і для самих учнів, які хочуть відстежувати свій прогрес в навчанні основам штучного інтелекту та виявляти свої недоліки, що потребують вдосконалення.

### 3.3 Інтерфейс інформаційної системи вивчення основ штучного інтелекту

Розроблена інформаційна система складається з двох окремих модулів: розпізнавання малюнків та класифікації зображень.

Веб-сторінка модуля з розпізнавання малюнків (рис. 3.9) містить поле, на якому користувач може намалювати об'єкт за допомогою миші. Після векторизації малюнка (кнопка «v») можна зберегти малюнок («Зберегти в першу кнопку»/»Зберегти в другу кнопку»). Після обрання, в яку кнопку необхідно зберегти малюнок, об'єкт зберігається у відповідну категорію для подальшого навчання нейронної мережі. При натисканні кнопки «b» система аналізує поточний малюнок та відображає передбачену категорію у спливаючому вікні.

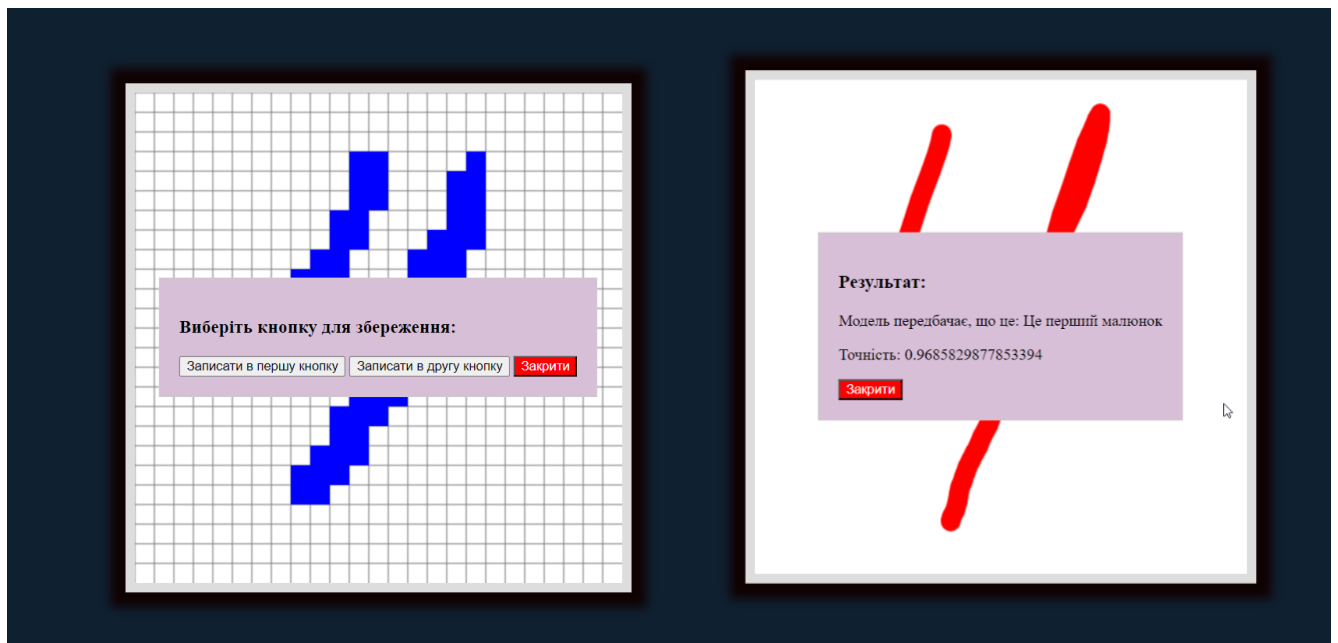


Рисунок 3.9 – Веб-сторінка модуля інформаційної системи з розпізнавання малюнків

Друга частина системи призначена для модуля класифікації зображень з веб-камери. На веб-сторінці (рис. 3.10) відображається потік з камери, під яким розміщені три кнопки для захоплення поточного кадру та його класифікації в одну з трьох категорій.

