

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Методи оцінки напряму погляду людини в
контексті LMS

(тема)

Виконав:

студент II курсу, групи СПМ-23-2
Ляпін Я.А.
(прізвище, ініціали)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва освітньої програми)

Керівник: доц. Барковська О.Ю.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав.

кафедри _____

(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Ляпіну Ярославу Антоновичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Методи оцінки напрямку погляду людини в контексті LMS _____

затверджена наказом по університету від “ 22 ” листопада 2024 р. № 1236 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії _____ 20 січня 2025 р. _____

3. Вхідні дані до роботи _____

Відеофайли: Датасет UnityEyes - 1,04 ГБ, 129285 зображень _____

Програмне забезпечення розробки: Visual Studio Code, Keras, Python _____

Апаратне забезпечення: GPU:NVIDIA GeForce RTX 3050 Ti Laptop _____

CPU: 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz _____

Камери: KS.0HD06.030, Imx290 Sensor Starlight _____

4. Перелік питань, що потрібно опрацювати у роботі _____

- визначення можливостей персоналізованого використання LMS;

- оцінка точності визначення напрямку погляду на основі існуючих методів;

- розробка моделі адаптивної системи управління навчанням на основі тривалості фокусування та напрямку погляду;

- тонке налаштування параметрів нейромережових моделей для визначення напрямку погляду;

- оцінка впливу умов зйомки на точність визначення напрямку погляду.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 17 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

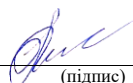
Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз можливостей персоналізованого використання LMS.	26.11.24-30.11.24	
2	Вибір та обґрунтування методики дослідження	02.12.24-05.12.24	
3	Вибір інструментальних засобів	06.12.24-10.12.24	
4	Оцінка ефективності існуючих систем	11.12.24-21.12.24	
5	Розробка моделі адаптивної системи управління навчанням	23.12.24-03.01.25	
6	Оформлення матеріалів кваліфікаційної роботи	04.01.25-07.01.25	
7	Подання кваліфікаційної роботи керівникові та її попередній захист	08.01.25-11.01.25	
8	Подання кваліфікаційної роботи на рецензування	13.01.25-17.01.25	

Дата видачі завдання 25 листопада 2024 р.

Студент


(підпис)

Керівник роботи


(підпис)

доц.каф.ЕОМ Барковська О.Ю.

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 101 с., 23 рис., 15 табл., 2 дод., 13 джерел.

СИСТЕМА, НАПРЯМ, ПОГЛЯД, LMS, МОДЕЛЬ, ВИЗНАЧЕННЯ, АДАПТАЦІЯ.

Метою кваліфікаційної роботи є дослідження та вдосконалення методів оцінки напрямку погляду людини в контексті створення адаптивної системи управління навчанням.

У ході виконання кваліфікаційної роботи був проведений ґрунтовний аналіз методів використання нейромережових моделей для оцінки напрямку погляду у контексті їх використання у якості елемента підсистеми модулю динамічної адаптації контенту.

Також було знайдено найбільш досконалу комбінацію параметрів для створення нейромережової моделей з найточнішими показниками якості серед обраних для проведення експерименту.

Отримані показники демонструють суттєві переваги над звичайними модулями CNN, що використовуються для задач аналізу зображень.

Ця розробка є актуальною, адже використання модулю оцінки напрямку погляду може використовуватися для рішення широкого кола завдань комп'ютерного зору у областях медицини, маркетингу, віртуальної та доповненої реальності.

Подальшими кроками розвитку проєкту є створення повноцінної системи управління навчанням з вбудованою системою контролю знань з більшою взаємодією з модулем оцінки напрямку погляду.

ABSTRACT

Master's thesis: 101 pages, 23 figures, 15 tables, 2 appendices, 13 sources.

SYSTEM, DIRECTION, VISION, LMS, MODEL, IDENTIFICATION, ADAPTATION.

The major goal of this thesis is to study and improve methods for assessing the direction of human gaze in the context of creating an adaptive learning management system.

During the course of the research, a comprehensive analysis of the methods of using neural network models to assess the direction of gaze in the context of their use as an element of the subsystem of the dynamic content adaptation module was carried out.

Additionally, the most optimal combination of parameters was found to create a neural network model with the most accurate quality parameters among those selected for the experiment.

The results showed significant advantages over conventional CNN modules used for image analysis tasks.

This research is relevant, as the gaze direction estimation module can be used to solve a wide range of computer vision tasks in the fields of medicine, marketing, virtual and augmented reality.

Further development of this project may include the creation of a fully-fledged learning management system with a built-in knowledge management system with greater interaction with the gaze assessment module.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Особливості побудови освітнього процесу	11
1.2 Особливості дистанційного навчання.....	13
1.3 Огляд LMS і TMS.....	17
1.3.1 Особливості та переваги Adaptive Learning System.....	22
1.4 Аналіз ролі машинного навчання в LMS та TMS.....	24
1.5 Контроль знань в навчанні	28
1.6 Мета та задачі дослідження	29
2 ВИБІР ІНСТРУМЕНТІВ І МЕТОДІВ РОЗРОБКИ.....	31
2.1 Вибір мови програмування	31
2.1.1 Python.....	32
2.1.2 Java.....	33
2.1.3 C++.....	33
2.1.4 JavaScript	34
2.1.5 Вибір мови програмування	34
2.2 Методи розпізнавання людини	35
2.2.1 Методи розпізнавання за ознаками	36
2.2.2 Методи розпізнавання за зображенням	37
2.2.3 Аналіз методів розпізнавання зображення.....	37
2.3 Вибір фреймворків та бібліотек.....	40
3 РОЗРОБКА МОДЕЛІ ТА МЕТОДІВ ОЦІНКИ НАПРЯМУ ПОГЛЯДУ	41
3.1 Опис методології проведення експериментів	43
3.1.1 Огляд інструментів для проведення експериментів.....	45

3.1.2 Огляд обраного методу виявлення обличчя і знаходження рис обличчя.....	46
3.1.3 Огляд обраного методу аналізу особливостей обличчя.....	47
3.1.4 Обґрунтування вибору датасету для проведення експериментів	47
3.2 Структура проекту	50
3.2.1 LeNet.....	52
3.2.2 AlexNet	53
3.2.3 VGGNet	54
3.2.4 ResNet.....	55
4 ПЛАНУВАННЯ ТА ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ.....	57
4.1 Затвердження базових параметрів мережі.....	57
4.2 Оцінка впливу гіперпараметрів моделі на метрики Accuracy, Loss	60
4.3 Оцінка впливу розміру батчу, оптимізатору та активаційної функції на точність детектування напряму погляду, втрати точності та час навчання	66
4.3.1 Оцінка впливу змін параметрів на VGGNet.....	66
4.3.2 Оцінка впливу змін параметрів на ResNet.....	71
4.3 Оцінка адаптації налаштованої нейромережевої моделі до зміни зовнішніх параметрів, як зміна кута нахилу голови, відстані від камери та освітлення.....	75
4.4 Впровадження моделі для використання у якості підсистеми модулю оцінки напряму погляду.....	78
ВИСНОВКИ.....	81
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	83
ДОДАТОК А.....	85
ДОДАТОК Б	95
Б.1 Файл GazeProcessor.py, що містить відповідний клас	95
Б.2 Файл main.py, що викликає програму.....	101

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

НМР – нейронні мережі на основі прийняття рішень

ННМ – нечіткі нейронні мережі

ШНМ – штучні нейронні мережі

ALS – система адаптації навчання (англ., Adaptive Learning System)

CNN – конволюційна нейронна мережа (англ., Convolutional Neural Network)

LMS – система управління навчанням (англ., Learning Management System)

TMS – система управління тренуванням (англ., Training Management System)

ВСТУП

Освіта – це невід’ємна частина існування сучасного соціуму. Освіта перетворює і спрямовує життя суспільства, зберігаючи у ньому все те, що має цінність для людини; визначає стратегію і реалістичні умови розвитку суспільства, формує нове мислення громадян, нове бачення смислу життя. У такому контексті освіта виступає як один із засобів управління розвитком суспільства. Як і суспільство, освіта повинна розвиватися, впроваджуючи у себе нові методи, ідеї та технології для підвищення якості освітнього процесу.

Загалом, система освіти являє собою консервативну структуру за власною суттю. Саме тому, не новації в освіті викликають суспільні зміни, а навпаки, зміни у суспільстві відкривають шлях новому в освіті, викликаючи необхідність її модернізації. Спрямованість у майбутнє і сучасна тенденція до зростання інтенсивності життя людини вимагають від освітніх систем перебудови і модернізації.

Підготовка кваліфікованих спеціалістів у різних галузях освіти вимагають залучення значних ресурсів, використання потужної матеріально-технічної бази та інструкторів. В той же час, враховуючи обставини у галузі освіти і науки України, підвищити якість підготовки фахівців можна шляхом вдосконалення методології вищої освіти використанням в навчальному процесі технологій дистанційного навчання.

Україна вже не перший рік вимушена приділяти багато уваги розвитку дистанційної форми освіти. Для школярів та студентів із тимчасово окупованих територій, прифронтових територій та територій ведення активних бойових дій, дистанційна освіта є найбільш ефективним інструментом забезпечення доступу до навчання. Успішність дистанційного навчання залежить від ефективної його організації, від керівництва самим процесом і майстерності викладачів, що приймають в ньому участь. Вивчення математичних дисциплін у випадку дистанційного навчання має

свою специфіку і особливості, які дозволяють значно розширити види навчальної роботи порівняно з традиційним навчанням.

Реалізація етапів розробки навчальних програм при дистанційній формі здобуття освіти досягається за допомогою використання спеціалізованих інструментів, таких як системи управління навчанням (LMS). Система управління навчанням упорядковує процес навчання: вона керує, забезпечує та відстежує різні освітні курси та навчальні програми для учнів та освітніх установ. LMS слугує центром для студентів та викладачів, оскільки надає корисні інструменти для створення інтерактивного контенту та адміністрування оцінювання, а також моніторингу всього процесу навчання.

Контроль за напрямком погляду людини не є новітнім підходом. Він має широке застосування у різних сферах, але в педагогічній діяльності він може використовуватися у якості додаткового інструмента для контролю знань, впровадження якого у вигляді використання систем дистанційного контролю знань, шляхом моніторингу та оцінки напрямку зору здобувача знань, що, в свою чергу, забезпечує неможливість використовувати додаткові, сторонні матеріали та літературу. Це, у свою чергу, збільшує достовірність і точність оцінки знань студентів.

Основною метою даної роботи є підвищення достовірності оцінювання рівня знань студентів при використанні систем дистанційного контролю знань, шляхом моніторингу та оцінки напрямку зору тестованого, що, в свою чергу, забезпечує неможливість використовувати додаткові, сторонні матеріали, літературу, електронні гаджети. аналіз методів оцінки напрямку погляду людини [3]. Для цього буде проведено аналіз предметної області і основних рішень щодо освітнього процесу, дистанційного навчання та контролю знань. Також буде проведено серію експериментів щодо використання різних методів оцінки напрямку погляду для систем управління навчанням. За результатами потрібно буде надати характеристику кожного методу та плюси та мінуси використання.

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Особливості побудови освітнього процесу

Розширений аналіз особливостей освітнього процесу було проведено за такими показниками: освітнє середовище та етапи розробки освітньої програми у залежності від форми освітнього процесу, а також шляхи реалізації освітніх програм [1]. За результатами аналізу [2], що наведено на рисунку 1.1, видно, що будь-яка форма освіти передбачає загальні етапи розробки навчальних програм, які є, в свою чергу, циклічними, такими як модель покрокового проектування навчальної програми ADDIE (аналіз, дизайн, розробка, впровадження, оцінка), загальна форма якої зображена на рисунку 1.2. Кожен етап створення навчальної програми пов'язаний з іншими. На кожному етапі формуються результати, які основою наступного етапу.

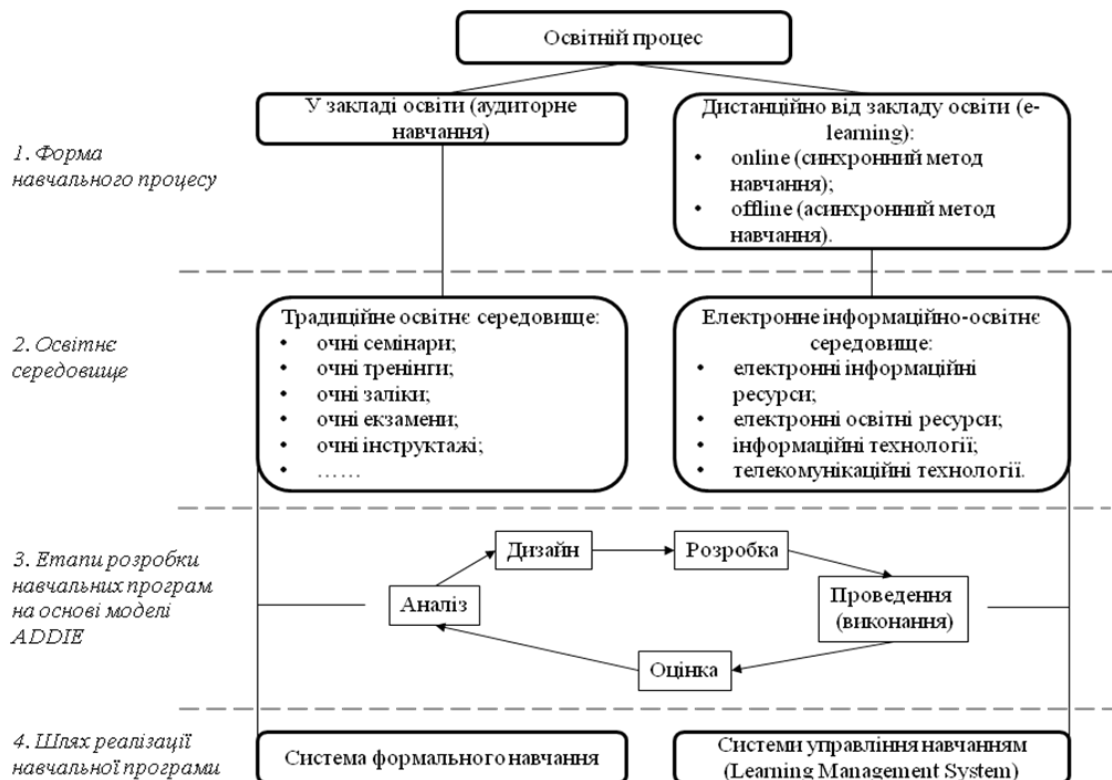


Рисунок 1.1 - Організація різних форм освітнього процесу

На першому етапі моделі ADDIE аналізуються основні потреби цільової групи, визначаються освітні цілі, визначається передбачуване навчальне середовище і визначаються знання та навички студентів. Етап дизайну – це наступний етап моделі ADDIE. Він включає визначення основних питань навчання, вправ та інструментів оцінки змісту, планів занять та вибору інструментів навчання. Етап розробки, відповідно, включає створення навчального матеріалу. Наступний етап розробки включає впровадження розробленої освітньої програми у навчання, після чого призводиться оцінка знань студентів і ефективності навчальної програми.



Рисунок 1.2 – Модель ADDIE

Модель ADDIE забезпечує структурований, покроковий процес проектування та розробки ефективного навчального досвіду. Цей системний підхід гарантує, що розробники навчання враховують усі відповідні фактори та створюють цілісні, добре сплановані курси. При цьому, модель ADDIE є достатньо гнучкою, щоб пристосуватися до широкого спектру навчальних контекстів і стратегій навчання, а етап оцінювання моделі ADDIE заохочує розробників навчальних курсів постійно оцінювати ефективність своїх курсів і вносити необхідні зміни. Такий акцент на постійному вдосконаленні гарантує, що навчальний навчальна програма залишатиметься актуальною, цікавою та ефективною протягом тривалого часу.

1.2 Особливості дистанційного навчання

За складних умов, з якими зіткнулася Україна, універсальним рішенням проблеми формату освітнього процесу стали дистанційні форми навчання, основним принципом яких є інтерактивна взаємодія між суб'єктами освітніх відносин. Дистанційна освіта, незважаючи на декілька десятиліть її розвитку, досить активно увійшла у наше життя саме останнім часом, спочатку в умовах карантину, а пізніше через російську агресію. Якщо раніше мова йшла про те, що використання дистанційної форми навчання забезпечує передусім неперевершену (порівняно з іншими формами навчання) швидкість оновлення знань за підтримки інформаційних ресурсів, що обираються здобувачами освіти зі світових та вітчизняних електронних інформаційних мереж, то в умовах карантину дистанційна освіта залишилася єдиною формою забезпечення всього освітнього процесу.

Дистанційне навчання – це форма організації освіти, що при використанні сучасних інформаційно-комунікаційних технологій, дає змогу навчатися на відстані без прямого контакту у аудиторії між викладачем і студентом. За таких умов акцент у навчанні зміщується на самостійну роботу студента, а викладач виступає в ролі організатора навчального процесу. Це спонукає до пошуку нових засобів навчання, які б відповідали вимогам і потребам освітнього процесу. В основі дистанційної форми здобуття освіти лежить інтенсивна та спрямована на самонавчання самостійна робота студента, який може навчатися в зручному для себе місці, за розкладом, під керівництвом викладачів.

Інноваційні освітні технології мають задовольняти певним системним педагогічним та інформаційно-технологічним вимогам, що продиктовані рівнем науково-технічного прогресу та максимально відповідати принципам відкритої освіти серед основних з яких мобільність учнів і вчителів, рівний доступ до освітніх систем, формування структури та реалізації освітніх послуг [4].

Розвиток нових технологій характеризується низкою показників, що стосуються різних сторін у реалізації систем електронного навчання. Ці показники тісно пов'язані із потребою формування в учнів освітніх компетентностей в контексті сучасних вимог гнучкості, мобільності, навчання та розвитку особистісних якостей студента.

