

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Система розпізнавання зображень на базі методів, заснованих на автоенкодері

Назва теми

КвРКІ. 200244.20.02.22 ПЗ

Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

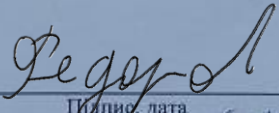
Виконав: студент IV курсу, група KI2-20-2


Підпис

Є. В. Семенюк

Ініціали, прізвище

Керівник


Підпис, дата

Є. Є. Федоров

Ініціали, прізвище

Нормоконтролер


Підпис, дата

С.М. Лисенко

Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем


Підпис

Т.О. Говоруценко

Ініціали, прізвище

«21» червня 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О. Говорущенко

“ 10 ” 01 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Семенюку Євгенію Валерійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Система розпізнавання зображень на базі методів, заснованих на автоенкодері

Керівник проекту (роботи) Федоров С. С., д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 15.02.2024 р. № 8

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Аналіз та огляд штучних нейронних мереж

Дослідження методів та аналіз автоенкодеру

Використання автоенкодеру для розпізнавання об'єктів у фото





5. Перелік графічного матеріалу (із визначенням обов'язкових креслень) _____

UML Схема програмного забезпечення

Зображення архітектур нейромереж автоенкодерів

Двійкове арифметичне кодування картичак

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІС		
Антиплагіат	Нічепорук А.О., доцент кафедри КІС		

7. Дата видачі завдання « 10 » 01 2024 р.

КАЛЕНДАРНИЙ ПЛАН

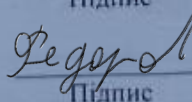
№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2024	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2024	виконано
4	Робота над розділом 2 – вибір компонентів для дослідження методів та аналіз автоенкодера	01.04.2024	виконано
5	Робота над розділом 3 – використання автоенкодера для розпізнавання об'єктів у фото	29.04.2024	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2024	виконано
7	Попередній захист ВКР	26.05.2024	виконано
8	Захист ВКР на засіданні ЕК	Червень 2024 року	

Студент


Підпис





С. В. Семенюк
Ініціали, прізвище

Керівник роботи


Підпис

Є. С. Федоров
Ініціали, прізвище

№ р я д к а	Ф о р м а т	Позначення	Найменування	К і л л и с т і в	№ ек з	П р и м і т к а
			<u>Текстові документи</u>			
1		КвРКІ 200244.20.02.22 ПЗ	Пояснювальна записка	70		
			<u>Графічні матеріали</u>			
2		КвРКІ 200244.20.02.22 Е8	UML Схема програмного забезпечення	1		
3		КвРКІ 200244.20.02.22 Е8	Зображення архітектур нейромерж автоенкодерів	1		
4		КвРКІ 200244.20.02.22 Е8	Двійкове арифметичне кодування карт ознак	1		

					КвРКІ 200244.20.02.22 ПЗ			
Зм	Арк	№ докум	Підпис	Дата	Система розпізнавання зображень на базі методів, заснованих на автоенкодері. Відомість проекту	Літера	Аркуш	Аркушів
Розробив		Семенов				У	1	1
Перевір.		Федоров				ХНУ, КІ2-20-2		
Н. контр.		Лисенко						
Зав.		Говорушенко		21.06				

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система розпізнавання зображень на базі методів, заснованих на автоенкодері».

Автор роботи: Семенюк Євгеній Валерійович.

Керівник роботи: Федоров Євген Євгенович.

Пояснювальна записка: 70 с., 50 рис., 8 табл., 3 дод., 52 джерела.

Графічна частина: 3 креслення.

ШНМ, АВТОЕНКОДЕР, РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, СЕГМЕНТАЦІЯ ЗОБРАЖЕНЬ, UNet, CNN, SVM, ReLU, ASU.

Метою дипломної роботи є дослідження та аналіз системи розпізнавання зображень на базі методів, заснованих на автоенкодері.

Об'єктом дослідження є процес розпізнавання зображень заснованого на автоенкодері.

Предметом дослідження є методи розпізнавання зображень, які засновані на базі автоенкодеру.

Завданням дослідження було:

- аналіз систем розпізнавання зображень;
- аналіз нейромережових методів розпізнавання зображень;
- аналіз розпізнавання об'єктів у фото.



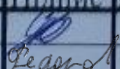
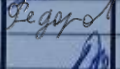
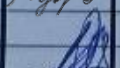

Підпис студента

30.05.2024

Дата

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ТА ОГЛЯД ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ	5
1.1 Архітектура штучних нейронних мереж	5
1.2 Аналіз нейромережі неглибокого автоенкодера	10
1.3 Використання автоенкодерів для класифікації	12
1.4 Аналіз систем розпізнавання зображень на прикладі обличчя	13
1.5 Постановка завдань	13
1.6 Висновки	19
2 ДОСЛІДЖЕННЯ МЕТОДІВ ТА АНАЛІЗ АВТОЕНКОДЕРУ	20
2.1 Реалізація UNet для сегментації зображень	20
2.2 Усунення шумів зображення за допомогою згорткового автокодувальника	27
2.3 ASU-CNN: ефективна глибока архітектура для класифікації зображень і візуалізації функцій	36
2.4 Архітектура, що поєднує згорткову нейронну мережу (CNN) і машина опорних вікторів (SVM) для класифікації зображень	45
2.5 Висновки	52
3 ВИКОРИСТАННЯ АВТОЕНКОДЕРУ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ У ФОТО	53
3.1 Навчений компресор зображень на основі автокодувальника	53
3.2 Swapping Autoencoder для глибокої обробки зображень	60
3.3. Автокодер із збереженням конфіденційності для спільного виявлення об'єктів	66
3.4. Висновки та перспективи	72
ВИСНОВОК	73
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	74

КвРКІ 200244.20.02.22 ПЗ					
Зм.	Арк.	№докум.	Підпис	Дата	
Виконав		Семенов С.В.			Система розпізнавання зображень на базі методів, заснованих на автоенкодері. Пояснювальна записка
Перевір.		Федоров С.Є.			
Н.контр.		Лисенко С.М.			Літера Аркуш Аркушів y 2 70 ХНУ КІ2-20-2
Затвер.		Говорушченко Т.О.		21.06	

ДОДАТОК А КОПІЯ КРЕСЛЕННЯ «UML СХЕМА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ» 79

ДОДАТОК Б КОПІЯ КРЕСЛЕННЯ «ЗОБРАЖЕННЯ АРХІТЕКТУР НЕЙРОМЕРЕЖ АВТОЕНКОДЕРІВ» 80

ДОДАТОК В КОПІЯ КРЕСЛЕННЯ «ДВІЙКОВЕ АРИФМЕТИЧНЕ КОДУВАННЯ КАРТ ОЗНАК» 81

КвРКІ 200244.20.02.22 ПЗ

Арк.
3

Зм.	Арк.	№ докум.	Підпис	Дата

ВСТУП

З розвитком технологій і збільшенням обсягу візуальних даних, системи розпізнавання зображень стали невід'ємною частиною сучасних інформаційних систем. Вони знаходять застосування у багатьох галузях, від медичної діагностики до автономного водіння та розпізнавання обличчя. Одним із перспективних підходів у цій сфері є використання автоенкодерів – спеціалізованих нейронних мереж, що вивчають компактні та значущі представлення зображень. Автоенкодери складаються з двох основних частин: енкодера, який зменшує розмірність вхідних даних та декодера, який відновлює дані з компактного представлення. Завдяки здатності автоматично виявляти та зберігати важливі особливості зображень, автоенкодери можуть ефективно використовуватися для розпізнавання, класифікації та покращення якості зображень. У цій роботі будуть досліджуватися основні принципи побудови систем розпізнавання зображень на базі автоенкодерів, їх переваги та потенційні області застосування.

Метою дипломної роботи є дослідження та аналіз системи розпізнавання зображень на базі методів, заснованих на автоенкодері.

Об'єктом дослідження є процес розпізнавання зображень заснованого на автоенкодері.

Предметом дослідження є методи розпізнавання зображень, які засновані на базі автоенкодеру.

Завданням дослідження було:

- аналіз систем розпізнавання зображень;
- аналіз нейромережових методів розпізнавання зображень;
- аналіз розпізнавання об'єктів у фото.

					КВРКІ 200244.20.02.22 ПЗ	Арк. 4
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ТА ОГЛЯД ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

1.1 Архітектура штучних нейронних мереж

У загальному випадку поняття “штучна нейронна мережа” охоплює ансамблі нейронів будь-якої структури, однак практичне застосування знайшли тільки деякі з них. Це пояснюється тим, що архітектура ШНМ безпосередньо зв'язана з методом її навчання. Навіть різні етапи розвитку ШНМ визначалися появою нових архітектур мереж і спеціально розроблених для них методів навчання. Можна виділити чотири основні різновиди архітектури ШНМ.

1. Одношарові прямонаправлені мережі. Шаровою називається ШНМ, що складається з груп нейронів, розділених по шарах. ШНМ, що містить k шарів, називається k -шаровою. Якщо сигнали в мережі поширюються тільки по напрямку з початку в кінець, то така ШНМ називається прямонаправленою. На рисунку 1.1 представлена прямонаправлена мережа з одним шаром нейронів.

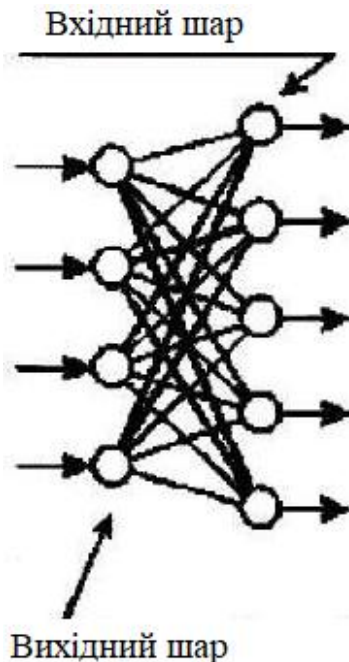


Рисунок 1.1 – Прямонаправлена мережа з одним шаром нейронів

Зм.	Арк.	№ докум.	Підпис	Дата

Вона містить у собі шар вхідних нейронів і шар вихідних. Нейрони вхідного шару просто ретранслюють сигнали на вихідний шар, не преутворюючи їх. У вихідному шарі відбувається перетворення сигналів і формування реакції мережі. Необхідно відзначити, що нейрони ШНМ також іноді називаються вузлами або обчислювальними модулями. Кількість нейронів в одному шарі визначає розмір шару. На відміну від прийнятої методики позначення, такі ШНМ називаються одношаровими, а не двошаровими. Цим підкреслюється, що обчислення виконуються лише одним шаром мережі.