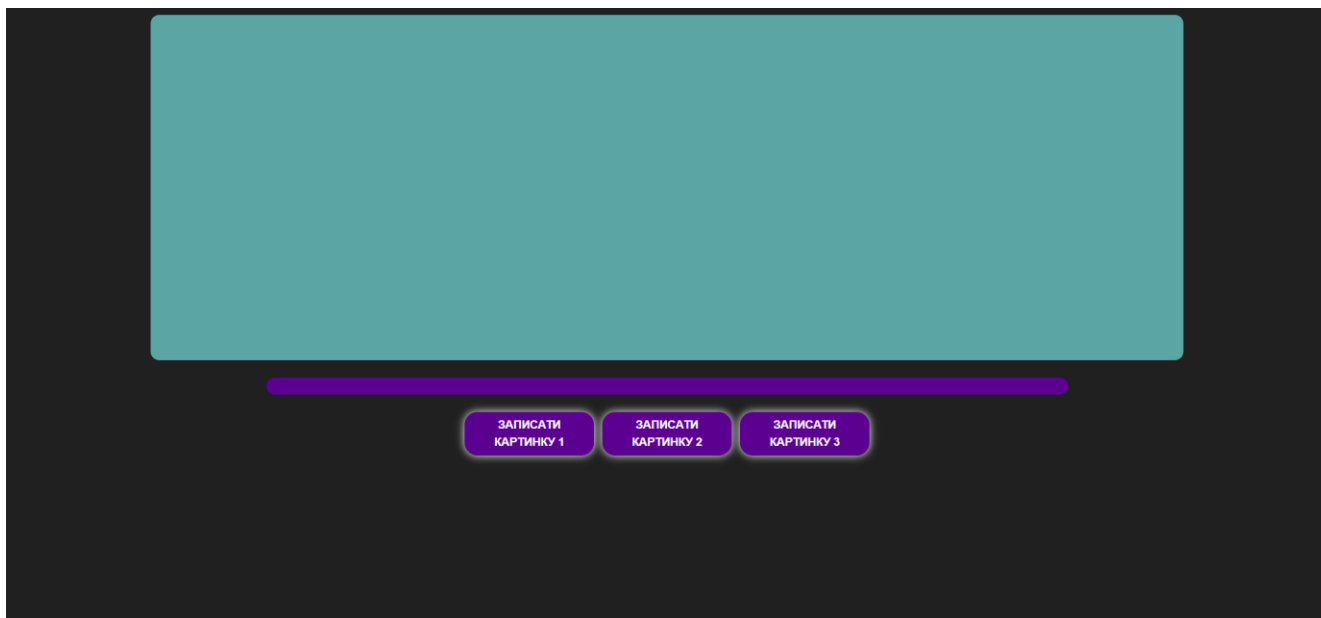


Рисунок 3.10 – Веб-сторінка модуля інформаційної системи з класифікації зображень

Після натискання кнопки система захоплює поточний кадр з камери, класифікує об'єкти на ньому та відображає результат у нижній частині інтерфейсу веб-сторінки (рис.3.11).

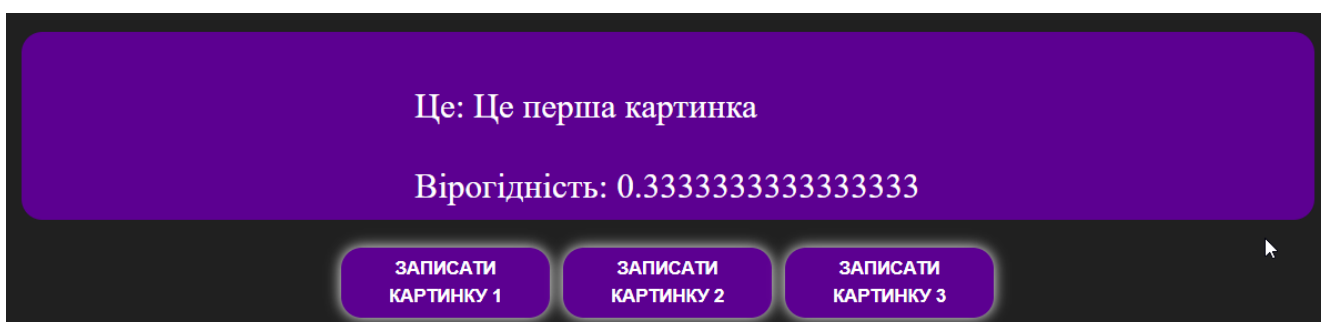


Рисунок 3.11 – Нижня частина інтерфейсу модуля інформаційної системи з класифікації зображень

Інтерфейс інформаційної системи вивчення основ штучного інтелекту має інтуїтивний та зручний графічний інтерфейс користувача, який дозволяє легко виконувати практичне завдання, інтерфейс є зрозумілим та мінімалістичним. Всі необхідні елементи керування чітко розміщені та мають зрозумілі назви.