Доступність – це важливий показник систем дистанційного навчання, що відповідає таким параметрам, як наявність та організація доступу до необхідних систем навчання. Поняття “доступу до е-навчання” можна трактувати у даній роботі, як зміст і обсяг послуг, наявних у певний час. Більш широкому впровадженню систем дистанційного навчання для забезпечення більшої доступності можуть перешкоджати такі причини, як наявність достатньої кількості комп'ютерів, програмного забезпечення і необхідних сервісів, доступу до Інтернет, включаючи ширококутний доступ та швидкість з'єднання. Від розгляду цих питань, суттєво залежить вибір платформи реалізації дистанційного навчання, на базі якої організується добір і використання різноманітних типів ресурсів, та їх оптимізація.

Якість освітніх послуг - це ще один показник, систем дистанційного навчання. Якість дистанційного навчання і її оцінювання мають декілька характеристик таких, як: зміст освіти, рівень підготовки методичних та навчальних матеріалів; персонал і кваліфікація викладачів; стан матеріально-технічного забезпечення; управління навчальним процесом; рівень знань та компетентностей учнів та інших.

Точність оцінки результатів навчання є важливим показником систем дистанційного навчання. Серед труднощів, які виникають при оцінюванні при дистанційному навчанні є такі, як висока вартість потужних серверів з великою кількістю клієнтів, ризик відмови обладнання, необхідність опанування технології оцінювання студентами та викладачами [5]. Визначення параметрів оцінки якості освітніх ресурсів – це окремий комплекс проблем, що відноситься до розробки вимог і стандартів для

освітнього програмного забезпечення. Не зважаючи на те, що стандарти у галузі дистанційного навчання були розроблені з метою визначення шляхів і способів використання у педагогічній діяльності навчальних об'єктів, реалізованих засобами інформаційно-комунікаційних технологій, така ситуація не була остаточним рішенням, а сприяла подальшому пошуку [6]. Загалом із розвитком електронного навчання зростають вимоги до якості освітніх послуг, яка, як свідчать дослідження, суттєво залежить від технологій оцінювання електронних ресурсів та матеріалів та від технологій їх створення та надання користувачеві.

Ще один показник, пов'язаний з вимогами до систем дистанційного навчання – це міра адаптивності. Це застосування спеціалізованих систем навчального призначення, що ґрунтуються на моделюванні індивідуальних траєкторій учня чи студента, його рівня знань. У зв'язку з цим, поширення набувають спеціальні адаптивні технології дистанційного навчання, що враховують особливості індивідуального прогресу здобувача. Адаптивність має включати такі особливості, як координація процесу навчання відповідно до рівня підготовки, підбір темпу навчання, діагностику досягнутого рівня засвоєння матеріалу.

Побудова адаптивної моделі студента, що враховувала б особистісні характеристики, такі як рівень знань, індивідуальні дані, поточні результати навчання, і розробка технологій відстеження його навчальної траєкторії є складною математичною і методичною проблемою [5]. Розвиток даного типу систем, здебільшого з елементами штучного інтелекту, є досить трудомістким. Зростання ступеню адаптивності є однією з тенденцій розвитку систем електронного навчання, що відбувається за рахунок удосконалення технологій подання, зберігання і добору необхідних засобів. Різні навчальні матеріали, ресурси і сервіси можуть бути надані за потребою здобувача знань, та дають можливість динамічної адаптації до досягнутого рівня знань та компетентності.

Ще один важливий показник має відношення до інтеграції та цілісності

систем дистанційного навчання. Ця проблема виникає у зв'язку з створенням відкритого середовища навчання, що забезпечує широкий доступ до освітніх ресурсів, вибір та зміну темпу навчання, його змісту, часових та просторових меж в залежності від потреб користувачів. У зв'язку з цим, наукові основи оцінювання інформаційних технологій та способів їх застосування потребують подальшого розвитку.

Сучасні технології спрямовані на підтримування різних типів діяльності вчителя у віртуальному комп'ютерному класі. Щоб організувати навчальну діяльність в таких спільнотах, що навчаються і взаємодіють віртуально в режимі онлайн, використовуються функції, що забезпечують колективний доступ до навчального контенту для групи користувачів, можливість для вчителя переглядати всі комп'ютери у групі, звертати увагу учнів, підключати або відключати учасників навчального процесу, поширювати файли або посилання серед цільової групи учнів, надсилати повідомлення конкретним учням. Організація навчання у віртуальному класі потребує застосування апаратно-програмних засобів доставки навчального контенту, що також суттєво залежить від добору відповідних технологій.

Ще одна вимога до систем дистанційного навчання стосується безпеки освітнього середовища і передбачає аналіз ризиків та переваг використання комп'ютерних технологій у навчанні.

З огляду на визначені вимоги систем дистанційного у сучасному освітньому процесі виникає потреба у певній інформаційно-технологічній платформі, яка могла б підтримувати нові форми навчання у відповідності сучасним вимогам доступності, гнучкості, мобільності, індивідуалізації та відкритості освіти. Продуктивним видається підхід, за якого проблеми розвитку е-навчання вирішувалися б через призму нових технологій, що надали б необхідну основу для дослідження цих систем, їх розробки і використання. Зокрема, перспективним є впровадження технології машинного навчання при розробці рішень великої кількості вимог .

1.3 Огляд LMS і TMS

Для розробки і управління дистанційним навчанням існують відповідні інструменти, зокрема це системи управління навчанням (LMS) та системи управління тренуванням (TMS). TMS і LMS – це два різні інструменти, що використовуються для організації та проведення дистанційного навчання. Вони мають багато спільного, однак є ряд особливостей, які відрізняють їх.

LMS - це комплексна платформа, яка містить усі необхідні інструменти для створення стабільного процесу дистанційної освіти. Дана платформа дозволяє керувати дистанційним навчанням, забезпечуючи такі функції, як керування навчальним контентом, проведення занять і відстеження прогресу. Навчальні матеріали даної платформи можуть бути представлені в різних форматах, таких як текст, презентації, та інші медіа формати. Система управління навчанням зазвичай керує процесами, пов'язаними з плануванням, проведенням та оцінюванням онлайн-курсів, такими як:

- створення курсів дистанційного навчання;
- завантаження контенту до курсів;
- проведення курсів онлайн;
- зарахування студентів до курсів;
- спілкування зі студентами;
- відстеження і оцінка успішності студентів.

Серед основних переваг використання системи управління навчанням, можна виділити доступність, просте відстеження прогресу та створення контенту. Доступність є однією з головних переваг системи управління навчанням і полягає у тому, що вони легко доступні для користувачів. Якщо користувач має доступ до мережі Інтернет, він може отримати доступ до курсів у будь-який час та з будь-якого пристрою. Така доступність дозволяє користувачам навчатися у власному темпі та в будь-якому місці. Додатково, системи управління навчанням надають можливість студентам зв'язатися зі своїм викладачем або колегами по курсу, щоб уточнити будь-які питання, що

виникають впродовж вивчення матеріалу.

Відстеження прогресу та ефективності занять має вирішальне значення як для викладачів, так і для студентів. Використовуючи платформу з системами управління навчанням, викладачі можуть легко відстежувати прогрес своїх учнів під час проходження курсів, а також мати доступ до інформації про кожний етап навчання, інформація про що буде представлена у зручних форматах, таких як таблиці та діаграми. Такий доступ до інформації про прогрес дозволяє викладачам проводити всебічний аналіз курсу та вносити вдосконалення на основі отриманих результатів.

Системи управління навчання забезпечують інструменти, які надають викладачам широкі можливості для адаптації звичайних курсів до дистанційного навчання, а також простоту створення онлайн-курсів. Тому розробка курсів, зміна і доповнення курсів матеріалами значно спрощується при використанні систем управління навчання.

Системи управління навчанням можна адаптувати до різних освітніх потреб. Залежно від розміру організації, бізнес-цілей та навчальних закладів, існують різні типи підходів до LMS. Серед типів систем управління навчання, можна виділити типи за певними особливостями, як показано на рисунку 1.3:

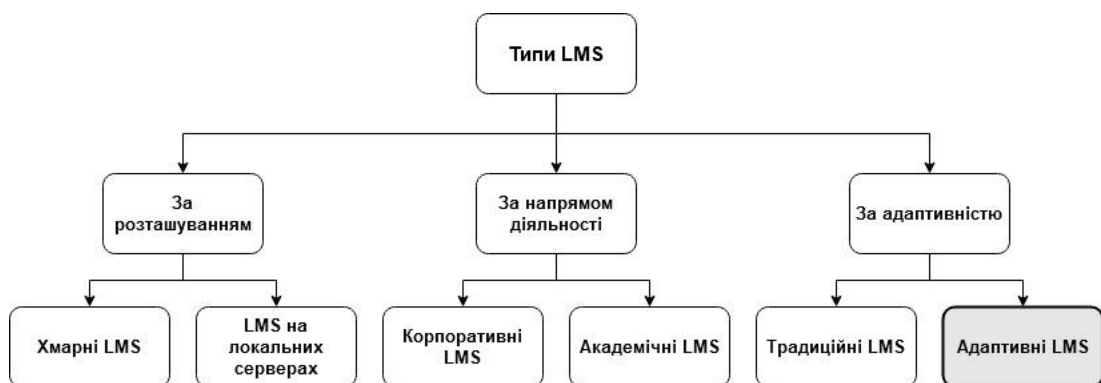


Рисунок 1.3 – Типізація LMS

Хмарна розробка програмного забезпечення LMS виконується та завершується на серверах провайдера. Це забезпечує простоту доступу та масштабованість. Така LMS не потребує встановлення і підтримується

постачальником послуг.

LMS можна розміщати на власних серверах. Така LMS пропонує повний контроль над середовищем електронного навчання. Даний варіант дозволяє більш широкий спектр налаштувань, однак такі системи додатково потребують займатися обслуговуванням та оновленнями систем.

LMS з відкритим кодом – це найдоступніша з точки зору впроваджень рішення для систем управління навчання. Такі LMS мають настроюваний безкоштовний вихідний код для організацій, які хочуть отримати економічні рішення.

Корпоративні LMS розроблені з урахуванням потреб бізнес-навчання. Такі LMS зосереджені на розвитку навичок та компетенцій, а також на професійному зростанні. Такі системи часто включають функції, що відстежують ефективність та аналізують рентабельність інвестицій.

Академічні LMS – це системи управління навчання, розроблені для закладів освіти, і роблять акцент на управлінні заняттями, оцінюванні та спілкуванні між учнями та викладачами. Академічна LMS створена для підтримки структурованого методу навчання, часто інтегруючись з іншими системами дистанційного навчання.

TMS – це програмний застосунок, який забезпечує управління тренуванням. Основною метою TMS є управління та контроль навчального процесу, моніторинг та аналіз його ефективності. На відміну від LMS, TMS орієнтована на корпоративний сектор, і, відповідно, надає складну систему звітності, різноманітні інтеграції з корпоративною стратегією та платформу, що налаштовується під конкретні бізнес-потреби.

Системи управління тренуванням орієнтовані на організацію навчальних сесій і призначені для адміністраторів навчального процесу. В основі цих систем лежать функції та можливості з орієнтацією на бізнес-процеси підприємства.

Використання основних функцій системи управління тренуваннями дозволяє оптимізувати витрати часу на керування навчальним процесом.

Основні функції системи управління навчальним процесом, включають ті, що зосереджені на плануванні та управлінні ресурсами, операціях і логістиці, управлінні інструкторами, фінансовому плануванні, результативності, звітності та управлінні циклом продажів. Узагальнюючи, функції системи управління навчальним процесом є наступними:

- організація логістики та ресурсів;
- управління календарем курсів та планування: онлайн або віртуальних тренінгів, особистих сесій та управління воркшопами;
- управління персоналом, що призводить тренування;
- управління навчанням під керівництвом інструктора - віртуального або аудиторного навчання;
- відстеження фінансів;
- інструменти для бізнес-аналітики.

Серед основних переваг використання системи управління тренуваннями, можна виділити економічну ефективність, швидку адаптацію співробітників та підтримку професійного рівня. Використовуючи програмне забезпечення TMS, компанії та підприємства надають своїм співробітникам якісне навчання, відповідне сучасним вимогам до професійних навичок співробітника. Навчання основам роботи та нюансам документообігу призводить до підвищення задоволеності працівників на всіх рівнях. Це, в свою чергу, дозволяє бізнесу стабільно розвиватися і прогресувати, не витрачаючи час на частий найм і повторне навчання нових співробітників.

Платформи TMS повинні надавати усі необхідні бази професійних знань у кілька кліків, усуваючи необхідність зупиняти прогрес роботи досвідченого працівника для допомоги, або навчання нового. Крім того, навчальна програма може містити навчальні матеріали, що дає можливість працівникам навчатися у найбільш корисний та ефективний для них спосіб. Детальна вбудована аналітична система може відстежувати ефективність навчання, а також дозволяє відстежувати прогрес входження нового співробітника в

робочий процес компанії та виявляти будь-які проблеми, з якими він може зіткнутися під час адаптації. Платформи TMS дозволяють підприємствам заощаджувати на навчанні завдяки їхнім комплексним можливостям та гнучкості. Використання системи управління тренуваннями дозволяє організувати і планувати як дистанційні, так і очні заняття, які, однак потребують викладача. З іншого боку, LMS - це програмне забезпечення, призначене для управління дистанційним навчанням та здобувачами знань. Воно дозволяє викладачам організувати, проводити та контролювати онлайн-курси через інтерфейс користувача. Розширене порівняння цих систем наведено на рисунку 1.4

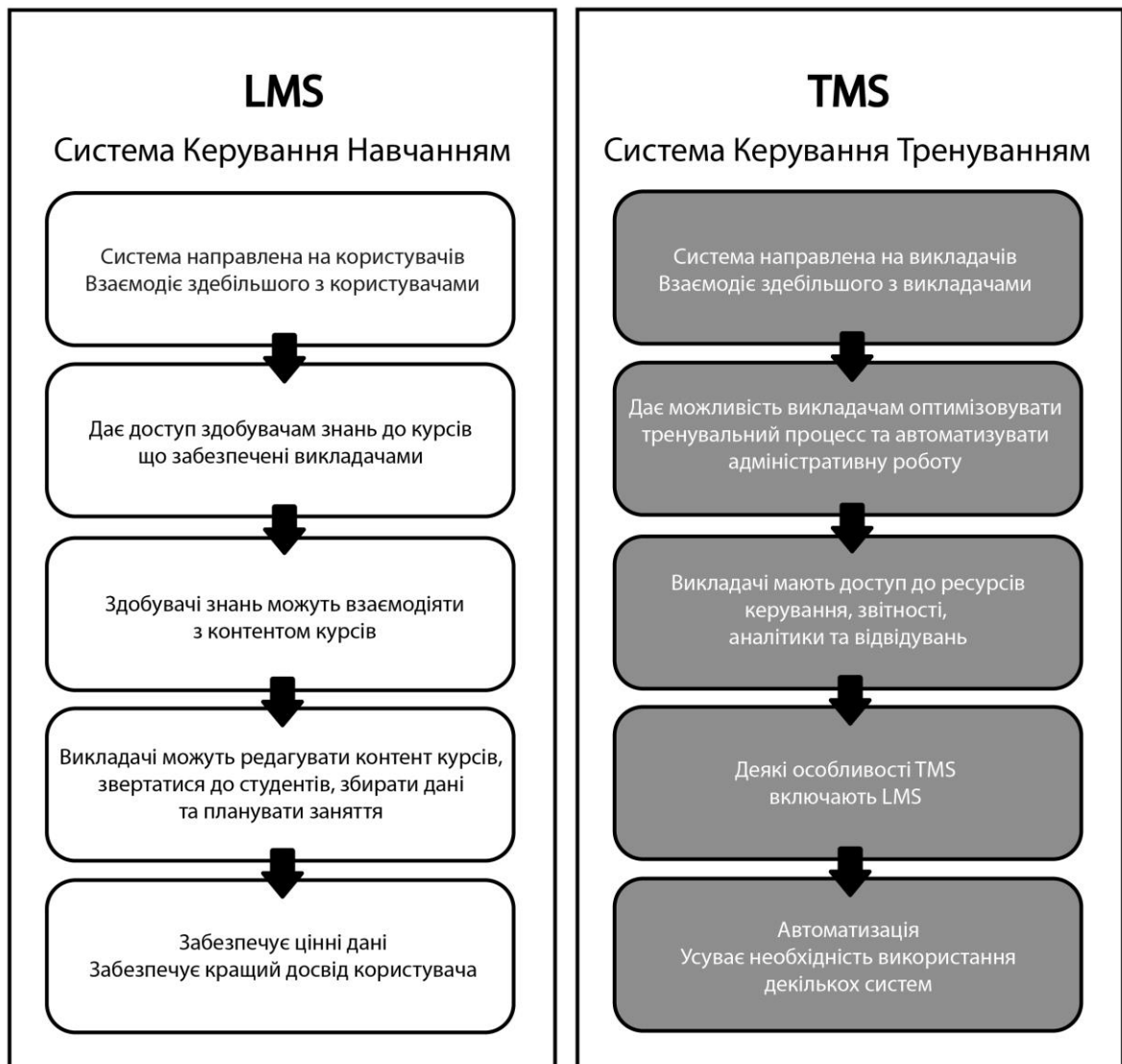


Рисунок 1.4 – Порівняння LMS та TMS

Різниця між системою управління навчанням та система управління навчальним процесом переважно полягає у різній направленості цих систем. Так, різницю між ними можна наочно переглянути на рисунку 1.4:

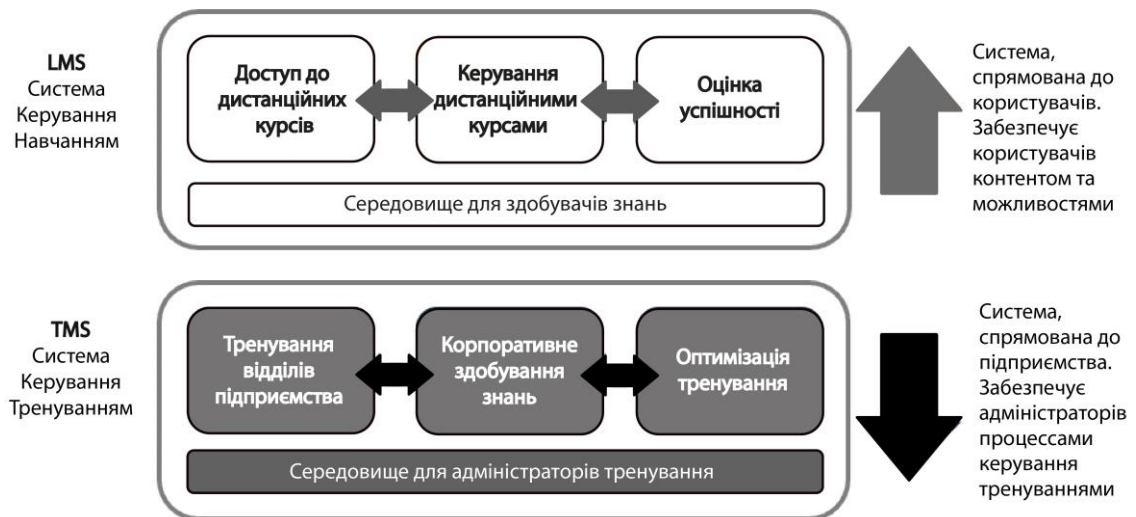


Рисунок 1.5 – Спрямованість систем

У разі потреби, система управління навчанням та система управління тренуванням можуть бути об'єднані, або можуть включати особливості одна одного. Саме тому організації, які поєднують як аудиторне, так і віртуальне навчання, можуть вирішити інвестувати в обидва типи програмного забезпечення. Незалежно від того, чи використовується для навчання підхід до тренування з викладачем, та дистанційне навчання окремо, чи разом у сценаріях змішаного навчання, знадобиться навчальне програмне забезпечення, яке може взаємодіяти одне з одним, щоб надати рішення, адаптоване до потреб користувачів.