2. Багатошарові прямонаправлені мережі. Вони характеризуються наявністю одного або декількох схованих шарів, що здійснюють перетворення інформації. Нейрони схованого шару називаються схованими нейронами або схованими вузлами. Використання схованих шарів дозволяє ШНМ здійснювати нелінійні перетворення вхід-вихід будь-якої складності або витягати з вхідних даних статистичні показники високих порядків. Ці унікальні властивості багатошарових мереж особливо виявляються при високій розмірності простору вхідних сигналів. На рисунку 1.2 представлена схема тришарової прямонаправленої ШНМ з одним схованим шаром. Для опису такої мережі використовується запис NN3-5-2. Тут 3 - розмір вхідного шару мережі, 5 - схованого і 2 - вихідного.

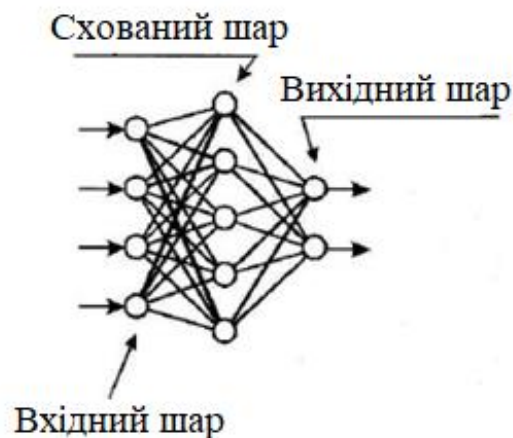


Рисунок 1.2 – Цілком зв'язана тришарова прямонаправлена мережа

Нейрони вхідного шару в таких мережах просто ретранслюють вхідні сигнали на перший схований шар, не преутворюючи їх. У схованих нейронах послідовно, шар за шаром, відбувається нелінійне перетворення сигналів. Сигнали з останнього схованого шару надходять на нейрони вихідного шару, що формують реакцію мережі.

Формально, не існує обмежень на типи активаційних функцій нейронів різних шарів ШНМ або навіть одного шару, однак зазвичай всі сховані нейрони вибираються одного типу. Вихідний шар ШНМ може складатися з нейронів з тим же типом активаційної функції, що й у нейронів схованого шару, але найбільш розповсюдженим є модель прямонаправленої мережі з лінійними вихідними нейронами. ШНМ цього типу з активаційними функціями нейронів схованого шару називаються багатошаровими перцептронами (БШП). БШП знайшли широке застосування при розв'язанні різних задач і є одним з головних об'єктів теоретичних досліджень. На рисунку 1.2 ШНМ називається цілком зв'язаною прямонаправленою ШНМ, тому що кожен нейрон з одного шару зв'язаний із усіма нейронами наступного шару. Широке застосування також знаходять частково зв'язані прямонаправлені ШНМ, у яких нейрони одного шару зв'язані тільки з визначеною частиною нейронів наступного шару. Така архітектура дозволяє закласти в ШНМ апріорні знання про бажаний закон обробки сигналів у мережі. Тришарові прямонаправлені ШНМ широко використовуються для розв'язання задач класифікації, розпізнавання образів, апроксимації і керування.

3. Рекурентні мережі. Цей тип ШНМ відрізняється існуванням зворотних зв'язків і елементів тимчасової затримки сигналу. Найбільш простим випадком рекурентної мережі є один шар нейронів, охоплений зворотними зв'язками. При цьому кожен нейрон одержує затримані вихідні сигнали всіх інших нейронів.

На рисунку 1.3 представлена рекурентна ШНМ, що містить схований шар нейронів. У цьому випадку кожен нейрон одержує, крім вхідних сигналів, ще й усі вихідні сигнали мережі. Частина ШНМ, яка охоплена зворотними зв'язками, може мати і більшу кількість схованих шарів.

					КВРКІ 200244.20.02.22 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

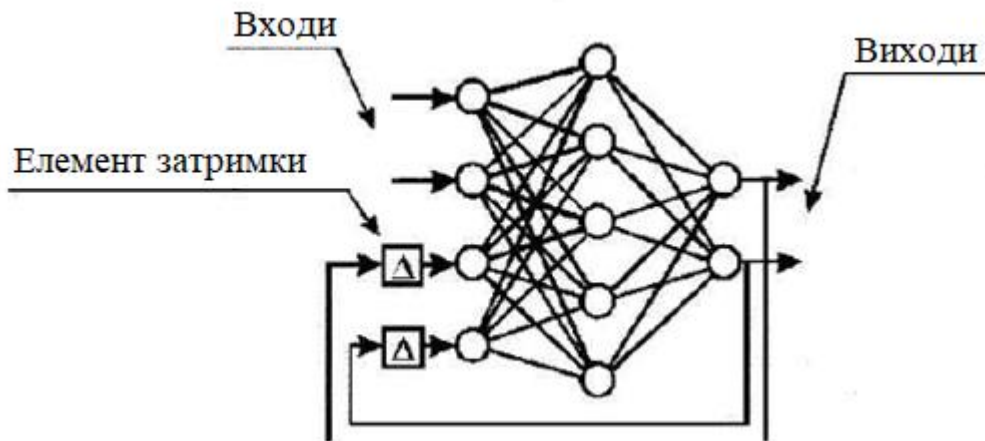


Рисунок 1.3 – Прямонаправлена мережа з одним шаром нейронів

Наявність зворотних зв'язків і елементів тимчасової затримки сигналів додає рекурентним мережам власні нелінійні динамічні властивості. Це також позначається на їхній здатності до навчання. Тренування рекурентних мереж вимагають урахування їхніх динамічних властивостей. Одним з головних застосувань рекурентних ШНМ є нейроемулятори динамічних об'єктів, тобто їхні нейромережні моделі. Такі мережі можуть також використовуватися для розв'язання задач апроксимації тимчасових послідовностей, класифікації, розпізнавання образів і керування.

4. Цілком зв'язані мережі. Характерною ознакою ШНМ цього типу є наявність зв'язків між усіма нейронами. Найбільш відомим різновидом цілком зв'язаних мереж є мережі Хопфілда. У них кожен нейрон має двосторонні зв'язки з всіма іншими нейронами мережі. У загальному випадку мережа Хопфілда має симетричну кільцеву структуру, у ній не можна виділити сховані нейрони і єдиний напрямок поширення сигналів. Робота цілком зв'язаної ШНМ і обмін даними контролюється одним головним нейроном. На рисунку 1.4 представлена цілком зв'язана мережа Хопфілда.

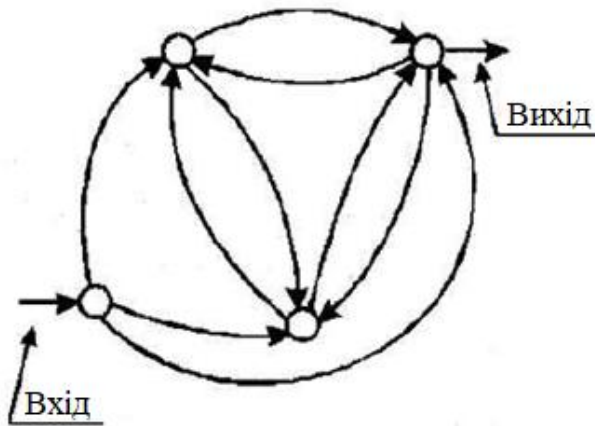


Рисунок 1.4 – Цілком зв'язана мережа Хопфілда

Мережа Хопфілда являє приклад цілком зв'язаної динамічної мережі, що ґрунтується на принципах самоорганізації, однак у ній не використовуються в явному виді елементи тимчасової затримки. Іншим прикладом служать решіткові мережі. Вони являють собою масив нейронів, кожний з яких зв'язаний із вхідними нейронами. Розмірність масиву нейронів визначає розмірність решіткової мережі. Така ШНМ є прямонаправленою, тому що в ній немає зворотних зв'язків, однак у ній не можна виділити скриті елементи або шари. На рисунку 1.5 представлена одномірна решіткова мережа з трьох нейронів.

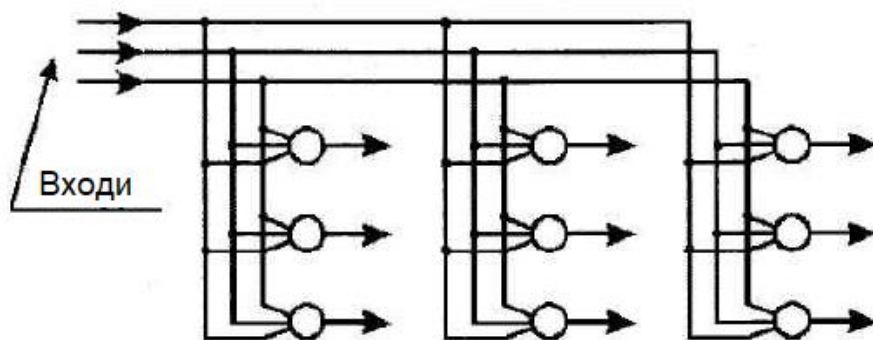


Рисунок 1.5 – Одномірна решіткова мережа з трьох нейронів

Цілком зв'язані мережі знаходять особливо широке застосування при розв'язанні задач класифікації і розпізнавання образів.

1.2 Аналіз нейромережі неглибокого автоенкодера

Автоенкодер, який є нерекурентною статичною багатошаровою мережею, що містить один або більше прихованих шарів та вихідний шар. На відміну від MLP вхідний і вихідний шар мають однакову кількість нейронів і на них подаються однакові дані, кількість прихованих шарів завжди непарна, кількість нейронів у прихованих шарах менше кількості нейронів у вхідному/вихідному шарі і зменшується в міру наближення до центрального (code) прихованого шару. Будь-який автоенкодер включає шифрувач (encoder) і дешифрувач (decoder). У випадку одного прихованого шару та лінійної функції активації автоенкодер стає аналогічним PCANN. Автоенкодер реалізує автоасоціативну пам'ять (її елемент (комірка) представлений парою зразків) і відновлює (витягує) запам'ятований зразок за ключовим зразком, що відповідає вхідному вектору.

Найважливішою властивістю автоенкодера є те, що та сама мережа із одними й тими самими вагами зв'язків може зберігати і відтворювати кілька запам'ятованих зразків. Для автоенкодера використовується навчання на основі корекції помилок, при цьому найчастіше застосовується метод градієнтного спуску, на якому заснований метод зворотного поширення (BP) для ШНМ прямого поширення. На рисунку 1.6 представлений не глибокий автоенкодер з одним прихованим шаром.

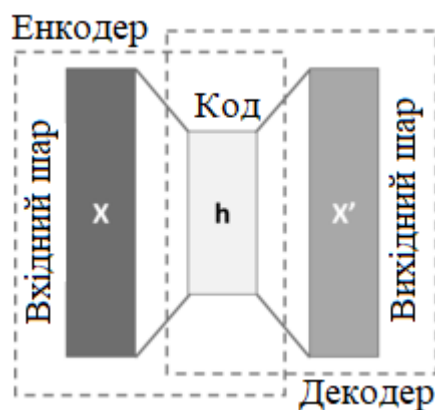


Рисунок 1.6 – Не глибокий автоенкодер (1 прихований шар)

Існують розріджені автоенкодері, в яких кількість нейронів прихованого шару більша, ніж кількість нейронів вхідного/вихідного шару. І тут під час навчання на прихований шар накладається штраф за розрідженість.