### **3.4 Керівництво роботи з інформаційною системою користувачу**

Розроблено керівництво для користувача інформаційної системи вивчення основ штучного інтелекту яке включає покрокову інструкцію:

Розпізнавання малюнків:

- відкрийте HTML-файл у веб-браузері;
- намалюйте фігуру на полотні за допомогою миші;
- натисніть клавішу «v», щоб векторизувати малюнок, і збережіть його на одній з двох кнопок;
- повторіть кроки 2-3 рази, щоб зберегти кілька малюнків на кнопках;
- намалюйте нову фігуру на полотні і натисніть клавішу «v», щоб побачити результат розпізнавання.

Класифікація зображень:

- відкрийте HTML-файл у веб-браузері;
- дозвольте доступ до веб-камери, коли з'явиться відповідний запит;
- зніміть зображення з веб-камери, натиснувши на одну з трьох кнопок;
- повторіть крок 3 рази, щоб захопити кілька зображень і додати їх до класифікатора;
- система буде безперервно захоплювати зображення з веб-камери і класифікувати їх за допомогою класифікатора KNN;
- під час наведення об'єкта на камеру відобразатиметься номер малюнка, куди було записано об'єкт.

### **Висновки до третього розділу**

В третьому розділі було реалізовано декілька практичних завдань з вивчення основ штучного інтелекту. Показано код та його опис. Реалізовано математичну модель оцінювання успішності процесу вивчення основ штучного інтелекту та розроблено керівництво користувачу, для роботи з інформаційною системою.



## ВИСНОВКИ

Технології штучного інтелекту стрімко розвиваються і стають невід'ємною частиною сучасного світу. Тому важливо забезпечити доступ до навчання основам роботи зі штучним інтелектом для різних категорій людей, зробивши цей процес цікавим та зрозумілим. Це дасть можливість імплементувати відповідні технології з виокремленим потенціалом у максимальну велику кількість галузей економіки країни.

У кваліфікаційній роботі проаналізовано та змодельовано предметну область дослідження – вивчення основ штучного інтелекту, зокрема, побудовано її IDEF0 та IDEF3-моделі. Здійснено проектування інформаційної системи вивчення основ штучного інтелекту та побудовано відповідні UML-діаграми класів, прецедентів та послідовності. Розроблено математичну модель оцінювання успішності користувачів системи. Крім того, реалізовано модулі інформаційної системи щодо виконання практичних завдань щодо розпізнавання малюнків і класифікації зображень на основі нейронних мереж. Особливістю системи є гейміфікована подача матеріалу у вигляді інтерактивних завдань та покрокове вивчення основ штучного інтелекту. Розроблено керівництво користувачу системи.