1.3.1 Особливості та переваги Adaptive Learning System

Адаптація навчального матеріалу до особливостей здобувачів знань – це особливість, яка є основою системи адаптації навчання (ALS). Дистанційне навчання, яке впроваджує та використовує навчальні

середовища, що забезпечують здобувачів знань адаптованим до їх особливостей контентом, забезпечує більш ефективний процес навчання.

Одним з підходів до адаптації контенту є використання системи рекомендації. Системи рекомендації для систем дистанційного навчання – це окремий елемент, що пропонує студенту контент, що його цікавить, спираючись на його властивості, уподобання та активності.

Адаптивні системи потрібні, щоб адаптувати контент під персональні потреби користувачів, наприклад, адаптація контенту для людей з інвалідністю, або обмеженими можливостями. Адаптація контенту може приводитися для людей, які не знають мови, або знають погано. Адаптація до особливостей окремого студента, допомагаючи у тих моментах, що викликають проблеми для студента. Відповідно, можна виділити основні вимоги до адаптивних систем, що відрізняють їх від традиційних:

- можливість перекладу контенту;
- взаємодія зі звуком – керування голосом, та автоматичне читання;
- генерація підказок.

Параметри, за якими такі системи можуть бути імплементовані, можуть бути визначені за допомогою моделей навчальних стилей [7,8]. Навчальні стилі – це впорядковані представлення про вподобання та особливості навчання окремих індивідів, представлені у вигляді моделей, що аналізують особливості навчання за певними характеристиками. Засновуючи алгоритм роботи системи для ідентифікації особливостей навчання користувачів і класифікації їх як навчальних стилів, можна побудувати тришарову схему інтерфейса адаптації навчального матеріалу, що наведено на рисунку 1.6. Дана система спрямована на створення рекомендацій щодо навчального контенту з метою забезпечення ефективнішого самостійного навчання шляхом індивідуалізації особливостей системи до окремих користувачів, надаючи рекомендації щодо контенту, та змінюючи особливості інтерфейсу користувача відповідно до бажань та особливостей користувача.

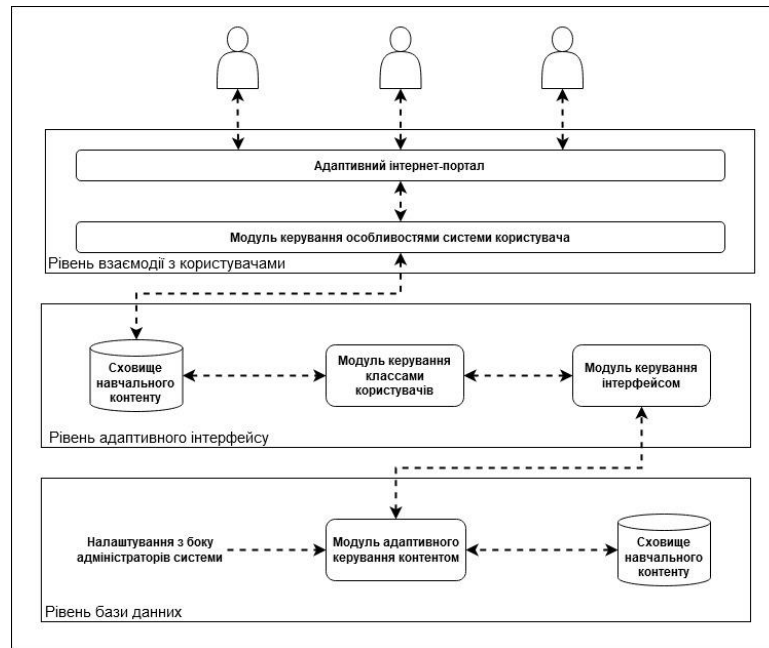


Рисунок 1.6 – Інтерфейс адаптації навчального матеріалу

Контроль навчання у системах з системами адаптації рекомендацій є особливо важливим етапом, бо крім основних цілей контролю знань на даному етапі затверджується якість рекомендацій, що були надані користувачу впродовж навчання.

1.4 Аналіз ролі машинного навчання в LMS та TMS

Машинне навчання має багато застосувань. Прикладами є прогнозування цін на житло, фільтрація спаму, освіта, структурування даних у системах охорони здоров'я, прогнозування реакції на ліки, та чимало інших рішень. Однак, окрім усього іншого, машинне навчання може активно використовуватися у управлінні навчанням, зокрема у якості інструменту для динамічної адаптації контенту для системи управління навчанням і індивідуалізації самої системи управління навчанням.

Машинне навчання стало надзвичайно популярною темою серед організацій, що займаються розвитком, які прагнуть впровадити підхід, заснований на даних, щоб покращити свій бізнес, отримуючи корисну інформацію з даних, які вони збирають. За допомогою моделей машинного

навчання організації можуть постійно прогнозувати зміни у своєму бізнесі та приймати відповідні рішення. Машинне навчання використовує алгоритми, які ітеративно навчаються на даних, щоб покращувати, описувати дані та прогнозувати результати. Після того, як модель навчена, вона може прогнозувати нові дані, які подаються на вхід. Результат, який дасть модель на нових даних, буде залежати від даних, використаних для навчання моделі.

Область використання машинного навчання, яка розглядається у даній роботі, - це освіта. Існує три основні способи надання освіти: очне, онлайн і змішане навчання. Серед можливостей покращити навчання за допомогою впровадження машинного навчання можна виділити:

- використання машинного навчання для оцінки робіт;
- аналіз успішності студентів;
- формування рекомендованого матеріалу;
- контроль знань;
- персоналізоване навчання.

Оцінювання робіт студентів є важливим етапом контролю знань будь-якої навчальної програми. У оцінювання є свої особливості і можливості інтеграції з машинним навчанням. Машинне навчання пропонує потенційне рішення для таких проблем оцінювання робіт, як витрата великої кількості часу на оцінку робіт та суб'єктивність оцінки складних робіт. Використовуючи алгоритми машинного навчання, можна забезпечити для оцінювачів робіт значно менший час на оцінювання робіт.

Аналіз успішності студентів – це ще один з варіантів використання машинного навчання. Аналіз успішності може допомогти з прогнозуванням відсіву студентів на курсах дистанційного навчання, виявлення прогресу студентів, або з оцінкою ефективності освітньої програми. Відсів студентів – це процес відрахування студентів впродовж здобуття знань і продовження навчання, що дуже помітно впродовж дистанційного навчання. Хоча машинне навчання було запропоновано для прогнозування відсіву на курсах електронного навчання, в цій галузі все ще існують можливості для

досліджень. Однією з потенційних можливостей є вивчення ефективності різних систем і моделей прогнозування відсіву в інших умовах проведення курсів, таких як змішане навчання, дистанційна та класична освіта. Загалом, існує багато можливостей використання машинного навчання у системах управління навчанням, основні з яких зображено на рисунку

Алгоритми машинного навчання можуть використовуватися для обробки великих обсягів даних, якими оперують платформи з системами управління навчанням, щоб визначити, які курси будуть найкращими для студента, який зацікавлений у вдосконаленні певного набору навичок. Використання інформації про вже пройдені студентом курси може допомогти створити правильні рекомендації курсів.

Персоналізоване навчання ґрунтується на індивідуальних особливостях учнів та способах їхнього навчання. Кожна людина навчається по-різному і має унікальний профіль учня. Цей профіль базується на індивідуальному стилі навчання, який складається з певної поведінки та ставлення. Персоналізація освіти кожного учня може призвести до кращого навчання. Одним із способів персоналізації освіти є використання систем з алгоритмами машинного навчання, які надають корисні пропозиції щодо нових матеріалів і повторення старих.



Рисунок 1.7 – Можливості застосування машинного навчання в LMS

На даний момент існують декілька платформ дистанційного навчання,

що використовують штучний інтелект. Серед таких платформ було розглянуто:

- Paradiso LMS;
- Docebo;
- EdApp;
- Absorb LMS.

Paradiso LMS – це навчальна платформа, яка широко використовує технології машинного навчання і пропонує інтуїтивно зрозумілий і зручний інтерфейс. Завдяки готовому контенту та шаблонам дизайну дана платформа дозволяє користувачам завантажувати контент, презентації та матеріали для швидкого створення професійних курсів з мінімальними зусиллями. Інструменти даної платформи дозволяють створювати курси дистанційного навчання на будь-яку тему за мінімальну кількість часу. Моделі машинного навчання допоможуть з такими інструментами, як графічне зображення даних, створення голосової озвучки презентацій та тексту і з оцінюванням. Дана платформа дозволяє широкий діапазон налаштування, пропонуючи редаговані шаблони для контенту та дизайну. Така гнучкість гарантує, що користувачі можуть швидко адаптувати та оновлювати курси відповідно до своїх конкретних навчальних цілей.

Docebo – це платформа, яка використовує штучний інтелект. Вона пропонує цілий ряд функцій для ефективного створення та налаштування навчальних програм. Серед цих функцій є високофункціональні інструменти для створення курсів дистанційного навчання для певної аудиторії, автоматизований та персоналізований навчальний процес, автоматичні тестування та формування рекомендованого матеріалу.

Система управління навчанням EdApp включає моделі машинного навчання, які спрощують процес створення курсів. Користувачі створюють цілі уроки без складнощів або великих досліджень. Дана платформа розроблена відповідно до поточних стратегій навчання, пропонуючи невеликий за обсягом контент, який прискорює розвиток компетенцій і

прискорює розробку курсу.

Absorb LMS - це система управління навчанням, яка тренується з кожним пошуковим запитом, зарахуванням на курс або додаванням контенту. Аналізуючи попередній вибір студентів, ця система управління навчанням постійно підлаштовується і передбачає потреби учнів на основі їхньої поведінки, а також намагається передбачити потрібний контент, відповідно до шуканих користувачем компетенцій.

Проаналізувавши системи управління навчанням, можна виділити декілька особливостей щодо використання моделей машинного навчання. Серед цих особливостей варто означити:

- широке використання у таких LMS інструментів створення курсів;
- широке використання інструментів адаптації та персоналізації навчання;
- мінімальне використання моделей машинного навчання в системах контролю знань.

Отже, у проаналізованих платформ системи контролю знань, не включають інструменти, основані на моделях машинного навчання. Це дає змогу провести дослідження життєздатності моделей машинного навчання у системах контролю знань у даній роботі

1.5 Контроль знань в навчанні

Контроль знань студентів - це невід'ємна і дуже важлива частина процесу навчання, яка полягає у перевірці рівня знань у студентів. Від його правильної організації багато в чому залежить ефективність управління навчально-виховним процесом і якість підготовки спеціаліста. При використанні досконалішої системи контролю знань, дані щодо ефективності навчання та рівня знань студентів є точнішими.

Головною вимогою до системи контролю є забезпечення змісту і достовірності інформації, для прийняття правильних і ефективних рішень з

управління якістю навчання.

Контроль навчальної діяльності повинен відбуватися, дотримуючись наступних принципів [9]:

- плановості;
- систематичності й системності;
- об'єктивності;
- відкритості й прозорості;
- економічності;
- тематичності (модульності);
- врахування індивідуальних можливостей студентів;
- єдності вимог.

Так, на даний момент, студентами при дистанційному, або гібридному навчанні широко можуть використовуватися сторонні матеріали та гаджети для того щоб допомогти з тестуваннями. Оцінки таких здобувачів знань не завжди відповідають їх рівню підготовки і це, відповідно, зменшує загальний рівень підготовки спеціалістів за даною навчальною програмою. Але використання систем контролю знань з технологічними рішеннями, що виявляють такі спроби “допомоги” при проходженні тестів, дозволяє підвищити якість навчання, зменшити тиск на викладачів та забезпечити точніші оцінки ефективності освітніх програм.

Згідно з проведеним дослідженням [1] найменше уваги у існуючих системах контролю знань приділено саме функціоналу, що запобігає використанню сторонніх матеріалів, тобто механізму, що підвищує достовірність оцінювання рівня знань користувача системи тестування.

1.6 Мета та задачі дослідження

Метою даної роботи є дослідження та вдосконалення методів оцінки напряму погляду людини в контексті створення адаптивної системи управління навчанням.

Для досягнення поставленої мети мають бути вирішені наступні задачі:

- визначення можливостей персоналізованого використання LMS;
- оцінка точності визначення напрямку погляду на основі існуючих методів;
- розробка моделі адаптивної системи управління навчанням на основі тривалості фокусування та напрямку погляду;
- тонке налаштування параметрів нейромережевих моделей для визначення напрямку погляду;
- оцінка впливу умов зйомки на точність визначення напрямку погляду.

Для цього буде проведено експерименти з використанням різних методів оцінки напрямку зору, проведено аналіз отриманих даних та надано характеристики та особливості різних методів.

Подальшим шляхом розвитку дослідження є створення LMS з вбудованою системою контролю знань, у якій крім аналізу напрямку погляду для оцінки знань та персоналізації навчального контенту, можливо впровадити більше взаємодії з відеозображенням користувача, такі як впровадженні методів прискореної високоточної авторизації користувача на основі розпізнавання обличчя, а також додаткових функцій, таких як керування системою за допомогою погляду, забезпечуючи людям з обмеженими можливостями інтерфейс віддаленої взаємодії з навчальною системою.

2 ВИБІР ІНСТРУМЕНТІВ І МЕТОДІВ РОЗРОБКИ

Згідно з поставленим завданням, необхідно провести експерименти, щодо дослідження особливостей використання системи моніторингу та оцінки напрямку зору для системи контролю знань. Для цього потрібно створити відповідну програму, яка симулюватиме використання системи оцінки напрямку зору. Для створення, потрібно обрати відповідні методи та інструменти, зокрема потрібно обрати мову програмування, методи детектування людини та очей і методи обробки даних.

Комп'ютерний зір – це сукупність технологій, які дають можливість цифровим системам проводити виявлення, відстежування та визначення особливостей об'єктів у реальному світі. Основною ідеєю використання елементів комп'ютерного зору у LMS знаходиться на етапі контролю знань прикладом якого є проходження тестів під наглядом системи, а не викладача.

2.1 Вибір мови програмування

Мови програмування – це основний інструмент розробки сучасного програмного забезпечення. Кожна мова програмування має свій унікальний набір переваг і недоліків, задовольняючи різні потреби та вподобання розробників. Вибір мови програмування для конкретного проекту часто залежить від таких факторів, як вимоги проекту, цілі продуктивності та особисті уподобання.

Мова програмування – це важлива складова даної роботи. При виборі мови програмування для даного проекту важливо виділити основні особливості, які важливі для даного проекту. Зокрема, варто виділити наступні особливості, за якими ми обираємо мову програмування:

- гнучкість та багатофункціональність;
- оптимізація показників продуктивності;

- підтримка спільноти.

Гнучкість та багатофункціональність – особливість, яка полягає у універсальності мови програмування з точки зору використання різних підходів та платформ для вирішення завдань. Оптимізація показників продуктивності – інша особливість, яку ми шукаємо це висока обчислювальна ефективність, або інструменти для оптимізації використання системних ресурсів.

Підтримка спільноти – це інша важлива особливість мов програмування, яка полягає у тому, наскільки широке коло спільноти використовує дану мову програмування та наскільки активно спільнота розробляє інструменти, бібліотеки та модифікації. Широка підтримка дозволяє мінімізувати час, який витрачається на вирішення проблем, що виникають під час розробки та дає широкий інструментарій вже розроблених рішень завдань.