Переваги:

- використовується для відновлення зразків;
- забезпечує хорошу якість узагальнення;
- можливе навчання у послідовному та пакетному режимі.

Недоліки:

- навчання відбувається повільніше, ніж у разі MLP;
- відсутнє автоматичне визначення числа прихованих шарів та числа нейронів у цих шарах;
- на відміну від ART, не вирішує проблему пластичності-стабільності;
- подає зображення вектором.

Переваги використання автоенкодерів у системах розпізнавання:

- автоматичне витягування ознак: автоенкодері можуть самостійно вчитися витягувати важливі ознаки з даних без необхідності ручної інженерії ознак;
- навчання без учителя: вони можуть ефективно працювати в режимі без навчання на позначених даних, що робить їх особливо використовуваними в завданнях без навчальних етикеток;
- спрощення відображення даних: використання автоенкодерів для стискання та розкодування даних може значно зменшити їхню розмірність, полегшуючи подальшу обробку;
- генерація нових даних: можливість генерації нових, реалістичних даних для розширення обсягу тренувального набору та поліпшення роботи моделей;
- самонавчання та адаптація: автоенкодері можуть адаптуватися до нових умов та сценаріїв без повного перенавчання, що забезпечує більш гнучку роботу в реальних умовах.

Недоліки використання автоенкодерів у системах розпізнавання:

					КВРКІ 200244.20.02.22 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

- потреба у великому обсязі даних: для ефективного навчання автоенкодерів може бути потрібний значний обсяг даних, що може бути складно забезпечити в деяких застосуваннях;
- обчислювальна складність: використання глибоких автоенкодерів може вимагати значних обчислювальних ресурсів та тривалого часу для навчання;
- втрата інтерпретованості: глибокі автоенкодери можуть ускладнити інтерпретацію вивчених репрезентацій та зрозуміння того, як саме вони використовують ознаки;
- чутливість до шуму: автоенкодери можуть бути чутливими до введеного шуму, що може вплинути на якість вивчених репрезентацій;
- питання безпеки та конфіденційності: використання автоенкодерів у завданнях розпізнавання може викликати питання безпеки та конфіденційності.

1.3 Використання автоенкодерів для класифікації

Під час навчання автокодувальників у неконтрольований спосіб (тобто за відсутності міток), їх можна використовувати також у напівконтрольованому режимі (де частина дані мають мітки) для покращення результатів класифікації. В даному випадку кодер використовується як екстрактор ознак і «підключений» до мережі класифікації. Це в основному виконується в напівкерованому навчанні, де надається великий набір даних контрольоване навчальне завдання, але лише невелика його частина позначена.

Основне припущення полягає в тому, що зразки з однаковою міткою повинні відповідати деяким латентне представлення, яке може бути апроксимовано латентним шаром автокодерів. По-перше, автокодери навчаються без нагляду. Потім (або паралельно) декодер відкладається, а кодер використовується як перша частина моделі класифікації. Його ваги можуть бути точно налаштовані або залишатися фіксованими під час навчання машина (SVM) навчається на вихідних функціях кодера. У випадках, коли домен має велику розмірність, а пошарове

навчання неможливо, рішення полягає в тому, щоб навчити кожен шар як лінійний перед додаванням нелінійності. У цьому випадку, навіть із зменшенням шуму на вхідних даних, для кожного існує рішення закритої форми і не потрібен ітераційний процес.

Інший підхід використовує автокодері як метод регуляризації для класифікації мережі. Наприклад, дві мережі підключені до кодера, мережа класифікації (навчена з позначеними даними) і мережа декодера (навчені реконструювати дані, позначені чи не позначені). Проведення реконструкції головка крім класифікаційної головки служить регуляризатором для останньої. На рисунку 1.7 представлено використання автокодерів, як регуляризації для керованих моделей.

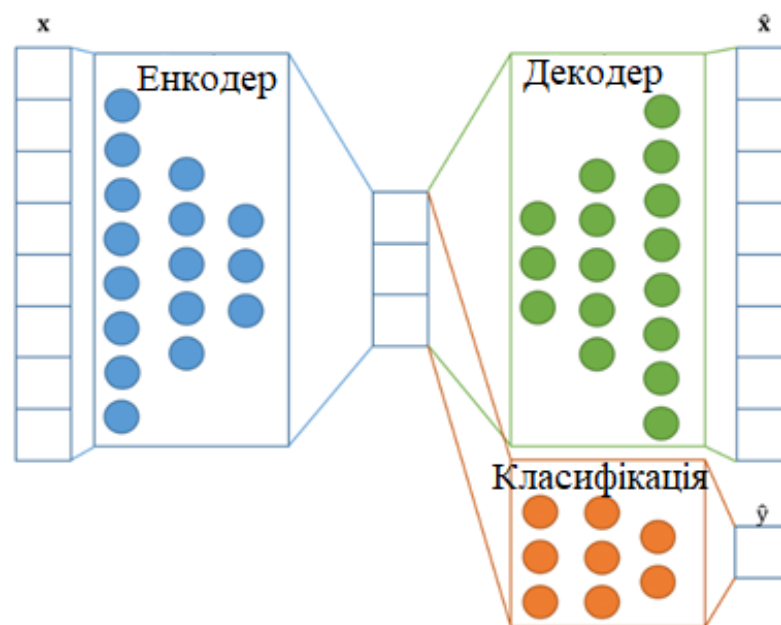


Рисунок 1.7 - Використання автокодерів, як регуляризації для керованих моделей

1.4 Аналіз систем розпізнавання зображень на прикладі обличчя

Перевірка обличчя слугує засобом доступу до програми, системи чи служби. Його основна мета — оцінити надане зображення обличчя порівняно з іншим обличчям і підтвердити чи вони відповідають. По суті, коли представлено два

зображення обличчя, алгоритм перевірки обличчя визначає чи вони зображують ту саму особу. Цей метод, відмінний від традиційних методів перевірки, таких як паролі або відбитки пальців, покладається на динамічні шаблони в рамках біометричної перевірки обличчя, що робить його надзвичайно безпечним і ефективним. Крім контролю доступу, розпізнавання облич знаходить застосування в криміналістиці та автентифікації транзакцій. Використання глибоких нейронних мереж у верифікації обличчя, продемонструвало значний прогрес у системах верифікації обличчя. Ці глибокі нейронні мережі також виявилися успішними в різних мовних програмах і стикаються зі сценаріями захисту від спуфінгу. Одним із яскравих прикладів є система Facenet, розроблена Google, яка використовувала сіамську мережу, навчену на міченому наборі даних, що включає 200 мільйонів облич. Вона досягла чудової точності понад 98% на наборі даних LFW і понад 95% на YTF, обидва набори даних для перевірки обличчя. DeepFace, розроблений Meta, використовував 3D-моделювання обличчя та дев'ятишарову мережу з приблизно 120 мільйонами параметрів. На основі 4,4 мільйона позначених зображень облич DeepFace досяг точності понад 97% на наборі даних LFW. Подальші вдосконалення були зроблені шляхом включення понад 500 мільйонів зображень обличчя для навчання, що призвело до підвищення продуктивності понад 98%. В іншому підході система перевірки обличчя використовувала граничні втрати та була навчена на 4 мільйонах позначених набір даних, що забезпечує точність понад 99% на LFW і понад 95% на YTF. ArcFace запровадив адитивну кутову втрату запасу, яка перевищила 99% точності на LFW та понад 98% на YTF. GroupFace, використовуючи численні представлення з урахуванням груп, досяг понад 99% і 97% на наборах даних LFW і YTF відповідно, хоча і ArcFace, і GroupFace вимагали позначеного навчального набору даних із 5,8 мільйонів зразків. CurricularFace реалізував адаптивну втрату навчання за навчальною програмою, перевищивши 99% точності на наборі даних LFW. MDCNN складається з двох розширених моделей нейронних мереж глибокого навчання, які досягли понад 99% і 94% на наборах даних LFW і YTF відповідно, використовуючи 1 мільйон позначених

					КВРКІ 200244.20.02.22 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

наборів навчальних даних. Методи напівконтрольованого та неконтрольованого навчання в контексті глибоких нейронних мереж зазвичай використовують один із двох основних підходів. Перший — це метод, заснований на регуляризації узгодженості, який включає термін регуляризації в цільову функцію, спрямований на забезпечення узгодженості під час навчання, особливо при роботі зі значною кількістю немаркованих даних. Ця регуляризація обмежує прогнози моделі залишатися стабільними або незмінними, коли вони стикаються з вхідним шумом або варіаціями. Вони представили підхід неконтрольованої адаптації домену, який використовує вдосконалені методи розширення даних, такі як ранд-аугмент і зворотний переклад для підвищення надійності моделі. Другий — це методи на основі міток проксі. У другому підході псевдомітки, які часто називають проксі-мітками, спочатку призначаються даним без міток. Згодом навчання відбувається на основі цих проксі-міток у поєднанні зі справжніми мітками реальної інформації як для мічених, так і для немаркованих даних. Цей метод ефективно використовує ці проксі-мітки для керування процесом навчання моделі. Ці дві стратегії представляють ключові шляхи для напівконтрольованого навчання з глибокими нейронними мережами, кожна з яких пропонує певні переваги та компроміси в різних програмах. Дослідження представило алгоритм кластеризації контрасту на основі коефіцієнта силуету. Цей інноваційний підхід оцінює ступінь поділу між кластерами шляхом ретельного вивчення як внутрішньокластерних, так і міжкластерних відстаней, що дозволяє точно виявляти дані, що не розподіляються. Робота пропонує дискримінаційну розріджену регресію найменших квадратів для напівконтрольованого навчання. Дослідження синтез обличчя та позово-інваріантне розпізнавання обличчя відбувається за допомогою генеративних змагальних мереж (GAN).

Покладаючись на неконтрольовані мережі, дослідники можуть використовувати внутрішню структуру та шаблони даних, щоб створити індивідуальні представлення для кожної людини. Цей підхід може підвищити

					КВРКІ 200244.20.02.22 ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

точність розпізнавання, особливо в сценаріях, коли позначені дані для кожної особи можуть бути дефіцитними або непрактичними для отримання.

Таким чином, неконтрольовані методи глибокого навчання, такі як RBM, DBN і автокодер, пропонують багатообіцяючий шлях для розпізнавання зображень обличчя, який не потребує мічених даних для всіх осіб. Натомість вони дозволяють тренувати окремі моделі для кожної цільової особи, таким чином використовуючи потужність неконтрольованого навчання для підвищення ефективності розпізнавання. На рисунку 1.8 представлено пропонуване вилучення вбудовування обличчя з попереднього навчання автокодувальника.