У подальшому перспективним є розширення теоретичного матеріалу та різноманітності практичних завдань з орієнтуванням на використання елементів гейміфікації. Доцільно також інтегрувати систему з електронними курсами та навчальними платформами для ширшого охоплення аудиторії.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Штучний інтелект (ШІ) – що це таке, як працює і навіщо потрібен.  
URL: <https://termin.in.ua/shtuchnyy-intelekt/> (дата звернення: 23.04.2024)
2. Штучний інтелект. Простий путівник, який допоможе зрозуміти ШІ.  
URL: <https://www.bbc.com/ukrainian/resources/idt-74697280-e684-43c5-a782-29e9d11fecf3/> (дата звернення: 23.04.2024)
3. Всеукраїнське дослідження використання ШІ у шкільній освіті. URL: <https://mon.gov.ua/storage/app/media/news/2023/12/20/Vseukrainske.doslidzhennya.v.ukorystannya.20.12.2023.pdf> (дата звернення: 23.04.2024)
4. 50 найперспективніших компаній, які будують бізнес на основі штучного інтелекту. URL: <https://forbes.ua/innovations/spisok-nayperspektivnishikh-privatnikh-kompaniy-yaki-buduuyut-biznes-na-osnovi-shtuchnogo-intelektu-20042023-13200/> (дата звернення: 23.04.2024)
5. Година коду. Творчість з ШІ. URL: <https://hourofcode.com/ua/uk/ai/> (дата звернення: 24.04.2024)
6. Штучний інтелект. URL: <https://osvita.diia.gov.ua/courses/artificial-intelligence/> (дата звернення: 24.04.2024)
7. ОНЛАЙН-КУРС. Основи AI. URL: [https://rsvp.withgoogle.com/events/ai-basics\\_2023\\_short/](https://rsvp.withgoogle.com/events/ai-basics_2023_short/) (дата звернення: 24.04.2024)
8. ОСНОВИ ШТУЧНОГО ІНТЕЛЕКТУ (ШІ). URL: <https://career.softserveinc.com/uk-ua/technology/course/ai-fundamentals> (дата звернення: 24.04.2024)
9. Лекція 6. Нотація IDEF0. URL: [https://elib.lntu.edu.ua/sites/default/files/elib\\_upload/Кондіус%20%20готовва/page9.html](https://elib.lntu.edu.ua/sites/default/files/elib_upload/Кондіус%20%20готовва/page9.html) (дата звернення: 25.04.2024)
10. Лекція 8. Нотація IDEF3. URL: [https://elib.lntu.edu.ua/sites/default/files/elib\\_upload/Кондіус%20%20готовва/page11.html](https://elib.lntu.edu.ua/sites/default/files/elib_upload/Кондіус%20%20готовва/page11.html) (дата звернення: 25.04.2024)

11. Як будувати UML-діаграми. URL: <https://dou.ua/forums/topic/40575/> (дата звернення: 26.04.2024)
12. Методи штучного інтелекту. URL: <https://www.educba.com/artificial-intelligence-techniques/> (дата звернення: 27.04.2024)
13. TensorFlow. URL: <https://www.tensorflow.org/> (дата звернення: 27.04.2024)
14. Перші кроки в NLP. URL: <https://dou.ua/lenta/articles/first-steps-in-nlp-tensorflow/> (дата звернення: 27.04.2024)
15. Що таке KNN (К-найближчі сусіди)? URL: <https://www.unite.ai/uk/%D1%87%D0%BE%D0%BC%D1%83-%D0%B4%D0%BE%D1%80%D1%96%D0%B2%D0%BD%D1%8E%D1%94-k-%D0%BD%D0%B0%D0%B9%D0%B1%D0%BB%D0%B8%D0%B6%D1%87%D0%B8%D1%85-%D1%81%D1%83%D1%81%D1%96%D0%B4%D1%96%D0%B2/> (дата звернення: 27.04.2024)
16. MobileNet. URL: <https://medium.com/@nocodingai/mobilenet-fc34af9f58a5> (дата звернення: 27.04.2024)
17. Brain.js. URL: [https://www.w3schools.com/ai/ai\\_brainjs.asp](https://www.w3schools.com/ai/ai_brainjs.asp) (дата звернення: 28.04.2024)
18. Використання Brain.js для машинного навчання URL: <https://scribblr.live/2023/04/30/Machine-Learning-in-JavaScript-using-Brainjs.html> (дата звернення: 28.04.2024)
19. Девід Крізель, рукопис. Короткий вступ до нейронних мереж, 2007. – 244 с. URL: [http://www.dkriesel.com/\\_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf](http://www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf) (дата звернення: 23.11.2023)
20. Нільс Дж. Нільссон. У пошуках штучного інтелекту: історія ідей і досягнень, жовтень 2009. – 707 с. URL: <http://ai.stanford.edu/~nilsson/QAI/qai.pdf> (дата звернення: 20.11.2023)
21. Ушакова І. О. Основи системного аналізу об'єктів та процесів комп'ютеризації: навчальний посібник. Ч. 2 Харків: Вид. ХНЕУ, 2008. – 324 с. (дата звернення: 02.11.2023)