2.1.1 Python

Python - це об'єктно-орієнтована мова програмування загального призначення, яка популярна в науці про дані завдяки своїм багатим бібліотекам і фреймворкам, що пропонують можливості глибокого навчання, структурованого машинного навчання та здатності працювати з великими обсягами даних.



Рисунок 2.1 – Переваги і недоліки мови програмування Python

Простий синтаксис Python та легкість інтеграції з іншим програмним забезпеченням робить її швидким та гнучким рішенням як для науковців, так і для розробників. Основні переваги та недоліки Python наведені на рисунку 2.1

Технології, які використовують цю мову програмування, дуже різноманітні і включають розробку веб-додатків, IoT, DevOps, блокчейн та науку про дані. Сфери застосування практично не мають обмежень, і Python може підійти для будь-якої галузі з простими та складними рішеннями.

2.1.2 Java

Java - це мова програмування високого рівня, яка базується на класах та є об'єктно-орієнтованою. Вона була створена з метою вимагати якомога менше базових технологій, наскільки це можливо. Тому Java є однією з найпродуктивніших мов протягом досить тривалого часу.

Вплив Java на сферу комп'ютерного програмування неможливо переоцінити. Ця мова сприяє швидкому вирішенню складних проблем програмування, з якими стикаються компанії, що займаються розробкою Java. Java також полегшила створення нових додатків і прискорила випуск оновлених версій. Java, як і будь-яка інша технологія, має як плюси, так і мінуси.

2.1.3 C++

C++ додав безліч нових можливостей у мову Сі. Його популярність була викликана об'єктно-орієнтованістю мови. Зараз C++ широко використовується для розробки програмного забезпечення, будучи однією з найпопулярніших мов програмування. З її допомогою створюють операційні системи, різноманітні прикладні програми, драйвери пристроїв, ігри тощо.

Серед найпопулярнішого програмного забезпечення, написаного на C++, або використовуючи інструменти, написані на C++, є СУБД MySQL,

інтернет-браузер Mozilla Firefox, більша частина програмного забезпечення від Microsoft: операційні системи сімейства Windows, IDE Visual Studio, Internet Explorer, Microsoft Office. Adobe Photoshop, Adobe Illustrator і Adobe Premiere Pro цілком написані на C++.

2.1.4 JavaScript

JavaScript - це мова програмування, яку використовують розробники для створення інтерактивних веб-сторінок. Функції JavaScript можуть поліпшити зручність взаємодії користувача з веб-сайтом: від оновлення стрічки новин у соціальних мережах і до відображення анімації та інтерактивних карт. JavaScript є мовою програмування під час розроблення скриптів для виконання на стороні клієнта, що робить його однією з базових технологій у всесвітній мережі Інтернет. Наприклад, карусель зображення, меню, що випадає за кліком, і динамічно мінливі кольори елементів на веб-сторінці, які виникають під час перегляду сторінок в Інтернеті, виконані за допомогою JavaScript. Для використання у проекті з комп'ютерним зором, JavaScript не дуже підходить, адже спеціалізується на веб-розробці.

2.1.5 Вибір мови програмування

Проаналізувавши популярні мови програмування, було зроблено вибір на користь мови програмування Python. Даний вибір був здебільшого обумовлений великою кількістю фреймворків та широкою підтримкою спільноти.

Серед областей застосування даної мови програмування, які особливо добре підходять для розробки на Python, є такі, як машинне навчання, наука про дані, обробка мови, та розробка малих проектів. Усі ці особливості мови програмування Python можуть бути використані при розробці проекту. [10].

2.2 Методи розпізнавання людини

Розпізнавання людини та обличчя - це проблема комп'ютерного зору, яка полягає у пошуку людей на зображеннях. Крім того, розпізнавання також початковий крок для багатьох технологій, пов'язаних обробкою обличчя, наприклад, верифікації обличчя, моделювання обличчя, відстеження пози, розпізнавання статі та віку та розпізнавання виразу обличчя. Для даної роботи важливість розпізнавання обличчя у тому, що для спостереження за учнями при проходженні тестів, їх спочатку треба розпізнати у кадрі. На даний момент існує вже багато методів і підходів розпізнавання обличчя, як зображено на рисунку 2.2

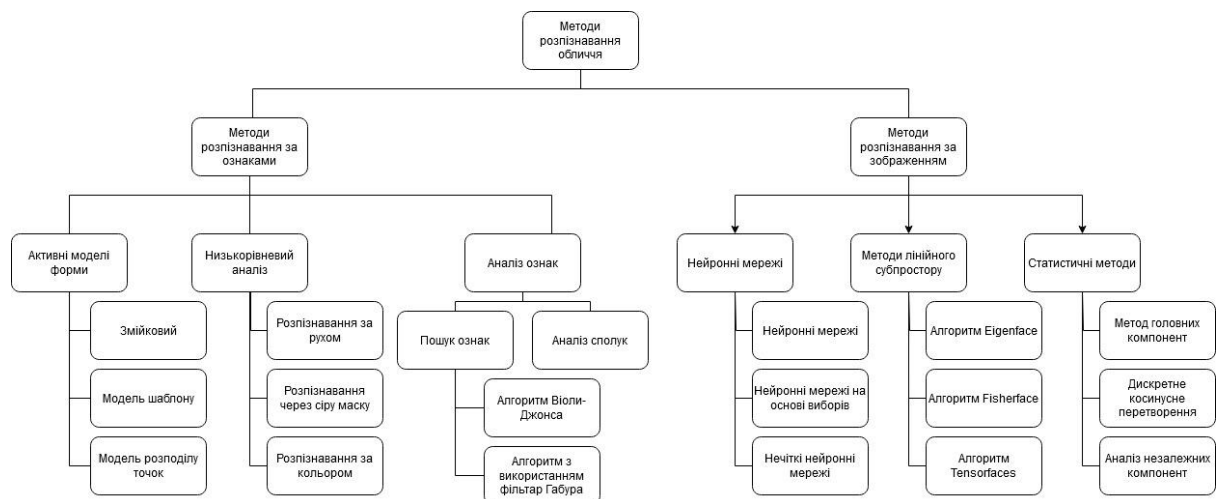


Рисунок 2.2 – Методи і алгоритми розпізнавання обличчя

Розпізнавання обличчя - це банальне завдання для людини, яке ми можемо виконувати природно, майже не докладаючи зусиль. Однак для машин це завдання є складним і вимагає виконання певних обчислювальних кроків. Основним завданням технологій розпізнавання обличчя є знаходження обличчя на зображенні. Основні проблеми, з якими стикається розпізнавання обличчя є оклюзія об'єктів та освітлення. Для рішення цих проблем, існує широкий спектр алгоритмів. В основному, наявні алгоритми поділяються на дві частини: підходи за ознаками і підходи за зображенням.

2.2.1 Методи розпізнавання за ознаками

Розпізнавання облич за ознаками полягає у пошуку особливих регіонів зображення, що мають риси з відповідними особливостями. Підходи, що базуються на розпізнаванні облич за ознаками, поділяються на такі підгалузі:

- використання активної моделі форми;
- низькорівневий аналіз;
- аналіз ознак.

При використанні алгоритмів активної моделі форми (ASM), коли система переглядає зображення, вона пов'язує певні його регіони з рисами обличчя, такими як ніс, рот, або очі, як тільки знаходить близькість до будь-якої з цих рис. Координати цих частин беруться як початкова точка, і на їх основі генерується маска, що на основі вже існуючих даних про обличчя вирішує, які регіони зображення відносяться до рис обличчя. Тренуючись на більшій кількості зображень, можна досягти кращої карти. ASM можна розділити на чотири групи: змійки, модель шаблону, що деформується, модель елемента, що деформується, і модель розподілу точок.

Низькорівневий аналіз використовується для знаходження контуру обличчя і виконується пошук за різними особливостями зображень, такими як колір, рух, та вилучення фону. Іноді низькорівневий аналіз розпізнає фонові об'єкти як обличчя, що може бути вирішено за допомогою аналізу ознак високого рівня.

Аналіз особливостей – це метод, який використовує геометрію обличчя людини як основу до пошуку особливостей. Один алгоритмів, що використовує цю методику – це алгоритм Віоли-Джонса, що використовується, здебільшого для пошуку обличчя. Використовуючи цей алгоритм в системах виявлення об'єктів, можна змінювати характеристики шуканого об'єкта, проводячи навчання на конкретних прикладах. У поєднанні з алгоритмом AdaBoost цей підхід забезпечує відповідну ефективність виявлення об'єктів.

2.2.2 Методи розпізнавання за зображенням

Більшість методів розпізнавання облич на основі зображень працюють за допомогою віконного сканування. Вікно сканується піксель за пікселем, щоб класифікувати обличчя і не-обличчя. Як правило, кожен метод в підходах на основі зображень варіюється з точки зору вікна сканування, розміру кроку, кількості ітерацій і частоти субдискретизації для отримання більш ефективного підходу. Підходи, що базуються на розпізнаванні облич за зображенням, поділяються на такі підгалузі:

- нейронні мережі;
- статистичні алгоритми;
- методи лінійного підпростору.

Алгоритми нейронних мереж натхненні біологічною нейронною мережею людського мозку. Нейронні мережі отримують дані і навчаються розпізнавати шаблон (для розпізнавання облич - шаблон обличчя). Потім мережі прогнозують результат для нового набору схожих облич. Нейронні мережі можна поділити на штучні нейронні мережі (ШНМ), нейронні мережі на основі прийняття рішень (НМР) та нечіткі нейронні мережі (НМН).

Лінійний підпростір - це векторний простір, який є підмножиною більшого векторного простору. У термінах обробки зображень менша частина кадру називається підпростором.

2.2.3 Аналіз методів розпізнавання зображення

Згідно з завданням, потрібно створити систему, яка розпізнаватиме обличчя та аналізуватиме його особливості у реальному часі. Для проведення експериментів з методами розпізнавання, було вирішено зосередитися на методах розпізнавання за ознаками. Крім того, важливими методами, які потрібно проаналізувати є методи з використанням нейронних мереж, адже згідно з дослідженням [11], дані методи мають найвищу продуктивність, а

також метод дискретного косинусного перетворення, бо даний алгоритм він має найнижчу обчислювальна складність. Загалом, основні переваги та недоліки методів, згідно з дослідженням [11] приведені у таблиці 2.1

Таблиця 2.1 - Переваги та недоліки методів розпізнавання обличчя

Методи	Алгоритми	Переваги	Недоліки
1	2	3	4
Активні моделі форми	Зміїна модель	Легкість в управлінні	Алгоритм повинен почати роботу близько до шуканої ознаки
	Модель шаблону	Пристосовується до будь-яких форм	Чутливий до ініціалізованої позиції
	Модель розподілу точок	Забезпечує компактну структуру обличчя	Знаходить об'єкти лінійно
Низькорівневий аналіз	Розпізнавання за рухом	Краще працює при руху	Погано працює з бородою, окулярами
	Розпізнавання через сіру маску	Простий у імплементації	Має нижчу продуктивність
	Розпізнавання за кольором	Працює швидше за аналоги	Дуже чутливий до стану освітленості
Аналіз ознак	Пошук ознак	Висока точність виявлення окремих ознак	Чутливий до стану освітлення та повертання обличчя
	Аналіз сполук	Добре працює при повертаннях об'єктів	Складний у реалізації
Нейронні мережі	Загальні нейронні мережі	Може працювати з неповними даними	Велика обчислювальна складність

Продовження таблиці 2.1

1	2	3	4
Нейронні мережі	Нейронні мережі на основі виборів	Забезпечує краще розпізнавання структури обличчя	Обмежений щодо розташування обличчя
	Нечіткі нейронні мережі	Підвищена точність розпізнавання ознак	Вимагає налаштування нечітких лінгвістичних правил
Методи лінійного субпростору	Eigenface	Простий і ефективний імплементації	Погано працює при масштабуванні зображення
	Fisherface	Ефективно працює із зображеннями різного освітлення та виразів обличчя	Результат залежить від вхідних даних
	Tensorface	Ефективно працює із зображеннями різного освітлення та виразів обличчя	Повинен бути навченим за допомогою специфічних даних
Статистичні методи	Метод головних компонент	Добре працює при статичних зображеннях краях	Погано працює при масштабуванні зображення
	Дискретне косинусне перетворення	Низька обчислювальна складність	Вимагає квантування
	Аналіз незалежних компонент	Ітеративний	Погано працює з великими обсягами даних

2.3 Вибір фреймворків та бібліотек

Отже, серед бібліотек мови програмування Python, для роботи потрібно обрати інструменти, які допоможуть з розробкою. Зокрема, проаналізувавши основні особливості проекту, було обрано такі бібліотеки:

- OpenCV – бібліотека для роботи з зображеннями, отримання даних з камери;
- Tenserflow - бібліотека для роботи з нейронними мережами;
- Dear PyGui - бібліотека для створення інтерфейсу;
- Mediarpipe - бібліотека з існуючими рішеннями щодо знаходження обличчя;
- Imutils - бібліотека що містить багато функцій для роботи з зображенням;
- Math – бібліотека для проведення обчислень;
- Time – бібліотека для роботи з часом;
- Threading – бібліотека для роботи з потоками.
- numpy – бібліотека для роботи з масивами

Бібліотеки для роботи з зображенням, такі як Dlib, Imutils, та OpenCV – є основними інструментами даного проекту при імплементації алгоритмів розпізнавання зображень. Бібліотека DearPyGui використовується для створення простого і ефективного інтерфейсу користувача.

Tensorflow – одна з найпопулярніших бібліотек, що має значну кількість документації, масову підтримку спільноти та вважається однією з найпродуктивніших [12] бібліотек для роботи з нейромережами.

3 РОЗРОБКА МОДЕЛІ ТА МЕТОДІВ ОЦІНКИ НАПРЯМУ ПОГЛЯДУ

Згідно з проведенням пошуком даних, адаптивна LMS – це система управління навчанням, яка адаптується під індивідуальні потреби користувача. Для досягнення цієї мети, вона повинна включати певні компоненти, такі як адаптивний інтерфейс користувача, та модуль адаптації навчального контенту. Згідно до описаних вимог і основних потреб користувачів системи управління навчанням, була побудована узагальнена модель адаптивної LMS, яку можна переглянути на рисунку 3.1

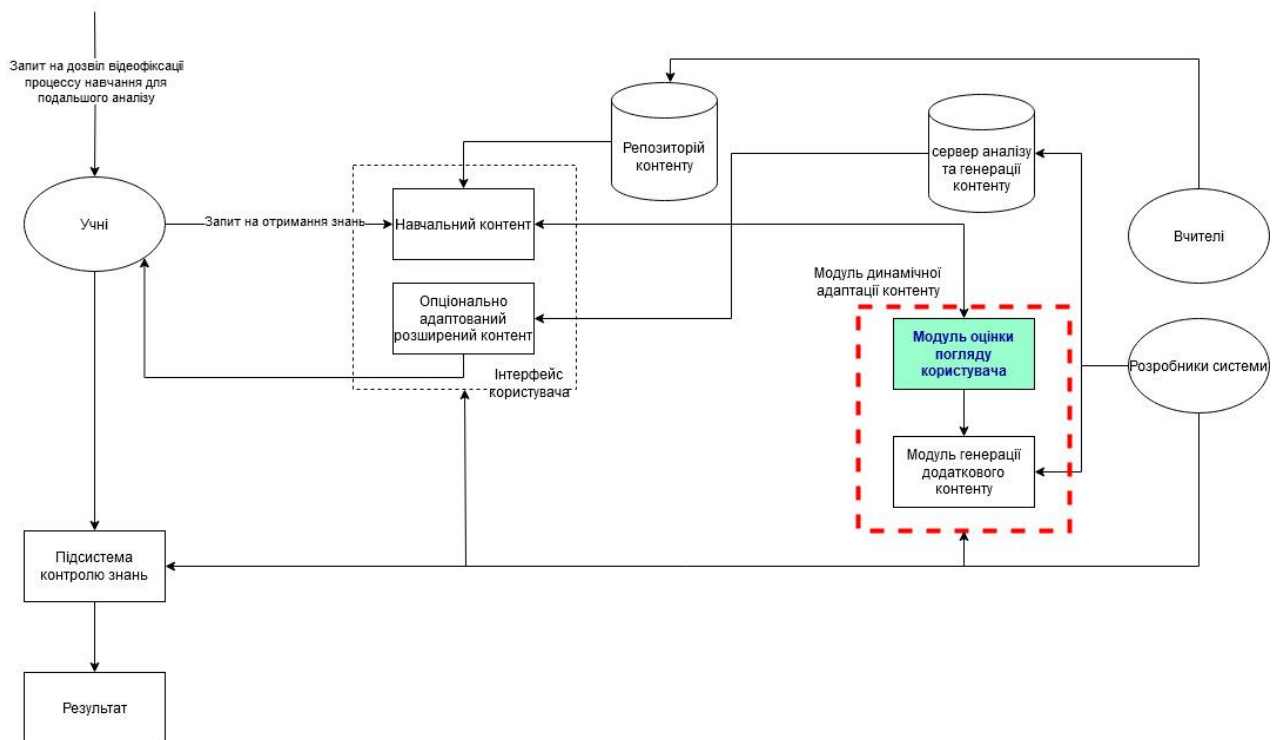


Рисунок 3.1 – Узагальнена модель адаптивної LMS

Дана модель визначає три різних класи користувачів, це:

- учні – це основні користувачі даної системи, які взаємодіють з користувацьким інтерфейсом і підсистемою контролю знань;
- вчителі – це основні впроваджені нового контенту у систему, які взаємодіють зі сховищем контенту;
- розробники системи – це клас користувачів, який слідкує на

налаштовує автоматизовані модулі даної моделі, такі як підсистема контролю знань, модуль динамічної адаптації контенту, та сервер аналізу та генерації контенту.