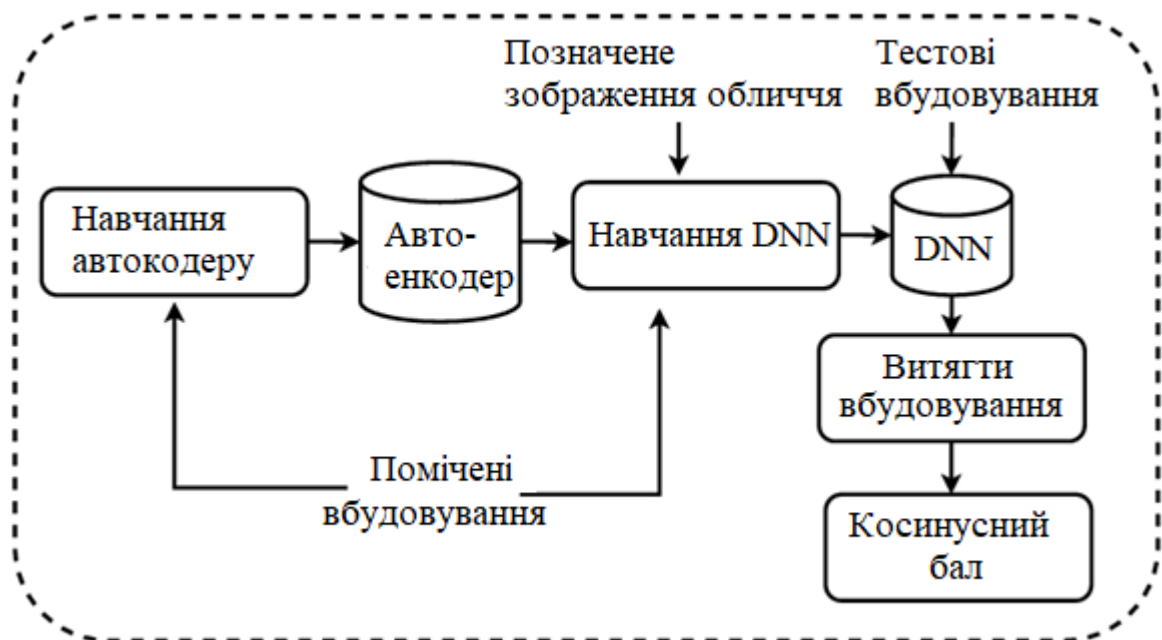


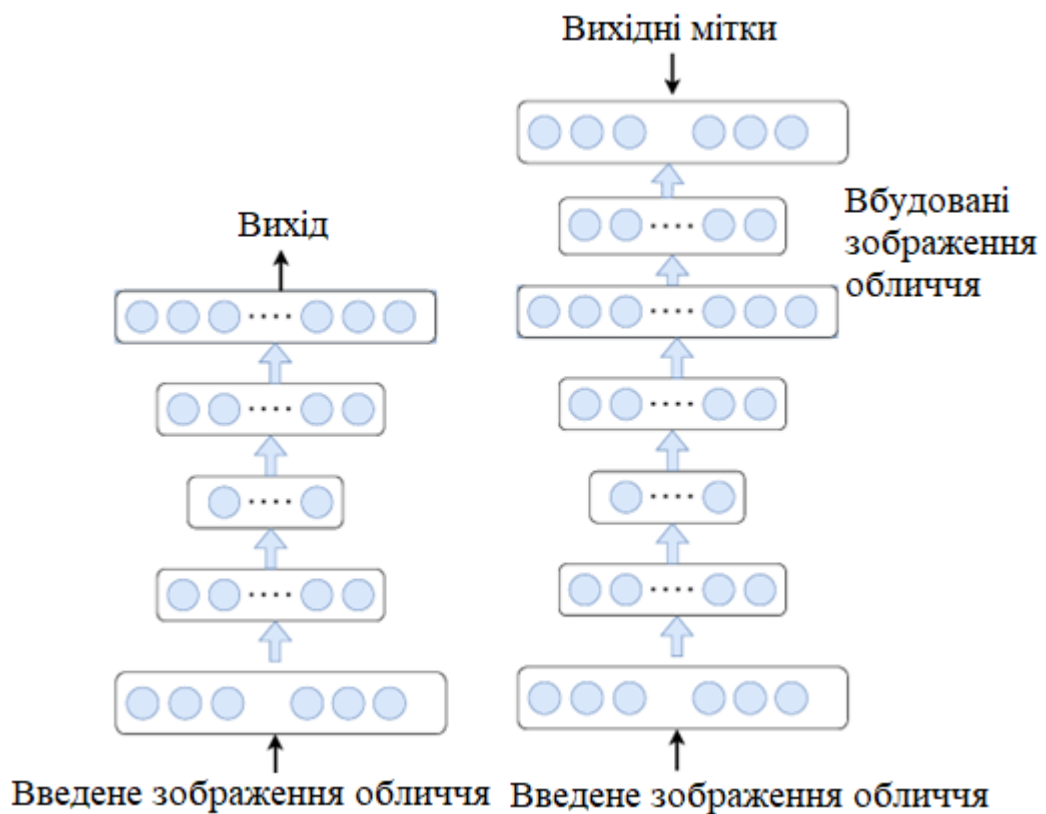
Рисунок 1.8 - Пропонуване вилучення вбудовування обличчя з попереднього навчання автокодувальника

Багато контрольованих методів, згаданих вище, вимагають обширних позначених навчальних даних, ресурсу, якого часто бракує в багатьох областях. Крім того, проблема зі збором достатньої кількості міток для програм розпізнавання в реальному світі ще більше ускладнює справу. Отже, у таких сценаріях продуктивність цих методів значно знижується.

У цій роботі представлено нову структуру для перевірки зображення обличчя, використовуючи автокодер для створення вектора вбудовування зображень обличчя. Спочатку автокодер навчається, використовуючи значний корпус зображень обличчя без міток. На цьому етапі навчання було дотримано традиційного підходу, спрямованого на мінімізацію втрати середньої квадратичної помилки (MSE) між вхідними даними та реконструйованим вектором зображення обличчя. Був використаний оптимізатор стохастичного градієнтного спуску (SGD), щоб полегшити цей процес оптимізації. Після навчання автокодувальника класифікатор глибокої нейронної мережі навчається під наглядом із маленьким позначеним набором навчальних даних, щоб отримати знання про обличчя окремих людей. Цей процес передбачає розширення архітектури автокодувальника шляхом додавання повністю підключеного рівня та рівня класифікації після останнього рівня. Запропонована глибока нейронна мережа ініціалізується параметрами з попередніх кроків попередньо навченого автокодувальника. Попереднє навчання автокодувальника допомагає в керованому навчанні, забезпечуючи початкове представлення даних, а також дозволяє мережі зближуватися відносно швидше за допомогою попереднього навчання порівняно з навчанням з нуля. Існує два помітні сценарії використання попередньо підготовленого автокодувальника для глибокої ініціалізації нейронної мережі. Один із підходів передбачає інтеграцію повністю підключених і класифікаційних рівнів безпосередньо після розділу кодера. Частина кодера стискає дані в більш низький розмір, зберігаючи достатньо інформації для реконструкції вихідних даних. Однак експериментальні висновки застерігають від цього методу. У запропонованому методі глибокого навчання нейронної мережі мережа отримує переваги від додаткової інформації, отриманої з міток зображення обличчя на виході. Обмежений вимірний простір кодера виявляється недостатнім для ефективного засвоєння знань з класифікаційного рівня вищого виміру. Інший підхід передбачає використання всього автокодувальника шляхом інтеграції повністю підключених і класифікаційних рівнів наприкінці архітектури автокодувальника. Було надано перевагу відновленню вхідних даних у вихідний

					КВРКІ 200244.20.02.22 ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підпис	Дата		

розмірний простір перед навчанням класифікатора глибокої нейронної мережі гібридного автокодувальника. Цей двоетапний процес дає кілька переваг. По-перше - це забезпечує нам зменшення розмірності. Спочатку було використано кодер для стиснення вхідних даних у нижчий вимір, ефективно видаляючи зайву інформацію. По-друге - це дозволяє розширювати інформацію. Згодом були розгортані дані назад у вихідний розмірний простір. Це розширення полегшує отримання додаткової інформації з міток зображення обличчя, підвищуючи здатність мережі вивчати розрізнявальні ознаки. На рисуюнок 1.9 представлено навчання автокодувальника та навчання DNN.



Рисуюнок 1.9 - Ліва сторона – це попереднє навчання автокодувальника, а права сторона – навчання DNN

Цей підхід дозволяє знайти баланс між зменшенням розмірності та збереженням інформації, що в кінцевому підсумку призводить до більш ефективного навчання під час процесу навчання глибокої нейронної мережі

гібридного автокодувальника. Автокодер має симетричну структуру з частиною кодера та частиною декодера, що нагадує звичайну конфігурацію автокодера. Тим часом глибока нейронна мережа підтримує подібну структурну симетрію. Нарешті, результати, отримані з останнього шару, служать затребуваними вбудованими зображеннями обличчя, які продемонстрували свою здатність ефективно зберігати індивідуальну інформацію про обличчя. Під час фази оцінювання витягуються вектори для вбудовування тестових зображень обличчя. Згодом ці вкладення використовуються в експериментальних випробуваннях, використовуючи техніку косинусної оцінки для цілей оцінки.

1.5 Постановка завдань

Об'єктом дослідження є процес розпізнавання зображень заснованого на автоенкодері.

Предметом дослідження є методи розпізнавання зображень, які засновані на базі автоенкодеру.

Метою дипломної роботи є дослідження та аналіз системи розпізнавання зображень на базі методів, заснованих на автоенкодері.

Завданням дослідження було:

- аналіз систем розпізнавання зображень;
- аналіз нейромережових методів розпізнавання зображень;
- аналіз розпізнавання об'єктів у фото.

1.6 Висновки

У першому розділі розглянуто поняття ШНМ та її архітектуру. Розглянуто переваги та недоліки використання автоенкодеру у системах розпізнавання зображень та використання автоенкодерів для класифікації. Зроблено аналіз систем розпізнавання зображень на прикладі облич та методів напівконтрольованого та неконтрольованого навчання в контексті глибоких нейронних мереж.

					КВРКІ 200244.20.02.22 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ДОСЛІДЖЕННЯ МЕТОДІВ ТА АНАЛІЗ НЕЙРОМЕРЕЖІ АВТОЕНКОДЕР

2.1 Реалізація UNet для сегментації зображень

Семантична сегментація зображення – це класифікація зображення на рівні пікселів, яка передбачає кластеризацію частин зображення, які належать до одного класу. Це завдання глибокого навчання є невід’ємною частиною комп’ютерного зору та розпізнавання образів і має значне застосування в таких сферах, як медична візуалізація, безпілотні автомобілі та аналіз супутникових зображень. У той час як згорткові нейронні мережі (CNN), такі як LeNet, VGGNet і GoogleNet, зазвичай використовуються для завдань класифікації, де результатом є єдина мітка класу, семантична сегментація вимагає інформації про локалізацію, тобто мітки класу для кожного пікселя. Отже, сегментація зображення з його попіксельною класифікацією вимагає більше обчислень, ніж класифікація об’єктів. Архітектури для сегментації зображень, такі як UNet, відіграють вирішальну роль у моделях дифузії, які використовуються для генерації зображень, таких як DALL-E OpenAI. Це підкреслює необхідність розробки апаратних реалізацій для цих мереж.