Сховища даних у даній системі представлені у якості репозиторію контенту, та серверу аналізу та генерації контенту. Репозиторій контенту – це сховище даних, яке дає доступ до даних занять, таких як лекції, тести та інші теоретичні матеріали. Ці матеріали заносяться до системи вчителями і передаються до користувачів у інтерфейсі користувача. Сервер аналізу та генерації контенту виконує аналіз генерованого навчального контенту та збереження вдалих результатів генерованого контенту. Додатково, дане сховище може аналізувати дані про студентів, включно з їхнім прогресом і поведінкою.

Підсистема контролю знань оцінює прогрес і знання студентів на основі результатів тестів або виконання завдань та передає дані в загальну систему для аналізу та адаптації навчання. Ще з учнями взаємодіє інтерфейс користувача, що надає доступ до навчального контенту, та відображає адаптовані матеріали.

Особливо цікавим у даній моделі слугує модуль динамічної адаптації контенту. Саме цей модуль приймає на вході дані про особливості навчання учнів і їх взаємодії з навчальним контентом. У нашій моделі, даними про особливість навчання учня слугує дані про погляд учня. Подібні дані повинні надходити від інтерфейсу користувача до модулю динамічної адаптації контенту, однак обрахування таких даних бере на себе під модуль оцінки напряму погляду користувача. Модуль оцінки погляду користувача отримує дані зображення користувача у реальному часі і повинен згідно зображень розраховувати такі параметри, як:

- час фіксації учня на певній точці на екрані, або поза ним;
- частота фіксації учня на певному елементі у тексті, або поза екраном;
- швидкість споживання навчального матеріалу учнем.

Проаналізувавши дані параметри, ми можемо виділити точки інтересу

конкретного користувача у неадаптованих даних і адаптувати їх до конкретного користувача. Отримані параметри повинні передаватися у модуль генерації додаткового контенту, що забезпечує створення, або модифікацію існуючого текстового контенту.

У такій системі основну роль при аналізі користувача і пошуку параметрів, згідно з якими адаптується контент, відіграє саме модуль оцінки напряму погляду, адже його точність дозволяє з найменшою помилкою отримати дані параметри щодо адаптації, і, відповідно, генерувати адаптований контент, що найкраще підходить до певного учня. Саме тому дослідженню модулю оцінки погляду користувача присвячена ця робота.

3.1 Опис методології проведення експериментів

Проведення аналізу напряму погляду – це комплексне завдання, яке вимагає виконання певних кроків, зокрема:

- знаходження обличчя;
- знаходження положення елементів обличчя;
- аналіз особливостей.

Знаходження обличчя – це етап, який полягає у визначенні розташування обличчя у кадрі. У даній роботі цей етап є важливим для подальшого знаходження рис обличчя.

Знаходження положення елементів обличчя включає знаходження розташування основних рис обличчя, таких як очі, рот та ніс. Даний етап цікавить здебільшого можливістю локалізації зони очей на екрані з метою подальшого аналізу їх особливостей.

Аналіз особливостей у даній роботі включає знаходження вектору погляду та зіставлення напряму погляду до приблизного положення на екрані. Саме даний етап нас цікавить як основний бенефіціар проведення експериментів з методів аналізу погляду.

Для даного дослідження основним елементом інтересу є точність аналізу

напрямку погляду при різних особливостях зовнішнього середовища. Зокрема, нас цікавлять наступні фактори, які виникають при аналізі напрямку погляду:

- освітлення;
- положення голови;
- дистанція.

Освітлення – це фактор, яким описується яскравість освітлення середовища, у якому знаходиться людина, що тестується. Відповідно, освітлення може бути гарним (300 люменів на м²) та поганим (30 люменів на м²). У нашому експерименті одним із параметрів зовнішнього середовища є освітлення, за яким ми протестуємо нашу підсистему аналізу погляду. Іншими параметрами, за яким ми будемо тестувати отримані результати є положення голови, а саме поворот голови під різними кутами, а також дистанція тестованого від камери. Виділимо параметри, за якими ми будемо проводити експерименти у таблиці 3.1.

Таблиця 3.1 – Визначення умов проведення експериментів

Гарне освітлення (X люмен)		
	Відстань від камери 30 см	Відстань від камери 100 см
Кут нахилу 0	Експеримент 1	Експеримент 4
Кут нахилу 15	Експеримент 2	Експеримент 5
Кут нахилу 30	Експеримент 3	Експеримент 6
Погане освітлення (Y люмен)		
Кут нахилу 0	Експеримент 7	Експеримент 10
Кут нахилу 15	Експеримент 8	Експеримент 11
Кут нахилу 30	Експеримент 9	Експеримент 12

Аналіз напрямку погляду є основним етапом, що лежить в основі для експериментів у даному дослідженні. Для вирішення даного завдання існує

чимало методів, зокрема методи на основі тривимірної проекції ока і зіниці, так і інші, на основі зовнішнього вигляду, наприклад методи з використанням моделей машинного навчання.

У цій роботі основним методом буде використання методів глибокого навчання для аналізу особливостей зображення ока. Таке завдання потребує створення, або пошук відповідного даних даних та вибір оптимальної архітектури моделі.

Для проведення експериментів зі змінними зовнішніми умовами, в яких може знаходитися користувач, проведемо дослідження для налаштування гіперпараметрів моделі (дивись розділи 4.1-4.3).

Спочатку у ході проведення експериментів ми оберемо базові параметри мереж, наступним етапом є перевірка того, як успішно та, чи інша архітектура виконує завдання. Після цього обираємо найкращі за отриманими результатами архітектури і проводимо точне налаштування параметрів для забезпечення найкращих результатів у вирішенні завдання.

3.1.1 Огляд інструментів для проведення експериментів

Для проведення експериментів, використовувалися дві різні камери. Перша – це вбудована камера ноутбука KS.0HD06.030, тоді як друга – спеціально взята для проведення тестування камера Imx290 Sensor Starlight.

KS.0HD06.030 – це камера бренду ACER, яка включає такі особливості як автоматичне фокусування та багатофарбовість. Вона має швидкість у тридцять кадрів на секунду та максимальну роздільну здатність у 720p.

Imx290 Sensor Starlight – це, навпроти, окремий модуль камери, який дозволяє зйомку у 1080p з шістдесятма кадрами на секунду. Крім того, дана камера містить чимало параметрів, що можна налаштувати, а також, що особливо важливо для експерименту, мінімальне освітлення у 0.0001 люкс. Тобто дана камера здатна працювати при майже повній відсутності освітлення. Зображення даної камери можна переглянути на рисунку 3.2



Рисунок 3.2 – Камера Imx290 Sensor Starlight

Обидві камери використовуються у двох основних задачах етапу тестування – у перевірці фінального результату та для створення експериментального тестового датасету.

3.1.2 Огляд обраного методу виявлення обличчя і знаходження рис обличчя

На даний момент існує багато імплементацій рішень для знаходження елементів обличчя. Для цього завдання було обрано використання існуючого рішення, а саме використання рішення бібліотеки MediaPipe, що дозволяє локалізувати особливості обличчя через велику кількість тривимірних точок. Дане рішення включає використання попередньо навченої моделі машинного навчання, що працює достатньо швидко для роботи з безперервним потоком зображень. Наше завдання включає використання OpenCV для отримання зображення з камери, бібліотека OpenCV дозволяє оброблювати кожний кадр окремо, а код, що виконує пошук елементів обличчя зображено у лістингу 3.1.

Лістинг 3.1 – Вилучення особливостей обличчя з кадру (файл GazeProcess.py)

```

mp_image = mp.Image(image_format=mp.ImageFormat.SRGB,
data=frame)
face_landmarker_result = landmarker.detect_for_video(mp_image,
timestamp_ms)

if face_landmarker_result.face_landmarks:
    lms_s = np.array([[lm.x, lm.y, lm.z] for lm in
face_landmarker_result.face_landmarks[0]])
    lms_2 = (lms_s[:, :2] * [frame.shape[1],
frame.shape[0]]).round().astype(int)

    mp_hor_pts = [lms_s[i] for i in OUTER_HEAD_POINTS]
    mp_ver_pts = [lms_s[i] for i in [NOSE_BRIDGE, NOSE_TIP]]
    model_hor_pts = OUTER_HEAD_POINTS_MODEL
    model_ver_pts = [NOSE_BRIDGE_MODEL, NOSE_TIP_MODEL]

```

Завдяки використанню бібліотеки MediaPipe вдалося просто і ефективно імплементувати алгоритм отримання ключових точок, що відповідають особливостям обличчя.

3.1.3 Огляд обраного методу аналізу особливостей обличчя

Етап аналізу напряму погляду є основною темою для експериментів у даному дослідженні. Для вирішення даного завдання існує чимало методів, зокрема методи на основі тривимірної проекції ока і зіниці, так і інші, на основі зовнішнього вигляду, наприклад методи з використанням моделей машинного навчання. У цій роботі основним методом буде використання методів глибокого навчання для аналізу особливостей зображення ока.

Таке завдання потребує створення, або пошук відповідного дата сету та вибір оптимальної архітектури моделі.

3.1.4 Обґрунтування вибору датасету для проведення експериментів

Вибір датасету для проведення експериментів є важливим етапом, оскільки він безпосередньо впливає на точність, ефективність та

релевантність результатів. Для даного дослідження датасет має містити два різних типи даних – зображення ока, і векторного представлення того, куди той дивиться. У якості потенційних варіантів для дата сету були переглянуті MPIIGaze та Gaze360. У нашому експерименті ми використовуємо два датасета – один – навчальний, генерований на основі програми UnityEyes, та інший – тестовий, на основі реальних зображень, зроблених при різних зовнішніх характеристиках, освітлення, та поворот голови.

Датасет на основі UnityEyes генерований за допомогою відповідної програми [13]. Генеровані зображення повертають зображення очей та json-файл, у якому відображено характеристики отриманих зображень ока. Приклад зображень, що генеруються даною програмою, можна переглянути на рисунку 3.3



Рисунок 3.3 – Зображення, що генеруються програмою UnityEyes

Для ефективнішого ходу експерименту, ми збережемо усі зображення у відповідний файл `eye_img.npy` – файл, що зберігає масиви зображень, а дані цих зображень, що нас цікавлять – у інший файл – `eye_ru.npy` – файл, що зберігає нахил ока. Дані про нахил ока зберігаються у вигляді двох значень,

що відповідають повороту ока у радіанах по різних осях.

Отриманий датасет має розмір у 1,04 гігабайт та містить 129285 зображень, кожне з яких має розміри 60 на 60 і представлено у єдиному каналі – тобто градації сірого. Дані цього дата сету пройшли попередню обробку, зокрема конвертація у зображення градацій сірого, та зміна розміру. Основним типом цільової змінної є значення повороту ока у радіанах.

Тестовий датасет створено на основі обробленого відео зображення, і він ідентично оброблений щоб бути за основними параметрами до дата сету на основі UnityEyes. Отриманий датасет має розмір у 59 мегабайт та містить 1163 зображень, кожне з яких має розміри 60 на 60 і представлено у єдиному каналі – тобто градації сірого. На рисунку 3.4 наведено приклад тестових зображень.



Рисунок 3.4 – приклад зображень, що зустрічаються у датасеті

У даному завданні замість класів використовується регресія. Використання регресії передбачає знаходження безперервних величин, таких як координати погляду на екрані або кути напрямку погляду. Отриманий датасет є збалансованим, що для моделі, що використовує регресію, означає рівномірне представлення цільової змінної в ключових діапазонах значень. Підхід з використанням регресії використовується у завданнях трекінгу погляду і, саме тому використовується у даній роботі.

3.2 Структура проекту

Для імплементації модулю оцінки погляду користувача, створимо послідовність операцій, необхідних для перетворення зображень на потрібні нам дані. Діаграма цих операцій зображена на рисунку 3.5.

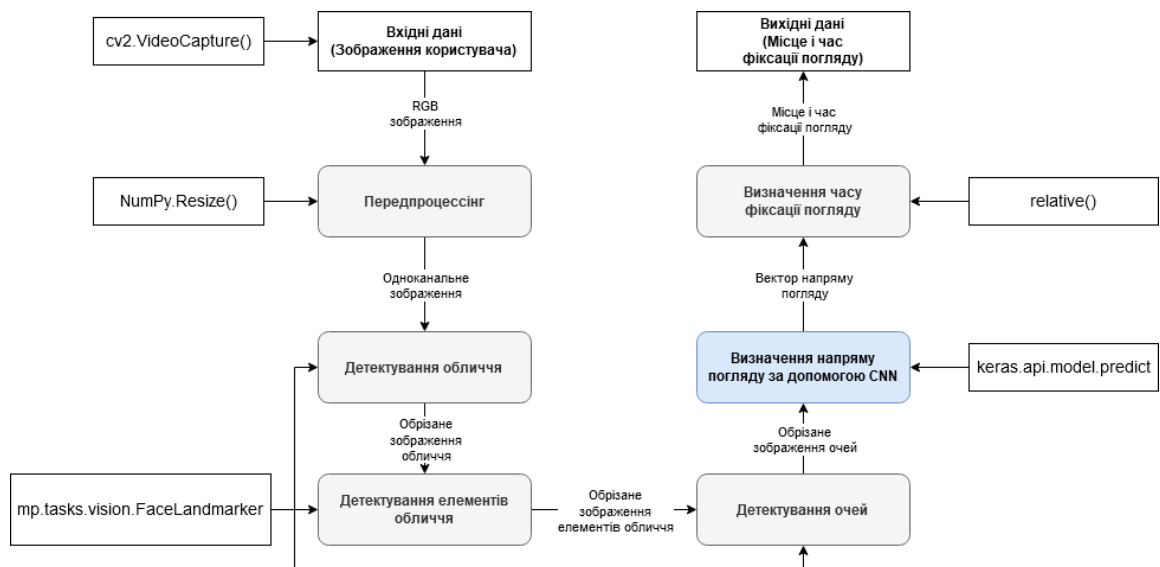


Рисунок 3.5 – Конвеєр обробки зображень

Дана модель приймає вхідні дані у якості зображення користувача. Дане зображення конвертується у одно каналне зображення за допомогою рішень бібліотек NumPy, а саме при використанні функції `resize()`. Даний вибір обумовлений ефективністю роботи даної функції для завдань щодо зміни особливостей зображень. На вихід передпроцесор подає оброблене зображення. На отриманому зображенні виконується знаходження елементів обличчя і пошук очей за допомогою інструментів бібліотеки `MediaPipe`, зокрема, за допомогою вбудованого інструмента знаходження точок обличчя, як це показано у підрозділі 3.1.2. Отримані дані зображення очей направляються до конвулюційної нейронної мережі, що знаходить вектор на основі зображення.

Використання конвулюційних нейронних мереж (CNN) для оцінки

погляду користувача є виправданим через те, що CNN можуть автоматично виділяти важливі ознаки на різних рівнях абстракції, що важливо для аналізу зображень очей. Вони здатні розпізнавати такі деталі, як форма ока, положення зіниці, райдужки та інші характеристики, необхідні для точної оцінки напрямку погляду. CNN має адаптуватися до різних умов, таких як зміни освітлення, кути зору чи вирази обличчя за рахунок ієрархії обробки ознак. Це досягається завдяки аналізу простіших деталей зображення на верхніх шарах, і складніших на більш глибоких. Також важливо, що CNN здатні моделювати складні залежності, що дозволяє точно аналізувати напрям погляду за наявності таких додаткових параметрів як, орієнтація голови чи відстань до камери. Вони також можуть ефективно працювати з великими наборами даних, що дозволяє враховувати варіативність користувачів, їх індивідуальні особливості. Тому, оскільки мережі CNN забезпечують високу точність, здатність до адаптації до різних умов і можливість обробляти велику кількість даних, вибір використання конвулюційних нейронних мереж для етапу визначення напрямку погляду, цілком виправдане. Наступним етапом є місце визначення фіксації напрямку погляду. Цей етап полягає у знаходженні місця, у яке дивиться користувач на екрані і реалізується завдяки співвідношенню вектора погляду до місця на екрані. Здебільшого, дане завдання виконується через серію перетворень, що виконані за допомогою інструментів бібліотеки NumPy та OpenCV. Для виділення часу, додатково використовується бібліотека Time

Даний проект включає використання різних моделей, що повинні навчатися за одним і тим самим даним сетом. Моделі відрізняються між собою за архітектурою і загалом у цьому дослідженні представлені п'ять різних архітектур, серед яких наступні:

- LeNet;
- AlexNet;
- VGGNet;
- ResNet.

Кожна з перерахованих моделей є тим, чи іншим різновидом конволюційних нейронних мереж (CNN). CNN мережі відзначаються своєю здатністю автоматично виявляти складні ознаки та структури в зображеннях, що робить їх незамінними для таких задач, як класифікація зображень, сегментація, розпізнавання об'єктів і чимало інших.

Основною особливістю CNN є використання конволюційних шарів, які дозволяють мережі аналізувати зображення через фільтри. Ці фільтри виділяють локальні ознаки, такі як краї, текстури або патерни, що потім використовуються для розпізнавання більш складних структур у нижчих шарах мережі. Завдяки цьому, CNN моделі здатні досягати високих результатів у задачах, пов'язаних із зображеннями. Важливим аспектом CNN є їх здатність до самостійного навчання, тобто мережа сама виявляє важливі патерни в даних, без необхідності ручного налаштування специфічних ознак.

3.2.1 LeNet

LeNet – це одна з перших архітектур глибоких конволюційних нейронних мереж, запропонована Яном Лекуном у 1998 році. Вона була спеціально розроблена для розпізнавання рукописних цифр. Ця архітектура продемонструвала ефективність конволюційних шарів для виділення ознак і автоматизації процесу розпізнавання.

Лістинг 3.2 – Структура LeNet

```
image_input = Input((120, 120, 3))
x = Conv2D(8, (5, 5), activation= 'relu')(image_input)
x = MaxPool2D(pool_size=(2, 2), padding='valid')(x)
x = Conv2D(16, (5, 5), activation= 'relu')(image_input)
x = MaxPool2D(pool_size=(2,2),strides=(2,2), padding='valid')(x)
x = Conv2D(120, (5, 5), activation= 'relu')(image_input)
x = Flatten()(x)
x = Dense(64, activation= 'relu')(x)
output = Dense(2, activation='sigmoid')(x)
```

Структурно LeNet складається з чергування конволюційних шарів, шарів пулінгу та повнозв'язних шарів. У цій архітектурі передбачено поступове зменшення розмірів вхідних даних і одночасне збільшення кількості каналів, що забезпечує створення узагальненого представлення зображення. Імплементовану структуру LeNet можна переглянути у лістингу 3.2.

Серед переваг архітектури LeNet можна виділити те, що вона відзначається простотою, що дозволяє простіше впроваджувати та аналізувати її. Мережа демонструє високу стійкість до спотворень у даних. Однак архітектура має і певні обмеження. Вона не здатна обробляти складні або великі зображення, що робить її непридатною для задач із високою варіативністю вхідних даних.

3.2.2 AlexNet

AlexNet – конволюційна нейронна мережа, яка стала подальшим розвитком архітектури LeNet. Ця модель показала, як глибокі нейронні мережі можуть успішно працювати з великими обсягами даних і складними задачами, використовуючи графічний процесор. Архітектура AlexNet відрізняється більшою глибиною, що дозволяє виділяти багаторівневі ознаки із зображень, поступово переходячи від простих локальних образів до складних абстракцій. Вона, подібно до LeNet має структуру, засновану на чергуванні шарів, але у порівнянні з LeNet має більш глибоку структуру. Імплементовану структуру AlexNet можна переглянути у лістингу 3.3

Лістинг 3.3 – Структура AlexNet

```
model = keras.Sequential()
model.add(layers.Conv2D(filters=64, kernel_size=(11, 11),
strides=(4, 4), activation="relu", input_shape=(240, 240, 1)))
model.add(layers.BatchNormalization())
model.add(layers.MaxPool2D(pool_size=(3, 3), strides=(2, 2)))
model.add(layers.Conv2D(filters=128, kernel_size=(5, 5),
strides=(1, 1), activation="relu",
padding="same"))
```

```

model.add(layers.BatchNormalization())
model.add(layers.MaxPool2D(pool_size=(3, 3), strides=(2, 2)))
model.add(layers.Conv2D(filters=256, kernel_size=(3, 3),
                        strides=(1, 1), activation="relu",
                        padding="same"))
model.add(layers.BatchNormalization())
model.add(layers.Conv2D(filters=256, kernel_size=(3, 3),
                        strides=(1, 1), activation="relu",
                        padding="same"))
model.add(layers.BatchNormalization())
model.add(layers.Conv2D(filters=512, kernel_size=(3, 3),
                        strides=(1, 1), activation="relu",
                        padding="same"))
model.add(layers.BatchNormalization())
model.add(layers.MaxPool2D(pool_size=(3, 3), strides=(2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(1024, activation="relu"))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(10, activation="softmax"))

```

AlexNet довела, що глибокі моделі з великою кількістю параметрів можуть демонструвати високі результати за умови достатньої кількості даних і потужних обчислювальних ресурсів.

3.2.3 VGGNet

VGGNet – це архітектура конволюційних нейронних мереж, розроблених дослідниками з Оксфорду у 2014 році. Важливою особливістю даної архітектури є використання великої кількості конволюційних шарів із невеликими ядрами згортки для зменшення розмірів просторових ознак. Імплементовану структуру VGGNet можна переглянути у лістингу 3.4

Лістинг 3.4 – Структура VGGNet

```

model = Sequential()
    model.add(Conv2D(64, (3, 3), activation='relu',
padding='same', input_shape=(240, 240, 1)))
    model.add(Conv2D(64, (3, 3), activation='relu',
padding='same'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Conv2D(128, (3, 3), activation='relu',
padding='same'))

```

```

    model.add(Conv2D(128, (3, 3), activation='relu',
padding='same'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Conv2D(256, (3, 3), activation='relu',
padding='same'))
    model.add(Conv2D(256, (3, 3), activation='relu',
padding='same'))
    model.add(Conv2D(256, (3, 3), activation='relu',
padding='same'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Conv2D(512, (3, 3), activation='relu',
padding='same'))
    model.add(Conv2D(512, (3, 3), activation='relu',
padding='same'))
    model.add(Conv2D(512, (3, 3), activation='relu',
padding='same'))
    model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
    model.add(Flatten())
    model.add(Dense(1024, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(1024, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(2, activation='softmax'))

```

Однією з головних переваг цієї архітектури є її простота і логічна організація, що сприяє легкому розумінню та реалізації. Невеликі ядра згортки дозволяють виділяти детальні локальні ознаки, а глибина мережі забезпечує високу якість розпізнавання складних об'єктів. Завдяки стандартній структурі, VGGNet стала основою для багатьох інших моделей та легко адаптується до різних задач. Попри численні переваги, VGGNet має значні недоліки. Серед недоліків, основним є велика кількість параметрів, яка спричиняє високі обчислювальні витрати.

3.2.4 ResNet

ResNet – це архітектура конволюційних нейронних мереж, яка була запропонована для вирішення проблеми затухаючого градієнта при тренуванні дуже глибоких мереж. Основною інновацією ResNet є використання залишкових зв'язків, які дозволяють передавати інформацію через шари без змін. Імплементовану структуру ResNet можна переглянути у лістингу 3.5

Лістинг 3.5 – Структура ResNet

```

image_input = Input((120, 120, 1))
x = Conv2D(64, 5, strides=2, activation='relu')(image_input)
x = MaxPool2D(pool_size=(2, 2), strides=(1,1),
padding='valid')(x)
for i in range(4):
    if i == 0:
        for j in range(block_layers[i]):
            x = identity_block(x, filter_size)
    else:
        filter_size = filter_size*2
        x = convolutional_block(x, filter_size)
        for j in range(block_layers[i] - 1):
            x = identity_block(x, filter_size)
x = Flatten()(x)
x = Dense(1024, activation='relu')(x)
x = Dense(256, activation='relu')(x)
output = Dense(2, activation='sigmoid')(x)

```

Структура ResNet будується на основі простих блоків – резидентних блоків. Кожен такий блок складається з двох або більше конволюційних шарів. Мережа починається з початкового шару, що виконує згортку з великим фільтром для виділення базових ознак зображення, після чого відбувається серія резидентних блоків. Після послідовності блоків мережа завершується шаром глобального середнього пудингу, і, повнозв'язних шарів. Структура ResNet суттєво спрощує навчання глибоких мереж, оскільки завдяки використанню залишкових зв'язків знижує ймовірність виникнення проблем із градієнтами, що дозволяє досягати високих результатів у вирішенні різних завдань комп'ютерного зору.

4 ПЛАНУВАННЯ ТА ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ

Відповідно до того, як було визначено процес проведення експериментів, ми перевіримо, як на зміну параметрів моделі змінюється результати у даному завданні. Від правильного вибору та налаштування параметрів залежать ефективність, точність і продуктивність системи. Серед параметрів, які впливають ефективність роботи мережі можна виділити наступні:

- параметри архітектури моделі;
- гіперпараметри навчання;
- параметри обробки даних.

Параметри архітектури моделі – це параметри і атрибути що визначають структуру моделі, таку як кількість шарів, типи шарів, кількість нейронів у кожному шарі, або розмір фільтрів у конволюційних шарах. Гіперпараметри навчання, у свою чергу, це параметри, що використовуються при навчанні моделі, такі як швидкість навчання, розмір батчу, кількість епох або коефіцієнти регуляризації. Гіперпараметри встановлюються вручну або автоматизованими методами перед початком навчання і не змінюються в цьому процесі. У даній роботі, здебільшого ми будемо проводити експерименти саме з гіперпараметрами навчання, однак частково ми звернемося і до параметрів архітектури моделі.

4.1 Затвердження базових параметрів мережі

Серед основних параметрів моделі, взаємозаміну яких ми будемо досліджувати є наступні:

- загальна архітектура;
- кількість епох навчання;
- розмір батчу;

- оптимізатор;
- функція активації прихованих слоїв;
- розподіл даних сету.

Перш ніж проводити заміри результатів навчання моделей при різних показниках, ми проведемо замір результатів при базових параметрах мережі.

Оберемо у якості базових параметрів моделі наступні:

- архітектура: LeNet
- кількість епох: 10;
- розмір батчу: 16;
- оптимізатор: SGD;
- функція активації: Relu;
- розподіл даних сету: 70% : 15% : 15%.

Тестова точність розпізнавання напряму погляду, яка досягнута на даній архітектурі становить 79% при значенні втрат 19%. В подальших експериментах, ми спробуємо покращити цей результат шляхом змін базових параметрів і документування отриманих результатів. Для цього виділимо основні особливості, які будемо перевіряти.

Кількість епох навчання – це значення, що визначає скільки разів модель проходить через весь набір даних під час навчання. Занадто мала кількість епох призводить до неповного навчання, тоді як занадто велика кількість епох призведе до перенавчання, коли модель добре працює на навчальних даних, але втрачає здатність узагальнювати нові дані. Оптимальна кількість епох залежить від складності задачі, розміру даних та архітектури моделі. Для проведення експерименту, ми розподілимо епохи навчання по етапам наступним чином: 1:5:10:15:25:50

Розмір батчу визначає кількість прикладів із тренувального набору даних, які обробляються моделлю за один крок оптимізації. Це ключовий параметр, який впливає на ефективність навчання і здатність моделі до генералізації. Для проведення експерименту, ми розподілимо розміри батчу по етапам наступним чином: 16:32:64:128.

Оптимізатор – це алгоритм, який керує процесом оновлення ваг моделі з метою мінімізації функції втрат. Він визначає, як і з якою швидкістю модель навчається, а також впливає на швидкість збіжності та здатність до узагальнення. Для проведення експерименту, ми обрали наступні оптимізатори:

- SGD;
- Adam;
- AdamW.

Функція активації визначає, як вихід кожного нейрона в моделі обчислюється та передається далі. Вона вводить нелінійність у нейронну мережу, що дозволяє їй моделювати складні залежності в даних. Для проведення експерименту, ми обрали наступні функції:

- Relu;
- LeakyRelu;
- Swish;
- GELU.

Розподіл датасету на тренувальний, валідаційний та тестовий набори полягає у поділу датасету на відповідні набори даних, кожен з яких виконує своє завдання. Тренувальний набір буде використовуватися для навчання моделі, валідаційний – для налаштування гіперпараметрів та оцінки продуктивності моделі під час навчання, а тестовий – для оцінки остаточної точності моделі. Для проведення експерименту розподіл виконувався наступним чином:

- тренувальний : валідаційний : тестовий набори = 70% : 15% : 15%;
- тренувальний : валідаційний : тестовий набори = 70% : 5% : 25%;
- тренувальний : валідаційний : тестовий набори = 50% : 25% : 25%.

Архітектури мережі, з якими проводилися експерименти, мають вигляд, зображений у підрозділі 3.2. Для даного завдання ми, спочатку почнемо з усіма чотирма архітектурами, але впродовж дослідження, знайдемо архітектуру, яка при певному набору параметрів досягає

найбільшої ефективності.

4.2 Оцінка впливу гіперпараметрів моделі на метрики Accuracy, Loss

Подальші кроки полягають у зміні кількості епох навчання при використанні різних архітектур для досягнення найвищих показників точності детектування напряму погляду (accuracy) та функції втрат (loss function) для трьох різних підходів до декомпозиції датасету. Результати наведено у таблицях 4.1, 4.2, 4.3.

Таблиця 4.1 – Точність нейромережевого детектування напряму погляду на основі CNN при розподілі датасету у співвідношенні тренувальний : валідаційний : тестовий набори = 70% : 15% : 15%

Архітектура мережі CNN	Кількість епох навчання CNN	Тестова точність детектування напряму погляду	Значення функції втрат	Час навчання, секунд
1	2	3	4	5
LeNet	1	0.5102	0.3634	226
	5	0.6187	0.2431	1074
	10	0.7894	0.1919	2175
	15	0.7961	0.1872	3127
	25	0.8072	0.1820	5621
	50	0.8086	0.1812	11278
VGGNet	1	0.6097	0.3152	327
	5	0.7863	0.1940	1655
	10	0.8149	0.1862	3221
	15	0.8231	0.1823	4826
	25	0.8316	0.1790	8284
	50	0.8331	0.1779	16582

Продовження таблиці 4.1

1	2	3	4	5
AlexNet	1	0.5914	0.3218	410
	5	0.7122	0.2138	1978
	10	0.7923	0.1928	4021
	15	0.8059	0.1862	6031
	25	0.8093	0.1829	10123
	50	0.8116	0.1791	21091
ResNet	1	0.7472	0.1913	1086
	5	0.8213	0.1912	5671
	10	0.8168	0.1872	10791
	15	0.8294	0.1863	16180
	25	0.8337	0.1836	26291
	50	0.8318	0.1747	50137

Отримані результати узагальнено у вигляді рисунку 4.1.

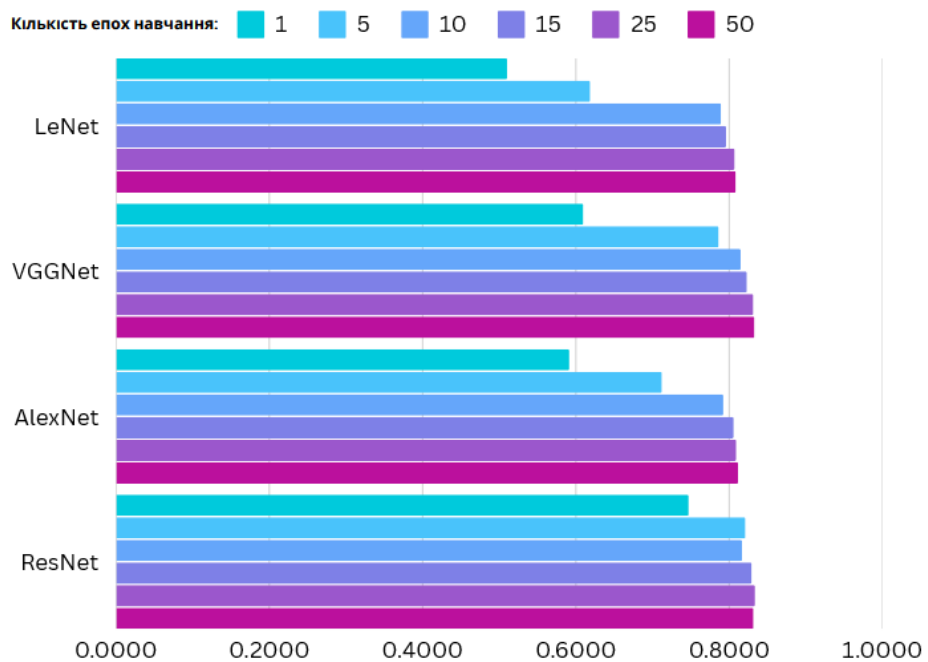


Рисунок 4.1 - Точність нейромережевого детектування при розподілі датасету у співвідношенні 70% : 15% : 15%

Як ми бачимо з результатів даного проміжку експериментів, максимальна точність наближується до значення у 83%, а найменше значення функції втрат до 17%. При цьому, найкращі результати показують архітектури VGGNet та ResNet, але ResNet – єдина архітектура, що демонструє показники перенавчання.

Таблиця 4.2 – Точність нейромережевого детектуваннянапрямупогляду на основі CNN при розподілі датасету у співвідношенні тренувальний : валідаційний : тестовий набори = 70% : 5% : 25%

Архітектура мережі CNN	Кількість епох навчання CNN	Тестова точність детектування на пряму погляду	Значення функції втрат	Час навчання, секунд
1	2	3	4	5
LeNet	1	0.5312	0.2461	206
	5	0.7635	0.1980	970
	10	0.7729	0.1997	2029
	15	0.7892	0.1869	3071
	25	0.8056	0.1833	5272
	50	0.8116	0.1767	10721
VGGNet	1	0.6390	0.2094	294
	5	0.7696	0.1970	1456
	10	0.8080	0.1888	2927
	15	0.8203	0.1836	4129
	25	0.8327	0.1785	6221
	50	0.8332	0.1772	11957
AlexNet	1	0.6732	0.2931	407
	5	0.7271	0.2312	2047
	10	0.7923	0.1938	4146
	15	0.7831	0.1883	6391
	25	0.8025	0.1846	10724
	50	0.8067	0.1789	20814

Продовження таблиці 4.2

1	2	3	4	5
ResNet	1	0.7769	0.1926	1021
	5	0.8141	0.1859	4983
	10	0.8358	0.1889	9782
	15	0.8294	0.1823	14689
	25	0.8385	0.1798	23823
	50	0.8352	0.1812	46218

Отримані результати узагальнено у вигляді рисунку 4.2.