Реалізація цих складних алгоритмів глибокої нейронної мережі на традиційному апаратному забезпеченні, заснованому на архітектурі фон Неймана, потребує ресурсів з точки зору споживання енергії, площі та часу. В першу чергу це пов'язано з розділенням блоків пам'яті та обробки. Отже, існує потреба в спеціалізованих апаратних конструкціях, які використовують парадигму обчислень в пам'яті, пропонуючи оптимізацію, адаптовану для ефективної реалізації глибоких нейронних мереж.

Кілька досліджень були направлені на спеціалізовану апаратну реалізацію завдань сегментації. Ці роботи базуються на оптимізації сегментації для реалізації FPGA та розгортанні конвеєрної архітектури НВІС та засновані на пристроях CMOS і споживають велику потужність і площу. З іншого боку, пристрої Spintronic споживають меншу енергію, площу та сумісні з технологією CMOS. Пристрої

					КВРКІ 200244.20.02.22 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

Spintronic також мають перевагу в тому, що вони мають різноманітний діапазон властивостей, таких як енергонезалежність, коливання, пластичність, висока витривалість, лінійний відгук і стохастична поведінка. Ці властивості дають широкий спектр інструментів для розробки спеціалізованого обладнання для впровадження глибокої нейронної мережі. Однак їхня увага в першу чергу зосереджена навколо оптимізації потоку даних і доступу до пам'яті для множення матриць і не заглиблюється в розробку спеціалізованого апаратного забезпечення для сегментації зображень. Також необхідні реалізація та оптимізація інших важливих компонентів, таких як рівні активації та об'єднання. Було запропоновано спінтронну реалізацію шарів згортки, деконволюції, ReLU та максимального об'єднання, які є важливими для завдань сегментації. Для реалізації цих рівнів використовуються MTJ доменної стінки та SHE-MTJ. Було використано гібридний метод моделювання, який поєднує мікромагнітне моделювання, нерівноважну функцію Гріна Келдиша, рівняння Ландау-Ліфшица-Гільберта-Слончевського і моделювання схеми HSPICE із програмуванням на Python для охоплення різноманітної фізики пристроїв spintronic і CMOS. На рисунку 2.1 представлено структуру UNet, приклад вхідного зображення та його відповідний вихід.

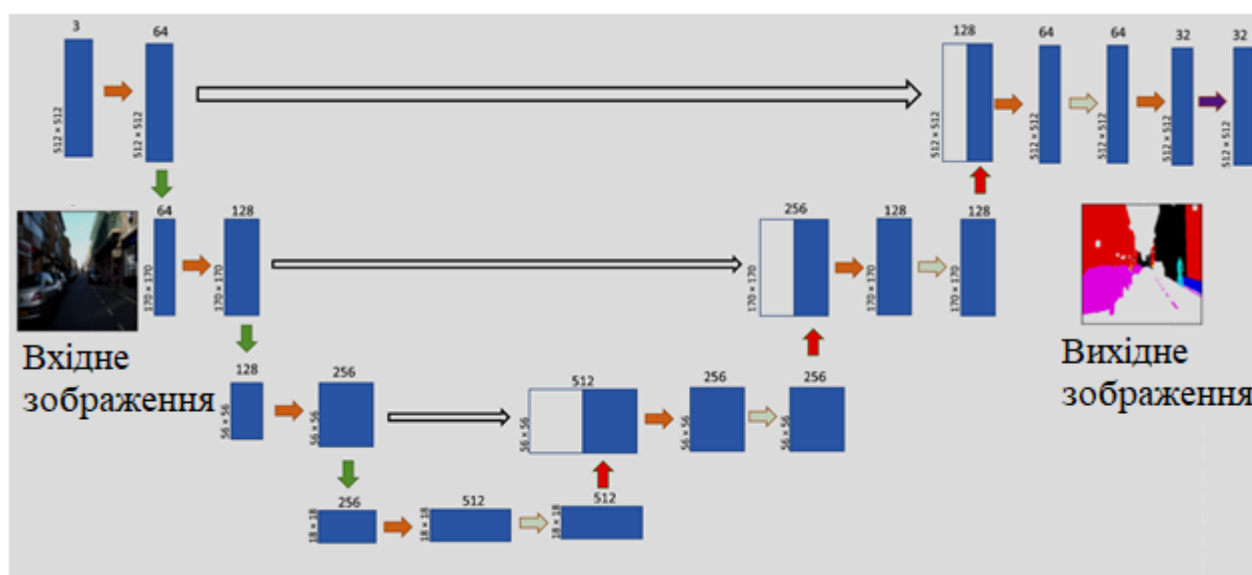


Рисунок 2.1 - Структура UNet, приклад вхідного зображення та його відповідний вихід

Існує кілька архітектур, розроблених для сегментації зображень, таких як UNet, SegNet тощо. Серед них UNet широко використовується в задачах сегментації. Архітектура UNet спочатку була запропонована Олафом Роннебергером та іншими для сегментації медичних зображень. Ця архітектура складається з двох основних компонентів: звужувального шляху (також відомого як кодер) і розширювального шляху (також відомого як декодер), з'єднаних шляхом копіювання (також відомий як пропуск з'єднання). Шлях, що скорочується, зменшує карту функцій під час вилучення функцій зображення, а шлях розширення використовує ці функції для локалізації об'єктів і реконструкції маски сегментації. Коли карта функцій зазнає скорочення на шляху скорочення, деяка інформація втрачається, щоб вирішити цю проблему, використовується з'єднання копіювання (з'єднання пропуску), щоб повторно ввести втрачену інформацію в шлях розширення. На рисунку 2.1 показано схему структури UNet, тут шлях скорочення містить шари згортки, ReLU та Max-pooling, тоді як шлях розширення містить шари деконволюції, згортки та ReLU, що завершуються функцією softmax. Реалізація сегментації зображення через UNet на апаратному забезпеченні вимагає розробки схем, призначених для згортки, деконволюції, функцій активації ReLU та рівнів максимального об'єднання.

Операція згортки передбачає множення матриці-вектора, де вхідні дані множаться на ядро. Ця операція множення матриці-вектора є фундаментальною для штучних нейронних мереж, де вхідна карта/карта ознак множиться на вагову матрицю. У згортці ядро можна розглядати як вагову матрицю. Виконання цього векторного множення вимагає великої кількості вибірок пам'яті при використанні традиційного апаратного забезпечення на основі архітектури фон Неймана. Таким чином, поперечні масиви стали дуже популярними для матриці-векторного множення. У поперечних масивах вагова матриця/ядро зберігаються в енергонезалежній пам'яті елементи (синапси), де аналогова пам'ять і обчислювальні блоки складно переплетені, що призводить до швидшого та енергоефективнішого множення матриць. На рисунку 2.2 вхідні дані

застосовуються до горизонтальних ліній, тоді як ваги ядра зберігаються як провідності синапсів у вертикальних лініях, а вихід векторного множення (зважена сума вхідних даних) визначається поточним значенням у вертикальних лініях. лінії.

На рисунку 2.2 представлено приклад поперечного масиву поперечних смуг на основі DW.

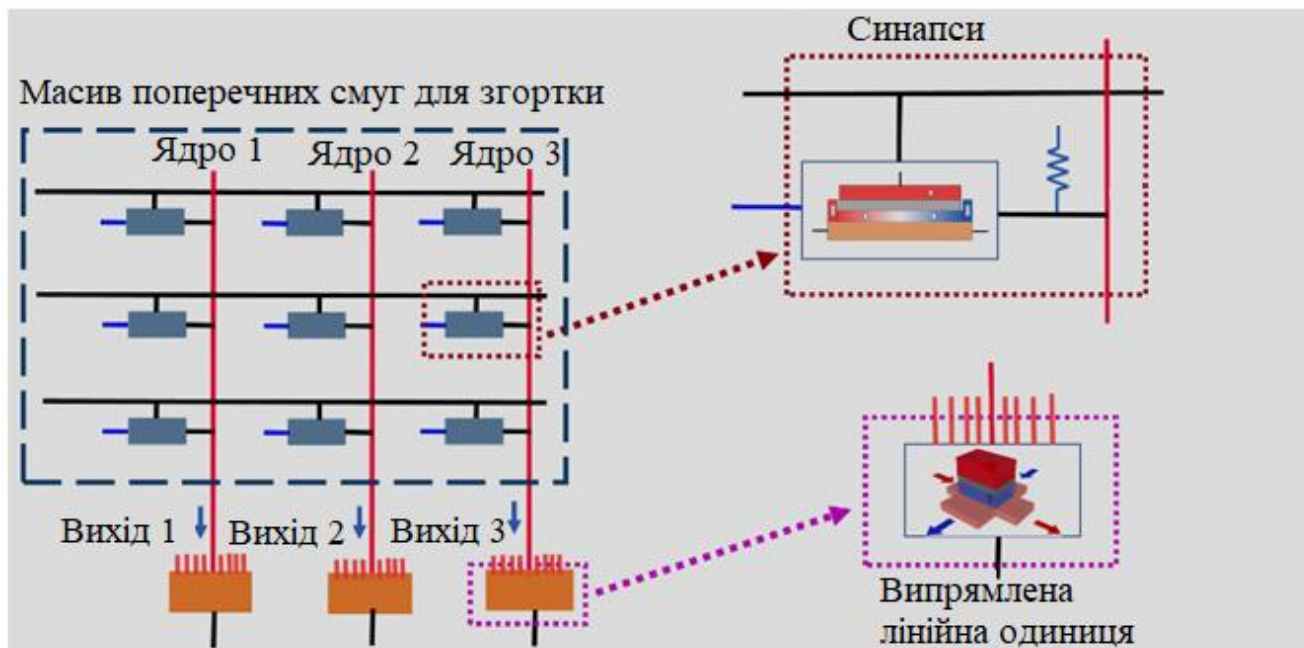


Рисунок 2.2 - Операція згортки з використанням масиву поперечних смуг на основі DW

Для реалізації такого поперечного масиву необхідний енергонезалежний синаптичний пристрій. Тому був використаний пристрій магнітного тунельного переходу на основі доменної стінки (DW-MTJ) для зберігання ваги ядра. Нейронна мережа може мати як позитивні, так і негативні ваги, але значення провідності DW-MTJ є лише позитивними. Щоб вирішити цю проблему, було додано провідність паралельно до DW-MTJ, як показано на рисунку 2.2.

Деконволюція, яку також називають транспонованою згорткою або дробово-поступовою згорткою, працює у зворотному напрямку згортки. Вона екстраполює нову інформацію з карти функцій і може розглядатися як з'єднання «один до багатьох». Деконволюція слугує технікою підвищення дискретизації зображень, у

результаті чого вихідний розмір перевищує розмір вхідного. Ця операція має значне застосування в генеративних змагальних мережах і повністю згорткових мережах. На рисунку 2.3 представлено операцію деконволюції, як комбінація операції введення нуля та операції згортання.