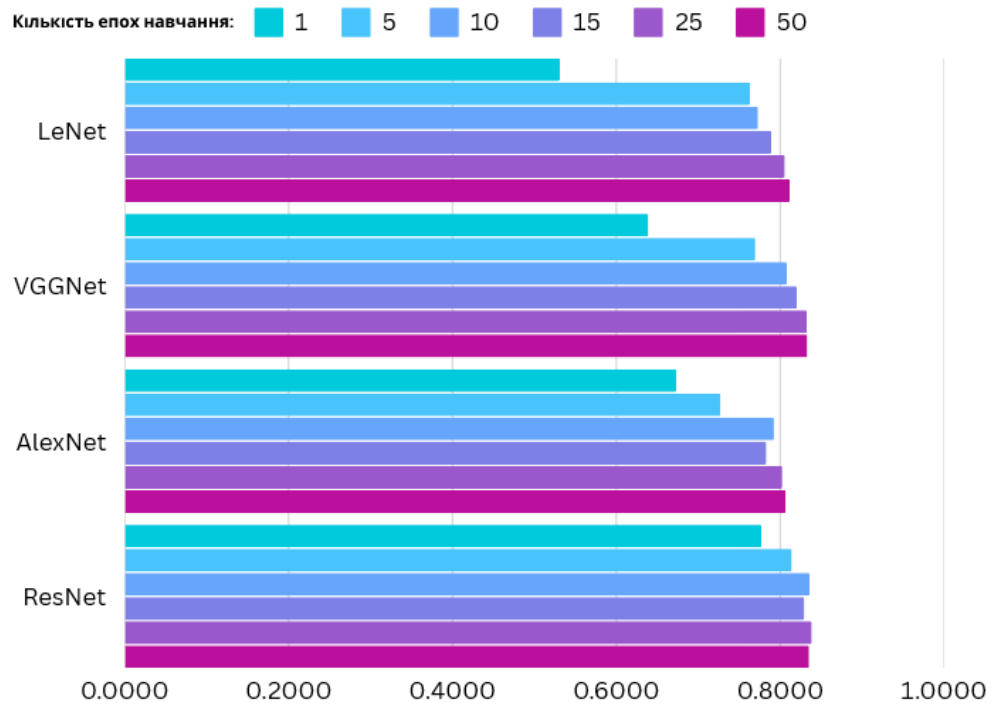


Рисунок 4.2 - Точність нейромережевого детектування при розподілі датасету у співвідношенні 70% : 5% : 25%

Як ми бачимо з результатів даного проміжку експериментів, максимальна точність наближується до значення у 84%, а найменше значення функції втрат до 17%. При цьому, найкращі результати показують архітектури VGGNet та ResNet.

Таблиця 4.3 – Точність нейромережевого детектуваннянапрямупогляду на основі MTCNN при розподілі датасету у співвідношенні тренувальний : валідаційний : тестовий набори = 50% : 25% : 25%

Архітектура мережі CNN	Кількість епох навчання CNN	Тестова точність детектування на прямому погляді	Значення функції втрат	Час навчання, секунд
LeNet	1	0.5852	0.2530	220
	5	0.7641	0.1975	813
	10	0.7777	0.1943	1638
	15	0.7751	0.1846	2516
	25	0.7963	0.1868	4616
	50	0.8121	0.1767	10721
VGGNet	1	7053	0.2530	238
	5	0.7696	0.1975	1230
	10	0.8151	0.1874	2518
	15	0.8332	0.1926	3731
	25	0.8314	0.1797	6727
	50	0.8348	0.1782	12530
AlexNet	1	0.6217	0.3218	410
	5	0.7122	0.2138	1978
	10	0.7923	0.1928	4021
	15	0.8059	0.1862	6031
	25	0.8093	0.1829	10123
	50	0.8146	0.1791	21091
ResNet	1	0.6969	0.2135	896
	5	0.8080	0.1822	4217
	10	0.8234	0.1769	9180
	15	0.8330	0.1723	13261
	25	0.8392	0.1704	22624
	50	0.8386	0.1773	45178

Отримані результати узагальнено у вигляді рисунку 4.3.

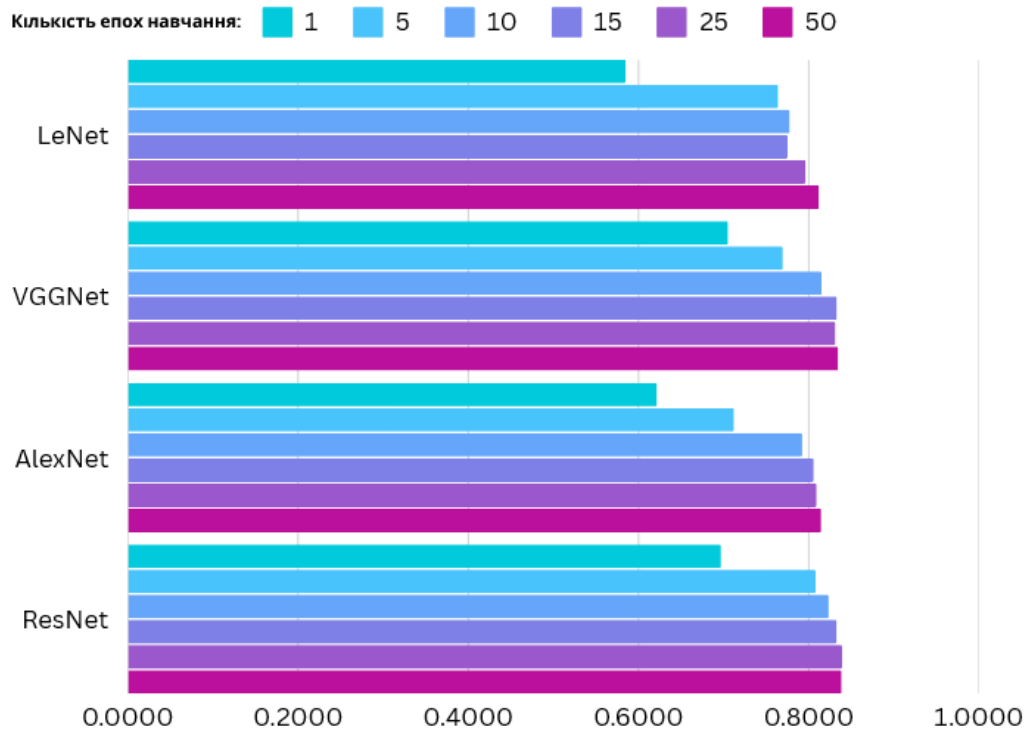


Рисунок 4.3 - Точність нейромережевого детектування при розподілі датасету у співвідношенні 50% : 25% : 25%

Як ми бачимо з результатів даного проміжку експериментів, максимальна точність наближується до значення у 84%, а найменше значення функції втрат до 17%. При цьому, найкращі результати показують архітектури VGGNet та ResNet.

По даним результатам можна зробити висновок, щонайвище значення тестової точності детектування напряму погляду та найнижче значення функції втрат вдається досягти при використанні архітектур ResNet і VGGNet, при кількості епох 25. Рекомендованим розподілом датасету для даного завдання є розподіл 70% : 15% : 15%.

Наступні дослідження, спрямовані на вибір розміру батчу, оптимізатору та активаційної функції, будуть проводитися на вище визначених гіперпараметрах.

4.3 Оцінка впливу розміру батчу, оптимізатору та активаційної функції на точність детектування наряду погляду, втрати точності та час навчання

Наступним етапом дослідження є оцінка впливу розміру батчу, вибору оптимізатору та активаційної функції на тестову точність детектування наряду погляду та значення функції втрат при фіксованих значеннях гіперпараметрів нейромережевої моделі. Ми будемо оцінювати дві різні моделі у умовах зміни активаційних функцій, оптимізаторів і розміру батчу, а саме ми будемо використовувати VGGNet та ResNet для подальших експериментів, як ті, що показали себе найкраще у заданій задачі.

4.3.1 Оцінка впливу змін параметрів на VGGNet

Змінимо такі параметри як розмір батчу, оптимізатор та функція активації прихованих слоїв для досягнення найвищих показників точності детектування наряду погляду та функції втрат. Результати наведено у таблицях 4.4 - 4.7.

Таблиця 4.4 – Точність нейромережевого детектування наряду погляду для архітектури VGGNet при кількості епох навчання у 25 та при розмірі батчу, що дорівнює 16

Активаційна функція	Оптимізатор	Тестова точність детектування наряду погляду	Значення функції втрат	Час навчання
1	2	3	4	5
ReLU	SGD	0.8327	0.1785	3116
	Adam	0.8241	0.1841	3213
	AdamW	0.7382	0.2041	3138
Leaky ReLU	SGD	0.8341	0.1841	2931
	Adam	0.8183	0.1786	3174
	AdamW	0.8515	0.1573	3625

Продовження таблиці 4.4

1	2	3	4	5
Swish	SGD	0.8027	0.1731	3014
	Adam	0.5349	1.0029	3637
	AdamW	0.8231	0.1784	3208
GELU	SGD	0.8198	0.1801	3281
	Adam	0.8120	0.1787	3342
	AdamW	0.7898	0.1826	3842

На даному етапі було отримано один з найкращих результатів навчання моделей, а саме – при комбінації функції Leaky ReLU та оптимізатора AdamW. Це результат при якому максимальна точність наближується до значення у 85%, а найменше значення функції втрат до 15,7%. Також, виникла помилка навчання при комбінації функції Swish та оптимізатора Adam, при якій значення як тренувальної, так і валідаційної точності застигло на одному і тому самому значенні, тобто модель при даних параметрах не навчалася.

Таблиця 4.5 – Точність нейромережевого детектування напряму погляду для архітектури VGGNet при кількості епох навчання у 25 та при розмірі батчу, що дорівнює 32

Активацийна функція	Оптимізатор	Тестова точність детектування напряму погляду	Значення функції втрат	Час навчання
1	2	3	4	5
ReLU	SGD	0.7911	0.1721	2382
	Adam	0.8217	0.1604	2530
	AdamW	0.8085	0.1626	2680
Leaky ReLU	SGD	0.7698	0.1820	2345
	Adam	0.7931	0.1782	2316
	AdamW	0.8191	0.1701	2687

Продовження таблиці 4.5

1	2	3	4	5
Swish	SGD	0.7974	0.1731	2672
	Adam	0.7351	0.1926	2741
	AdamW	0.7618	0.1877	2639
GELU	SGD	0.8416	0.1601	2984
	Adam	0.8120	0.1741	3278
	AdamW	0.8149	0.1746	3474

На даному етапі найкращий результат виник при комбінації функції GELU та оптимізатора SGD. Це результат при якому максимальна точність наближується до значення у 84%, а найменше значення функції втрат до 16%.

Таблиця 4.6 - Точність нейромережевого детектування напряму погляду для архітектури VGGNet при кількості епох навчання у 25 та при розмірі батчу, що дорівнює 64

Активацийна функція	Оптимізатор	Тестова точність детектування напряму погляду	Значення функції втрат	Час навчання
1	2	3	4	5
ReLU	SGD	0.8330	0.1638	2249
	Adam	0.8295	0.1612	2236
	AdamW	0.5340	1.0232	2631
Leaky ReLU	SGD	0.7953	0.1808	2126
	Adam	0.6449	0.1925	2316
	AdamW	0.7845	0.1844	2581
Swish	SGD	0.8010	0.1821	1926
	Adam	0.7931	0.1903	2205
	AdamW	0.8106	0.1864	2849
GELU	SGD	0.7971	0.1791	3260
	Adam	0.8145	0.1724	4071
	AdamW	0.8531	0.1526	3874

На даному етапі було отримано один з найкращих результатів навчання моделей, а саме – при комбінації функції GELU та оптимізатора SGD. Це результат при якому максимальна точність наближується до значення у 85%, а найменше значення функції втрат до 15.3%. Також, виникла помилка навчання при комбінації функції Relu та оптимізатора AdamW, при якій значення як тренувальної, так і валідаційної точності застигало на одному і тому самому значенні, тобто модель при даних параметрах не навчалася.

Таблиця 4.7 - Точність нейромережевого детектування напряму погляду для архітектури VGGNet при кількості епох навчання у 25 та при розмірі батчу, що дорівнює 128

Активувальна функція	Оптимізатор	Тестова точність детектування напряму погляду при збалансованій навчальній вибірці	Значення функції втрат	Час навчання
ReLU/Sigmoid	SGD	0.7931	0.1939	2016
	Adam	0.7368	0.2106	1825
	AdamW	0.6785	0.2541	1970
Leaky ReLU	SGD	0.8379	0.1636	1974
	Adam	0.7861	0.1825	2271
	AdamW	0.7671	0.2081	2206
Swish	SGD	0.7343	0.2094	2031
	Adam	0.5573	0.2803	2412
	AdamW	0.6195	0.2631	2356
GELU	SGD	0.6319	0.2028	3274
	Adam	0.6408	0.1967	3561
	AdamW	0.6571	0.1983	3418

На даному етапі найкращий результат виник при комбінації функції Leaky ReLU та оптимізатора SGD. Це результат при якому максимальна

точність наближується до значення у 83%, а найменше значення функції втрат до 16%. На даному етапі значно впали загальні результати, що наводить на ідею про те, що у при такому розмірі батчу дана модель не може узагальнювати дані з тією самою ефективністю як при меншому розмірі. Отримані результати узагальнено у вигляді рисунку 4.4.

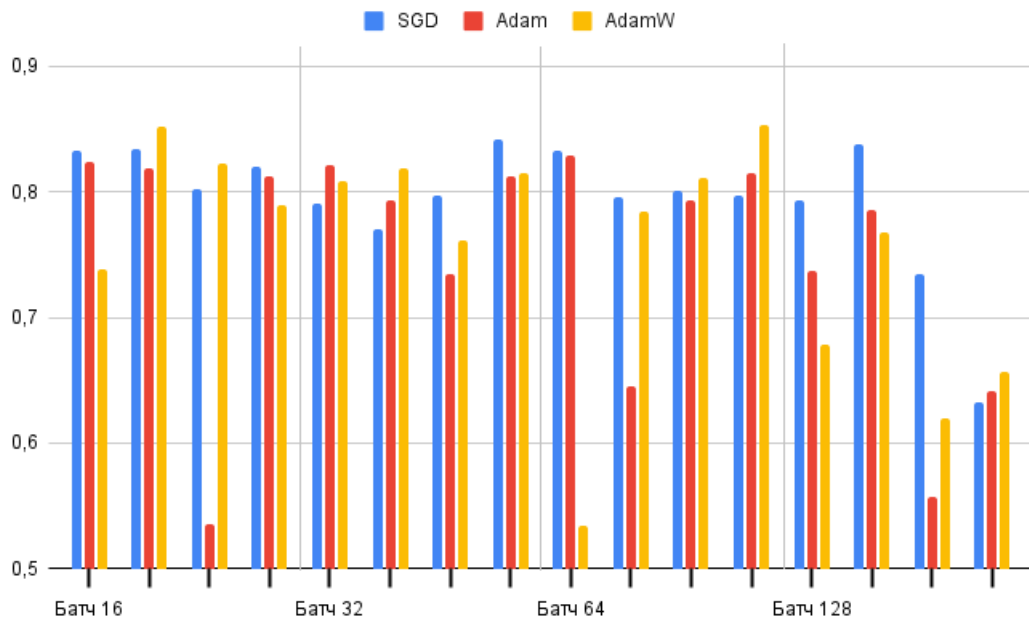


Рисунок 4.4 - Точність неймережевого детектування при використанні архітектури VGGNet

По даним результатам можна зробити висновок, щонайвище значення тестової точності детектування напряму погляду та найнижче значення функції втрат досягти при використанні архітектури VGGNet вдається при кількості епох 25 та комбінації функції GELU та оптимізатора SGD. Це результат при якому максимальна точність наближується до значення у 85%, а найменше значення функції втрат до 15.3%.

4.3.2 Оцінка впливу змін параметрів на ResNet

Змінимо такі параметри як розмір батчу, оптимізатор та функція активації прихованих слоїв для досягнення найвищих показників точності детектування на пряму погляду та функції втрат. Результати наведено у таблицях 4.8, 4.9, 4.10, 4.11.

Таблиця 4.8 – Точність нейромережевого детектування на пряму погляду для архітектури ResNet при кількості епох навчання у 25 та при розмірі батчу, що дорівнює 16

Активацийна функція	Оптимізатор	Тестова точність детектування на пряму погляду	Значення функції втрат	Час навчання
ReLU	SGD	0.8392	0.1704	22624
	Adam	0.8446	0.1609	25741
	AdamW	0.8378	0.1727	26909
Leaky ReLU	SGD	0.8136	0.1782	22831
	Adam	0.8253	0.1706	26003
	AdamW	0.8227	0.1732	27838
Swish	SGD	0.8498	0.1597	23016
	Adam	0.8518	0.1491	26361
	AdamW	0.8612	0.1562	28187
GELU	SGD	0.8362	0.1727	23179
	Adam	0.8318	0.1698	26503
	AdamW	0.8398	0.1731	28380

На даному етапі найкращий результат виник при комбінації функції Swish та оптимізатора AdamW. Це результат при якому максимальна точність наближується до значення у 85%, а найменше значення функції втрат до 15,4%.

Таблиця 4.9 – Точність нейромережевого детектування напряму погляду для архітектури ResNet при кількості епох навчання у 25 та при розмірі батчу, що дорівнює 32

Активаційна функція	Оптимізатор	Тестова точність детектування напряму погляду	Значення функції втрат	Час навчання
ReLU	SGD	0.8331	0.1702	21912
	Adam	0.8389	0.1668	25741
	AdamW	0.8218	0.1786	26102
Leaky ReLU	SGD	0.8149	0.1817	21942
	Adam	0.8164	0.1824	24751
	AdamW	0.8191	0.1842	26316
Swish	SGD	0.8304	0.1733	22193
	Adam	0.8409	0.1606	25863
	AdamW	0.8357	0.1702	26847
GELU	SGD	0.7944	0.1832	22017
	Adam	0.8221	0.1741	25813
	AdamW	0.8215	0.1761	27141

На даному етапі найкращий результат виник при комбінації функції Swish та оптимізатора Adam. Це результат при якому максимальна точність наближується до значення у 84%, а найменше значення функції втрат до 16%.

На даному етапі найкращий результат виник при комбінації функції Relu та оптимізатора Adam. Це результат при якому максимальна точність наближується до значення у 84%, а найменше значення функції втрат до 17%.