Рисунок 2.3 - Операція деконволюції, як комбінація операції введення нуля та операції згортання

Операція деконволюції може бути досягнута шляхом введення нулів у вхідну матрицю та виконання операції згортки. Рисунок 2.3 ілюструє операцію деконволюції як комбінацію вставки нуля та згортки. Нулі вставляються вздовж кожного рядка та стовпця, у тому числі на краях вхідної матриці, таким чином збільшуючи розмір вхідних даних. Потім ця матриця з підвищеною дискретизацією використовується як вхідні дані для згортки. Ця комбінація вставки нуля та згортки дає той самий ефект, що й деконволюція. Хоча цей метод передбачає надлишкові операції множення з нулями, він дозволяє використовувати операцію згортки, для якої було розроблено апаратну реалізацію з використанням поперечних. Такий підхід зменшує складність конструкції апаратного забезпечення для завдань сегментації. Отже, у проекті мережі представлено деконволюцію через операцію згортки з додатковим кроком вставки нуля. На рисунку 2.4 представлено огляд налаштування моделювання.

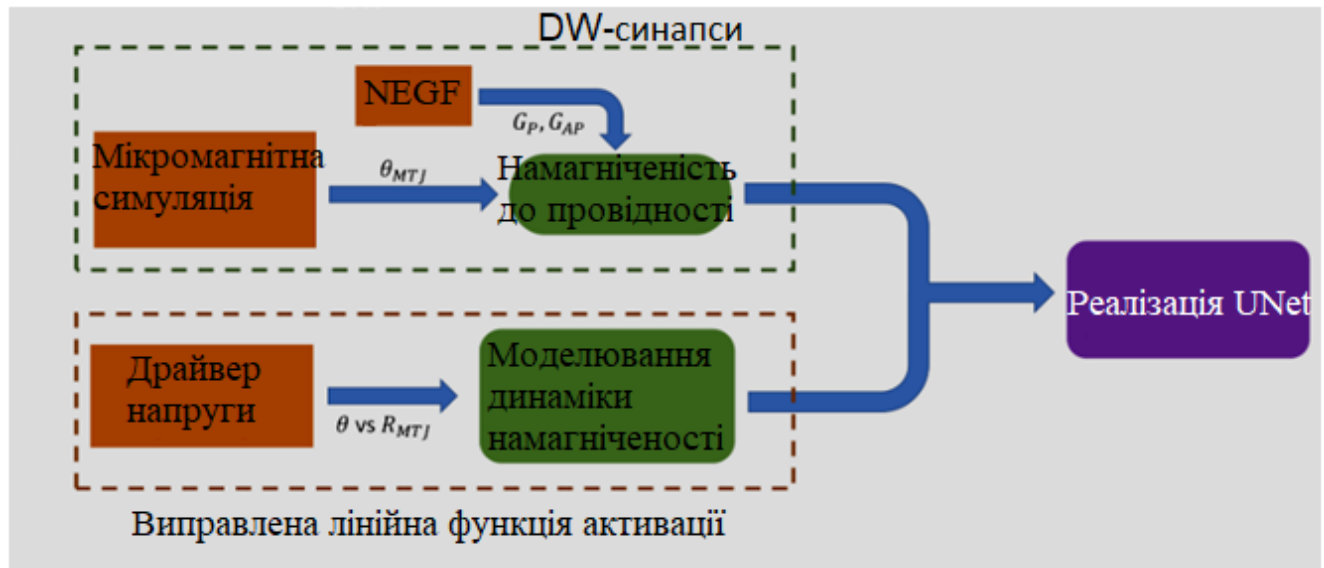


Рисунок 2.4 - Огляд налаштування моделювання

Функції активації відіграють вирішальну роль у нейронних мережах, вводячи нелінійність, що дозволяє мережі вивчати складні структури та розрізняти виходи. Виправлена лінійна функція активації (ReLU) стала вибором за замовчуванням для різних мереж, оскільки було показано, що вона покращує навчання в нейронних мережах. У згорткових нейронних мережах (CNN), UNet і повністю зв'язаних згорткових мережах зазвичай включається рівень об'єднання, щоб зменшити розмір і параметри під час вилучення функцій. Серед різних методів об'єднання популярним є максимальне об'єднання, яке також має можливість пригнічувати шум шляхом відкидання шумових активацій.

Дизайн UNet було оцінено, використовуючи базу даних Cambridge-driving labeled video (CamVid). Ці дані були отримані з точки зору водіння автомобіля, сцена водіння збільшує кількість і різноманітність спостережуваних класів об'єктів. Набір даних містить 701 кольорове зображення з розмірами 512×512 пікселів, кожен піксель позначається в одному з 32 можливих класів. Ці класи включають такі об'єкти, як будівлі, автомобілі, дороги, діти, велосипедисти тощо. Щоб оцінити мережу, було розілено 701 зображення на набори з 369 для навчання, 100 для перевірки та 232 для тестування. На рисунку 2.5 представлено точність(%) і втрати UNet протягом 150 епох для наборів даних тестування та перевірки.

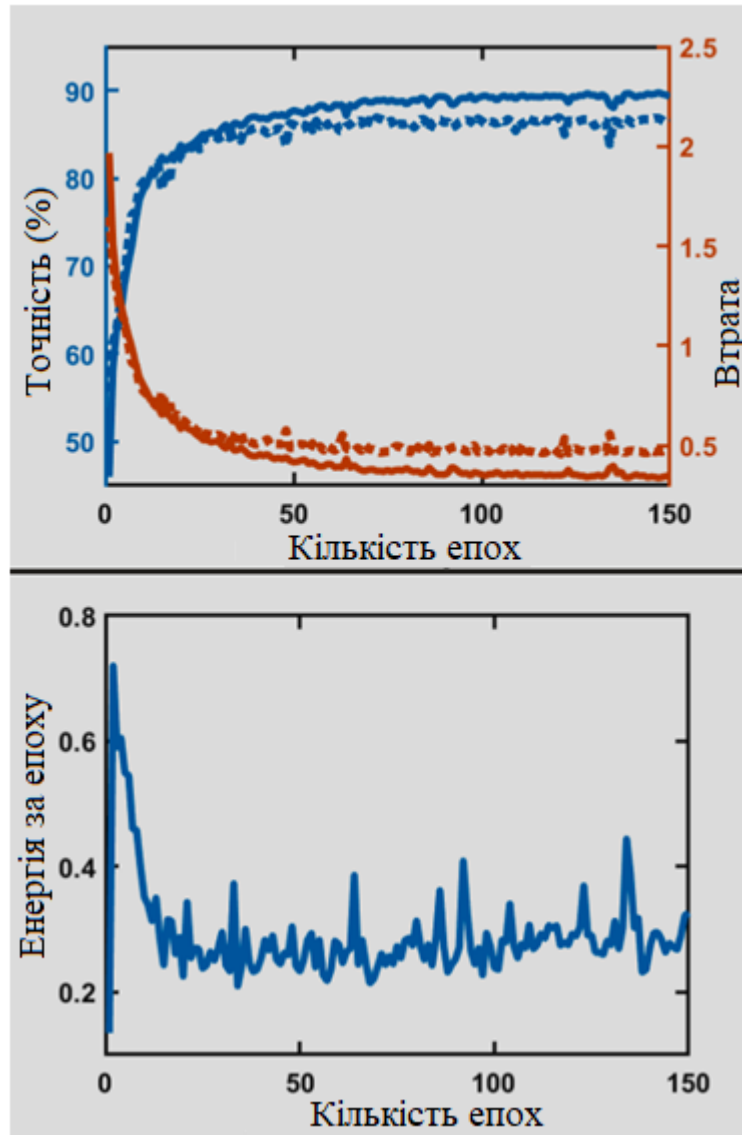


Рисунок 2.5 – Точність(%) і втрата UNet через 150 епох для набору даних CamVid.
Споживання енергії у всіх DW-синапсах під час навчання мережі як функція кількості епох

Було досягнуто точності перевірки 86,48 % і точності тестування 82,34 %, ці результати дуже схожі на результати повністю програмної реалізації, де точність перевірки становить 87,95 %, а точність тестування — 84,53 %. Було розраховано енергію, споживану синапсами під час тренування, як показано на рисунку 2.5. Розсіювання енергії за епоху зменшується, коли мережа проходить тренування та ваги зближуються. Загальна енергія, споживана всіма синапсами під час навчання за 150 епох, становить 43,59.

На рисунку 2.6 представлено результат UNet чотирьох тестових зображень разом із основними мітками істинності. Тут прогнозовані результати сегментації на основі спінтронної апаратної реалізації UNet дуже нагадують базові мітки істинності.

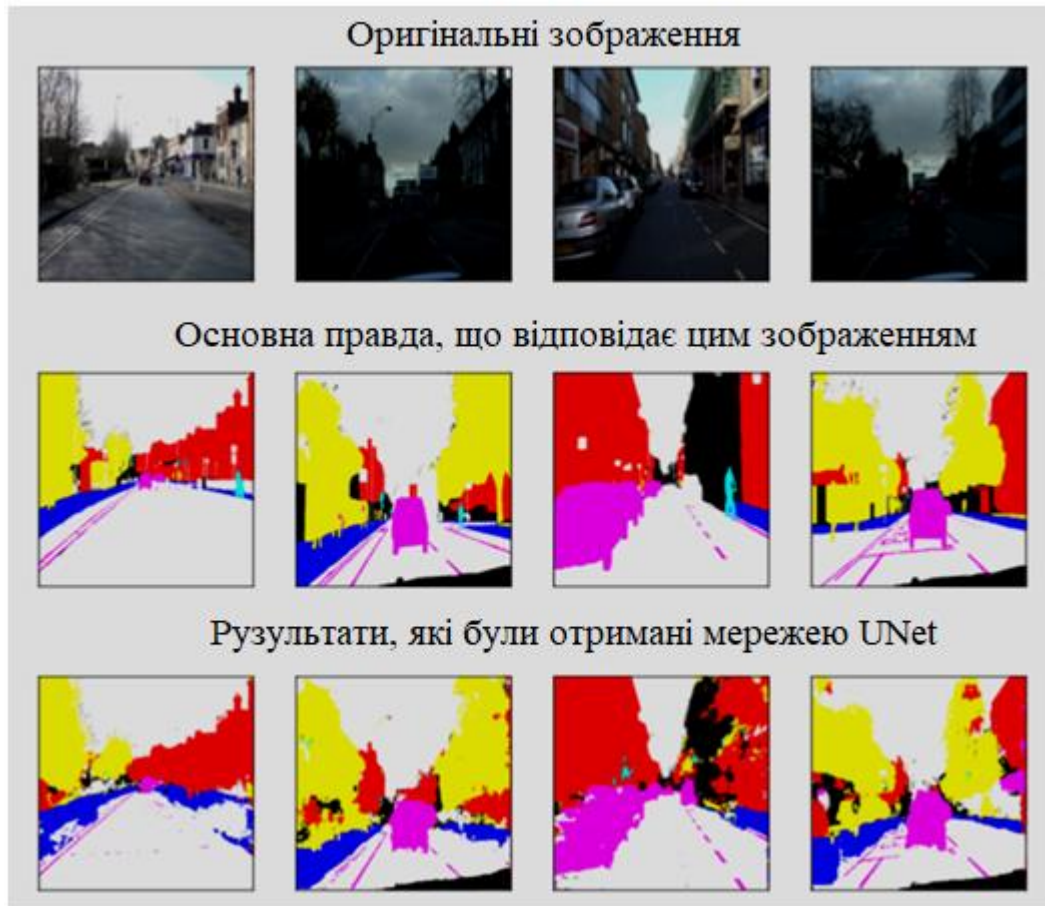


Рисунок 2.6 - Порівняння результатів UNet із фактичною правдою

Кожен колір у зображеннях міток відповідає окремому елементу зображення, наприклад, автомобілі зображені рожевим кольором, а будівлі – червоним.

2.2 Усунення шумів зображення за допомогою згорткового автокодувальника

Усунення шумів зображення стосується процесу усунення спотворень, інакше відомих як «шум» із даного вхідного зображення, щоб створити чіткіше зображення. Шум означає випадкові зміни яскравості/інтенсивності або інформації

про колір, присутній на цифровому зображенні. Шум має кілька способів виникнення, зокрема під час отримання даних, шум, спричинений помилками аналого-цифрового перетворювача, бітові помилки під час передачі, статистичні квантові флуктуації, несправності датчика зображення та багато іншого. Цей шум може зробити багато зображень марними, якщо не існує методів його усунення. Таким чином, усунення шумів на зображенні стає важливою проблемою, яка досі не доведена до досконалості. Щодня з'являється все більше нових методів, кожен з яких має певні випадки використання, але жоден із цих методів ніколи не міг повністю відновити зображення до його початкової форми. Згортковий автокодер — це модель глибокого навчання, створена за допомогою шарів нейронної мережі, зокрема згорткових шарів. Обидві моделі порівнюються між собою, щоб визначити точнішу.

Термін автокодер вперше з'явився у 1980-х роках для вирішення проблеми «зворотного поширення без вчителя», яка протистоїть використанню вхідних даних як очікуваного виходу. Разом із правилами геббійського навчання автокодери стали рудиментарними у сфері неконтрольованого навчання. Автокодер — це тип нейронної мережі, яка вчиться шифрувати/кодувати даний немаркований вхід у просторі розмірів, який може мати або не мати того самого порядку, що й вхідні дані. Він зазвичай відображає вхідні дані в просторі нижчих розмірів (латентний простір). Потім - це кодування використовується для реконструкції оригінального зображення. Упродовж цього процесу модель вивчає відображення різних вхідних зображень на певні точки в латентному просторі шляхом навчання і під час цього навчання автокодер навчається ігнорувати розбіжності або шум, присутній у вхідних даних. Автокодери використовуються в різноманітних завданнях, таких як класифікація зображень, генеративне моделювання, розпізнавання обличчя тощо. Базова архітектура Автокодер зазвичай складається з 2 частин:

- кодера, який відображає вхідні дані в латентний простір;

					КВРКІ 200244.20.02.22 ПЗ	Арк. 28
Зм.	Арк.	№ докум.	Підпис	Дата		

– декодера, який декодує точки в латентному просторі для реконструкції вхідних даних. Замість того, щоб просто відверто дублювати вхідні дані, автокодер призначений для деконструювання та реконструкції вхідних даних і, таким чином, навчається розпізнавати лише важливі частини даних.

На рисунку 2.7 представлено архітектуру згорткового автокодера.

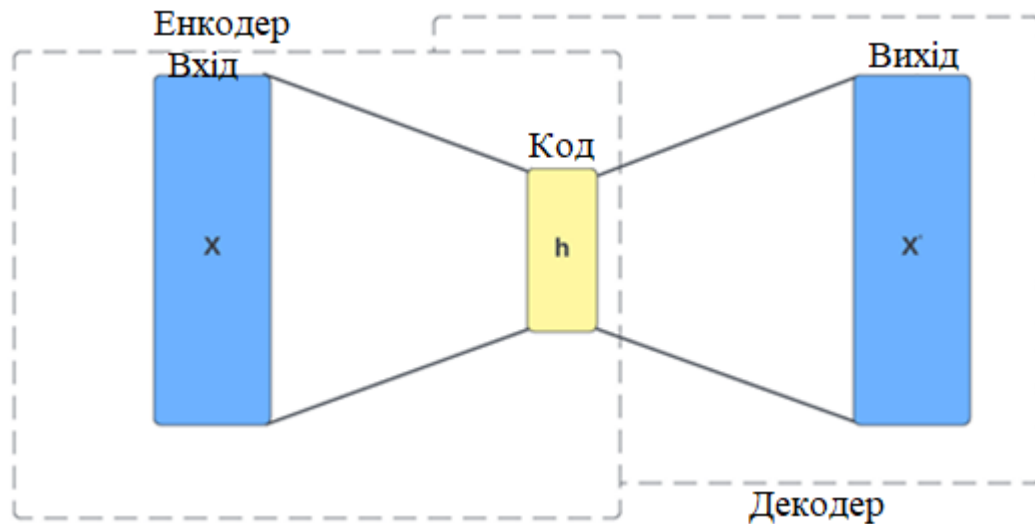


Рисунок 2.7 - Архітектура згорткового автокодера

Автокодер із шумоподавленням (DAE) за архітектурою досить схожий на стандартний автокодер, за винятком того, що він вносить шум у вхідні зображення, присутні в наборі даних під час навчання та перевірки. Введений шум має випадковий характер. Враховуючи функцію T , що генерує випадковий шум, їй надається вхідний сигнал x , який потім перетворюється на зашумлену версію x , позначену як $T(x)$. Цей $T(x)$ потім використовується як вхідні дані для нейронної мережі.

Ефективний DAE повинен обробляти будь-яке зашумлене зображення та реконструювати вихідне зображення, одночасно усуваючи шум. Розумний спосіб створення моделі передбачає використання згорткових шарів у нейронній мережі замість щільних шарів. Це зроблено тому, що згорткові шари, як відомо, вивчають дійсно хороші представлення та особливості зображень через їхню роботу, тоді як повнозв'язані шари не можуть цього зробити.

Архітектура, розроблена як для повністю підключеного автокодера, так і для згорткового автокодера, описана нижче. Ця модель приймає вхідні дані розміром $28 \times 28 \times 1$ і зводить його до масиву розмірністю 784, який тепер стає вхідними. Ці вхідні дані передаються на щільний шар, що складається з 64 нейронів, які, зрозуміло, повністю підключені до вхідних даних. Вихід цього щільного шару передається на інший щільний шар, що складається з 784 нейронів. Число 784 вибрано таким чином, щоб змінити форму результату назад у зображення 28×28 пікселів. Вихідні дані другого щільного шару змінюються назад у розмірний масив $28 \times 28 \times 1$, такої ж форми, як і вхідні дані, що позначає остаточний вихід кодера. На рисунку 2.8 представлено схему повністю підключеного автокодувальника.

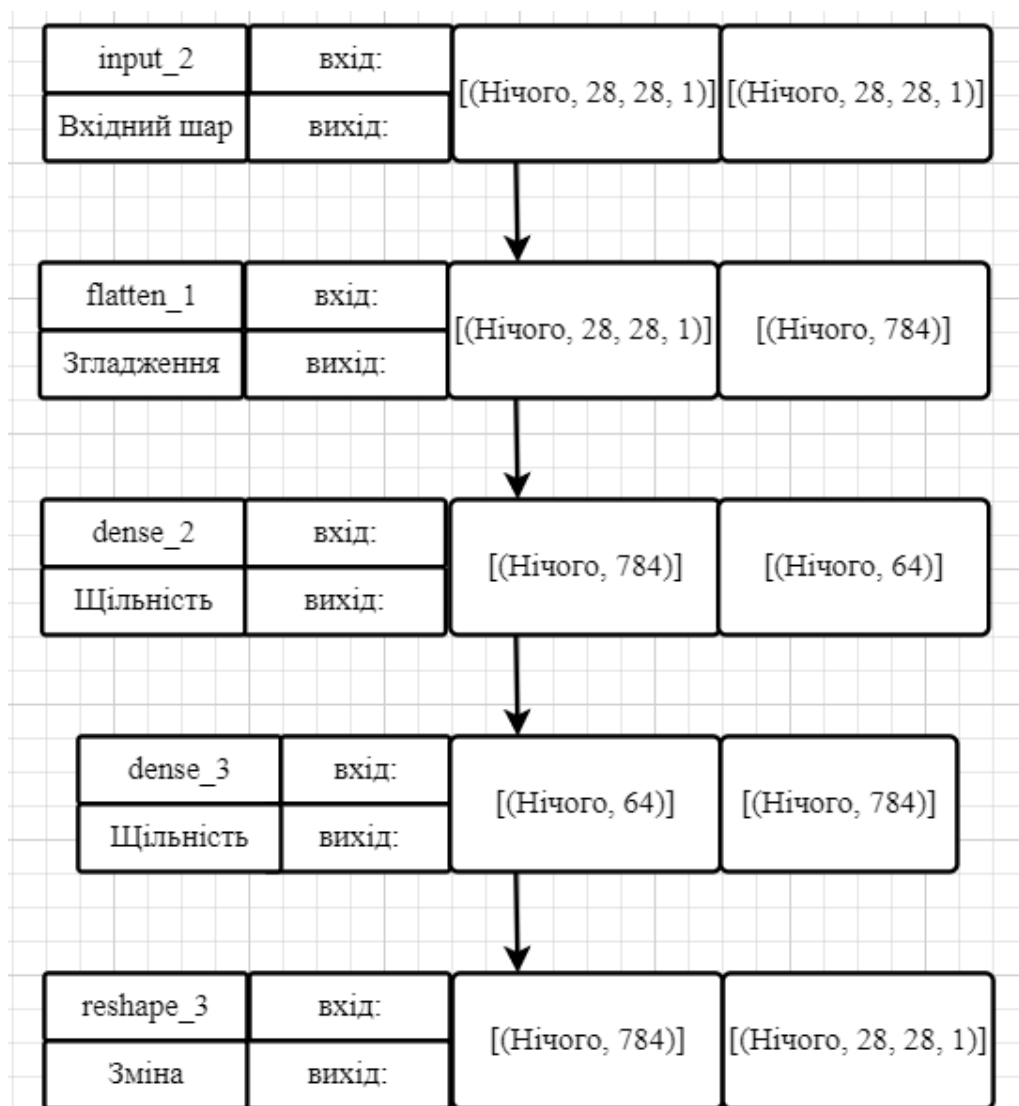


Рисунок 2.8 - Схема повністю підключеного автокодувальника

Згортковий автокодер використовує той самий принцип, що й звичайний автокодер, з тією лише різницею, що замість щільних шарів використовуються згорткові шари. Вхідний сигнал розмірності $28*28*1$ пропускається через 2 набори двовимірних згорткових шарів (32 фільтри, вагова матриця $3*3$, активація relu) і кодує вхідні дані в прихований простір розмірності $7*7*32$. Потім латентні координати використовуються як вхідні дані для шарів транспонування Conv2D (32 фільтри, вагова матриця $3*3$, активація relu), щоб збільшити масштаб зображення до початкової форми.