Таблиця 4.10 – Точність нейромережевого детектування напряму погляду для архітектури ResNet при кількості епох навчання у 25 та при розмірі батчу, що дорівнює 64

Активаційна функція	Оптимізатор	Тестова точність детектування напряму погляду	Значення функції втрат	Час навчання
ReLU	SGD	0.8306	0.1740	21273
	Adam	0.8380	0.1726	25061
	AdamW	0.8261	0.1793	25757
Leaky ReLU	SGD	0.8206	0.1821	21026
	Adam	0.8276	0.1893	24159
	AdamW	0.7849	0.2058	25630
Swish	SGD	0.8318	0.1706	21461
	Adam	0.8362	0.1747	25046
	AdamW	0.7886	0.1974	25917
GELU	SGD	0.8313	0.1796	20963
	Adam	0.8276	0.1831	25035
	AdamW	0.7853	0.1972	25931

Таблиця 4.11 – Точність нейромережевого детектування напряму погляду для архітектури ResNet при кількості епох навчання у 25 та при розмірі батчу, що дорівнює 128

Активаційна функція	Оптимізатор	Тестова точність детектування погляду, %	Значення функції втрат, %	Час навчання, секунд
1	2	3	4	5
ReLU/Sigmoid	SGD	0.7799	0.2037	20254
	Adam	0.8086	0.1893	24104
	AdamW	0.7941	0.1942	24415

Продовження таблиці 4.11

1	2	3	4	5
Leaky ReLU	SGD	0.8058	0.1886	20176
	Adam	0.7941	0.1951	23412
	AdamW	0.8148	0.1843	24046
Swish	SGD	0.7882	0.1939	20278
	Adam	0.8135	0.1856	23741
	AdamW	0.7841	0.1939	24345
GELU	SGD	0.8146	0.1843	19356
	Adam	0.7893	0.1939	24416
	AdamW	0.7787	0.2012	24296

На даному етапі найкращий результат виник при комбінації функції GELU та оптимізатора SGD. Це результат при якому максимальна точність наближується до значення у 82%, а найменше значення функції втрат до 18%. Отримані результати узагальнено у вигляді рисунку 4.4.

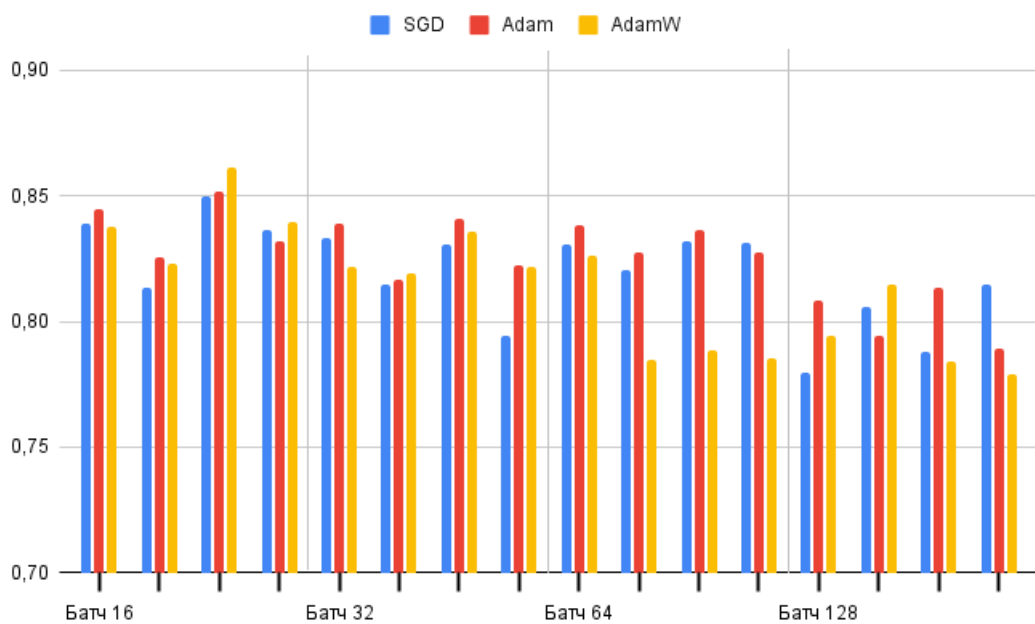


Рисунок 4.5 - Точність неймережевого детектування при використанні архітектури ResNet

По даним результатам можна зробити висновок, щонайвище значення тестової точності детектування напряду погляду та найнижче значення функції втрат досягти при використанні архітектури ResNet вдається при кількості епох 25 та комбінації функції Swish та оптимізатора AdamW. Це результат при якому максимальна точність наближується до значення у 85%, а найменше значення функції втрат до 15.4%.

4.3 Оцінка адаптації налаштованої нейромережевої моделі до зміни зовнішніх параметрів, як зміна кута нахилу голови, відстані від камери та освітлення

Проаналізувавши результати експериментів, що зведені у розділі 4.2, виділимо дві налаштовані моделі з параметрами, які дають найкращі результати і проведемо перевірку отриманих моделей використовуючи дата сет, оснований на реальних зображеннях, що створені при різних особливостях навколишнього середовища.

У якості моделей оберемо дві, серед яких модель з архітектурою VGGNet має наступні параметри:

- кількість епох: 25;
- розмір батчу: 64;
- оптимізатор: SGD;
- функція активації: GELU;
- розподіл дата сету: 70% : 15% : 15%.

Результати оцінки даної моделі наведено у таблиці 4.12

Отримані результати, наведені у таблиці 4.12, указують, що результати значно залежать як від освітлення, так і від відстані до камери. Найбільш впливовим параметром, який відображається на зниженні значення детектування напряду погляду, є освітлення. Другим по впливу параметром є відстань до камери.

Таблиця 4.12 – Результати перевірки моделі VGGNet на основі реальних зображень

Гарне освітлення (300 люменів на м ²)						
	Відстань від камери = 30			Відстань від камери = 100		
	Кут нахилу = 0°	Кут нахилу = 15°	Кут нахилу = 30°	Кут нахилу = 0°	Кут нахилу = 15°	Кут нахилу = 30°
Точність, %	0.8151	0.8134	0.7913	0.7818	0.7282	0.7139
Значення функції втрат, %	0.1813	0.1884	0.2017	0.1961	0.2184	0.2131
Час оцінки, с	471	464	473	484	467	472
Погане освітлення (30 люменів на м ²)						
	Відстань від камери = 30			Відстань від камери = 100		
	Кут нахилу = 0°	Кут нахилу = 15°	Кут нахилу = 30°	Кут нахилу = 0°	Кут нахилу = 15°	Кут нахилу = 30°
Точність, %	0.7951	0.7814	0.7847	0.6713	0.6927	0.6876
Значення функції втрат, %	0.2004	0.1916	0.1932	0.2532	0.2476	0.2580
Час оцінки, с	481	462	475	468	469	474

Наступним етапом є перевірка моделі з архітектурою ResNet має наступні параметри:

- кількість епох: 25;
- розмір батчу: 16;
- оптимізатор: AdamW;
- функція активації: Swish;
- розподіл даних: 70% : 15% : 15%.

Результати оцінки даної моделі наведено у таблиці 4.12

Таблиця 4.13 – Результати перевірки моделі ResNet на основі реальних зображень

Гарне освітлення (300 люменів на м ²)						
	Відстань від камери = 30			Відстань від камери = 100		
	Кут нахилу = 0°	Кут нахилу = 15°	Кут нахилу = 30°	Кут нахилу = 0°	Кут нахилу = 15°	Кут нахилу = 30°
Точність, %	0.8380	0.8264	0.8209	0.8063	0.7937	0.7962
Значення функції втрат, %	0.1726	0.1791	0.1786	0.1916	0.1986	0.1949
Час оцінки, с	532	561	568	541	545	559
Погане освітлення (30 люменів на м ²)						
	Відстань від камери = 30			Відстань від камери = 100		
	Кут нахилу = 0°	Кут нахилу = 15°	Кут нахилу = 30°	Кут нахилу = 0°	Кут нахилу = 15°	Кут нахилу = 30°
Точність, %	0.8184	0.7931	0.7946	0.7631	0.7541	0.7591
Значення функції втрат, %	0.1801	0.1935	0.1985	0.2153	0.2236	0.2209
Час оцінки, с	553	548	561	543	575	550

Отримані результати, наведені у таблиці 4.13, указують, що для моделі ResNet, результати значно залежать як від освітлення, так і від відстані до камери. Найбільш впливовим параметром, який відображається на зниженні значення детектування напряму погляду для моделі ResNet, є відстань до камери. Другим по впливу параметром є освітлення.

Узагальнюючи отримані результати, маємо, що найбільше на якість детектування наряду погляду впливають два параметра – освітлення і відстань до камери. Особливо сильно зменшується точність при накладенні обох параметрів одночасно. Найменш впливовим параметром виявився поворот голови.

4.4 Впровадження моделі для використання у якості підсистеми модулю оцінки наряду погляду

Для наглядного проведення тестів модулю детектування наряду погляду, створимо окремий модуль, що буде виконувати послідовність дій, відображену на рисунку 3.4. Для цього, створимо систему, що буде отримувати зображення з камери. Для цього використаємо бібліотеку OpenCV і її базовий синтаксис обробки відео, а саме клас VideoCapture. Для перед обробки.

Передпроцесінг зображення виконується за допомогою бібліотеки NumPy з використанням функції `resize`, яку ми використовуємо для зміни розмірності зображення, залишаючи лише один канал зображення.

Для знаходження обличчя на зображенні і виділення очей використовується бібліотека `Mediapipe`. `Mediapipe` є універсальним інструментом для розробки програм, пов'язаних із комп'ютерним зором та обробкою зображень у реальному часі. Використання даної бібліотеки обґрунтоване тим, що вона надає ефективні та оптимізовані алгоритми.

Отримуючи обличчя, ми використовуємо модель для передбачення вектору погляду. Це виконується з використанням бібліотеки `Keras` і функції `predict`, що дозволяє отримати дані, оброблені моделлю. Отримані дані у вигляді даних про те, куди дивиться людина, перетворюємо за допомогою методів `NumPy` у позицію на екрані, у яку приблизно користувач дивиться. Цю позицію, за допомогою інструментів `OpenCV`, перетворюємо у приблизну зону інтересу.

Отриману зону інтересу можна подальшим чином аналізувати, зокрема обраховувати час, за який користувач переглядає той, чи інший елемент, або сегмент на екрані і робити відповідні висновки про інтерес користувача до певного контенту.

Приклад використання отриманої програми можна переглянути на зображеннях 4.2-4.4



Рисунок 4.6 – Погляд поза екраном справа



Рисунок 4.7 – Погляд на екрані



Рисунок 4.8 – Погляд поза екраном зліва

Отриманий модуль аналізу напряму погляду дозволяє визначати позицію, куди дивиться користувач в реальному часі при різних параметрах зовнішнього середовища. Його застосування відкриває нові можливості для розвитку адаптивних систем управління навчанням і інших проектів в областях медицини, маркетингу, віртуальної та доповненої реальності.

ВИСНОВКИ

У рамках даної роботи з дослідження методів оцінки напряму погляду у контексті LMS, було проведено аналіз предметної області щодо впровадження системи оцінки напряму погляду у LMS для підвищення достовірності оцінювання рівня знань студентів при використанні систем дистанційного контролю знань, шляхом моніторингу та оцінки напрямку зору тестованого. На основі проаналізованих даних, було виділено мету і основні задачі для досягнення даної мети (дивись розділ 1.6).

Визначення можливостей персоналізованого використання LMS було проведено на етапі пошуку інформації, зокрема інформація про адаптивні системи управління навчанням, про які розписано у розділі 1.3.1, є рішенням, що забезпечує персоніфікованість навчального процесу при використанні LMS. Це дозволило виявити ключові параметри та інструменти, які сприяють адаптації навчального контенту до потреб окремого користувача.

Оцінка точності визначення напряму погляду на основі існуючих методів була виконана у рамках розділу 2.2. У рамках цього підрозділу було виділено основні рішення визначення напряму погляду, їх сильні та слабкі сторони. Було переглянуто особливості використання ШНМ у проблемах комп'ютерного зору і виділено основні переваги використання нейромережевих методів для аналізу напряму зору. Це, дозволило сформулювати критерії порівняння методів та вибрати підхід до виконання оцінки напряму погляду.

Було обрано систему у рамках якої завданню оцінки напряму погляду приділено відповідний модуль, що виконує аналіз зображення і передбачає, куди користувач дивиться. Було розроблено модель адаптивної системи управління навчанням з особливістю на основі аналізу тривалості фокусування та напряму погляду. Модель можна переглянути на рисунку 3.1. Розробка моделі дозволила розробити основні характеристики модулю

оцінки напряму погляду і вибрати інструменти для імплементації.

Для створення модулю, що оцінює напрям погляду, було обрано створити нейромережеве рішення на основі CNN. Для найкращого налаштування даної моделі було проведено серію експериментів з параметрами модуля оцінки напряму погляду на основі нейромережевої моделі для визначення параметрів, що найкраще підходять для виконання завдання аналізу напряму погляду. Для отриманих моделей було проведено аналіз впливу умов зйомки на точність визначення напряму погляду. Це дозволило створити модель з параметрами, що якнайкраще відповідають завданню.

Відповідно до результатів, отриманих у розділі 4, поставлені для даної роботи цілі було успішно досягнуто. А саме, отримано модуль оцінки напряму погляду, який було обрано після проведеної серії експериментів, що мали на меті знаходження найточнішого методу аналізу напрямку погляду.

Отримана система і дані цієї роботи можуть слугувати для проведення подальших досліджень аналізу напряму погляду. При цьому, дане дослідження має важливість не лише в контексті систем управління навчанням, а і в контексті завдань комп'ютерного зору та детектування напряму погляду у областях медицини, маркетингу, віртуальної та доповненої реальності.

Подальші дослідження можуть включати створення повноцінної системи управління навчанням з вбудованою системою контролю знань, у якій крім аналізу напряму погляду для оцінки, можливо впровадити більше взаємодії з зображенням користувача, такі як впровадженні методів прискореної високоточної авторизації користувача на основі розпізнавання обличчя, а також додаткові функції, такі як керування системою за допомогою погляду, забезпечуючи людям з обмеженими можливостями інтерфейс до взаємодії з навчальною системою

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Еліас Т. Універсальний дизайн навчання Moodle // Міжнародний огляд досліджень у галузі відкритого та розподіленого навчання. – 2010. – Т. 11. – №. 2. – С. 110-124
2. Барковська, О., Ляпін, Я., Музика, Т., Рундук, І., & Ботнар, П. (2024). Модель моніторингу напрямку погляду в системі комп'ютерного контролю знань. Інформаційні технології та засоби навчання, 99(1), 63–75. <https://doi.org/10.33407/itlt.v99i1.5503>
3. Ляпін Я.А., Барковська О.Ю. Методи оцінки напряму погляду людини в контексті lms. // Проблеми інформатизації. Тези доповідей дванадцятої міжнародної науково-технічної конференції. - ЧДТУ, ВА ЗС АР, УТіГН, НТУ “ХПІ”, ХНУРЕ, "ПД ПКНДІ АП", 2024. - 21 – 22 листопада 2024 року. – с.68.
4. Биков В.Ю. Моделі організаційних систем відкритої освіти / В.Ю.Биков. – Київ: Атіка, 2009. – 684 с.
5. Доннелі Р. Прикладне електронне навчання та електронне викладання у вищій освіті / Р.Доннеллі, Ф.МакСуїні. – Hershey, New York, 2009
6. Шахід Аль Нур. Запропонована архітектура хмарних обчислень для системи освіти в Бангладеш та вплив на поточну систему освіти // IJCSNS Міжнародний журнал комп'ютерних наук та мережевої безпеки, Том 10 №10, жовтень 2010. - р.7-13
7. Урош О., Ругеджі Д., Нанчовська-Щербец І., Босніч З. Поєднання моделей стилів навчання та вирішення проблеми нового користувача в навчальних рекомендаційних системах. 2016. https://www.researchgate.net/publication/308890022_Combining_Learning_Style_Models_and_Alleviating_the_New_User_Problem_in_Learning_Recommender_Systems

8. Фейсал, М. Х., Аль-Амірі, А. В., та Альсумаїт, А. А. (2015). Адаптивна система електронного навчання. 17-та міжнародна конференція. <https://doi.org/10.1145/2837185.2837249>
9. Рабіман Р., Нуртанто М., Холіфах Н. Проектування та розробка системи електронного навчання за допомогою системи управління навчанням (LMS) у професійній освіті // Online Submission. - 2020. - Т. 9. - №. 1. - С. 1059- 1063
10. Келер, Адріан, Бредскі, Гері Р.. Вивчення OpenCV 3: Комп'ютерний зір на С++ з бібліотекою OpenCV. Японія: O'Reilly Media, 2017. Хоуз, Джозеф, Мінічіно, Джо. Вивчення комп'ютерного зору OpenCV 4 з Python 3: Ознайомлення з інструментами, методами та алгоритмами для комп'ютерного зору та машинного навчання, 3-тє видання. Великобританія: Packt Publishing, 2020
11. Хасан, М. К., Ахсан, М. С., Абдулла-аль-Мамун, Н., Новаз, С. Х. С. и Лі, Г. М. (2021). Методи виявлення людського обличчя: Комплексний огляд та майбутні напрямки досліджень. Електроніка, 10(19), 2354. <https://doi.org/10.3390/electronics10192354>
12. Смарт, Марк. Вступ до науки про дані за допомогою Python: Основи Numpy та Pandas. N.p.: Amazon Digital Services LLC - KDP Print US, 2018
13. UnityEyes. *Department of Computer Science and Technology* /. URL: <https://www.cl.cam.ac.uk/research/rainbow/projects/unityeyes/> (date of access: 24.07.2024).