Деталі кожного рівня вказані в таблиці 2.1, в якій показуно загальну кількість параметрів, які можна навчити, для кожної моделі.

Таблиця 2.1– Деталі шару згорткового автокодера

Шар	Тип	Розмір ядра	Крок	Заповнення
1	Conv2D	3	1	незмінне
2	Pool(max)	2	1	незмінне
3	Conv2D	3	1	незмінне
4	Pool(max)	2	1	незмінне
5	TransConv2D	2	2	незмінне
6	TransConv2D	2	2	незмінне
7	Conv2D	3	1	незмінне

Тренувальні параметри вказані в таблиці 2.2.

Таблиця 2.2– Тренувальні параметри

Модель	Тренувальні параметри
Повністю підключений автоенкодер	101, 200
Згортковий автоенкодер	28, 353

На рисунку 2.9 представлено схему згорткового автокодера

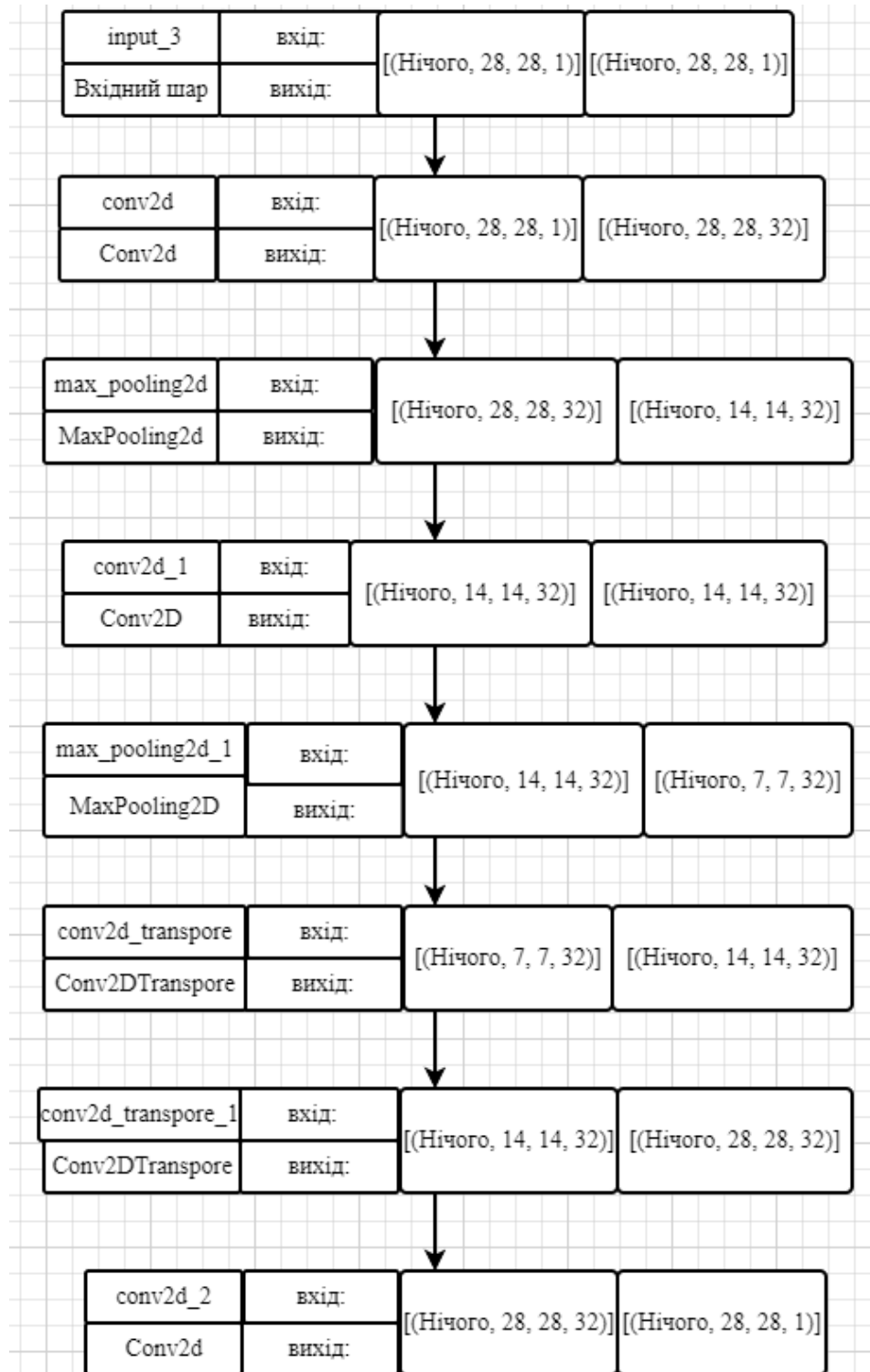


Рисунок 2.9 - Схема згорткового автокодера

Оскільки набір даних, який буде потрібен для виконання поточного завдання, вимагає шумових зображень, ми визначаємо функції для додавання випадкового

гаусового шуму до елементів набору даних. Новий набір даних тепер складається з зашумлених зображень і може бути візуалізований нижче. На рисунку 2.10 представлено зразки зображень із набору даних MNIST.

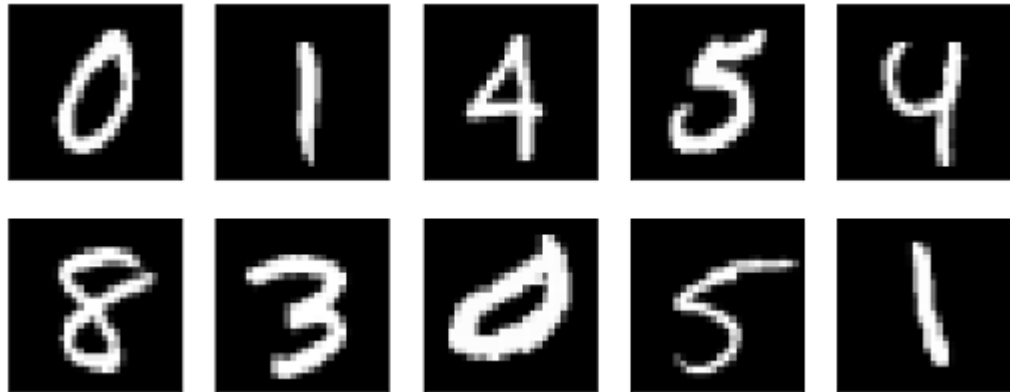


Рисунок 2.10 - Зразки зображень із набору даних MNIST

На рисунку 2.11 представлено зразки зображень із набору даних MNIST після додавання гаусового шуму.



Рисунок 2.11 - Зразки зображень із набору даних MNIST після додавання гаусового шуму

Нейронні мережі моделюються відповідно до зображених на рисинку 2.8 і рисунку 2.9 з використанням бібліотек tensorflow і keras. Навчання буде проводитися на графічній карті Google Colab Tesla K80 із 2496 ядрами CUDA та 12 ГБ відеопам'яті GDDR5.

Кожна модель навчається протягом 20 епох, а втрати під час навчання та перевірки відображаються наприкінці кожної епохи. Обидві моделі використовують оптимізатор Адама, а функція втрат — «двійкова крос-ентропія». Час навчання для кожної моделі становив приблизно 100 секунд через те, що вхідні зображення мають малий розмір і, отже, не вимагають надто великої потужності обробки.

Остаточні значення втрат під час перевірки для кожної моделі вказані в таблиці 2.3.

Таблиця 2.3– Втрата моделі

Модель	Втрата перевірки
Повністю підключений автоенкодер	0, 2305
Згортковий автоенкодер	0, 0871

Хоча втрата обох моделей, здається, не сильно відрізняється за значенням, прогнози кожної моделі розповідають нам іншу історію. Прогнози, зроблені повністю підключеним автокодером, показані нижче разом із відповідними вхідними даними.

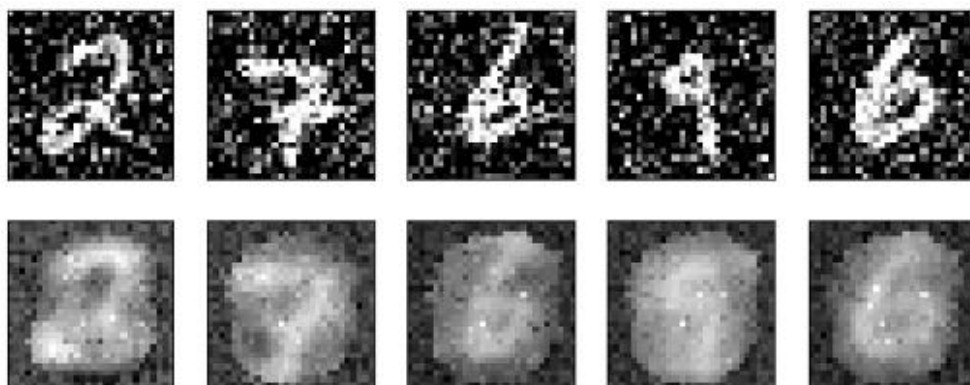


Рисунок 2.12 - Шумні входи (вгорі) разом із відповідними приглушеними прогнозами (внизу) повністю підключеного автокодера

На рисунку 2.12 представлено повністю підключений автокодер, який складається лише з щільних шарів, погано справляється зі зменшенням шумів, натомість він, здається, робить вхідний сигнал ще більш шумним. Це пов'язано з тим, що щільні шари не дуже добре вивчають функції та представлення для завдань, пов'язаних із зображеннями. У той час як передбачення разом із вхідними даними для згорткового автокодера результати представлено на рисунку 3.13, які показано нижче.



Рисунок 2.13 - Зашумлені входи (вгорі) разом із відповідними приглушеними прогнозами (внизу) згорткового автокодера

Різниця в прогнозах, зроблених двома моделями з точки зору вихідних даних, величезна і з рисунка 2.13 видно, наскільки краще згортковий автокодер справляється з усуненням шумів. Майже весь шум із вхідного сигналу видаляється, що забезпечує чітке зображення майже без шумів. Це пов'язано з високою здатністю згорткового шару зберігати значущі представлення вхідних даних під час навчання і, отже, відчувувати будь-який шум під час відображення зображення у прихованому просторі.

Таким чином, візуалізується завдання зменшення шуму зображення 28*28 пікселів у відтінках сірого за допомогою автокодерів. Проведено відмінності між двома різними архітектурами автокодерів, а саме щільним автокодером і згортковим автокодером, за допомогою яких можна переконатися, що використання згорткових шарів є дуже важливим у випадку зменшення шуму

зображення за допомогою глибокого навчання. Таким чином, більш складне усунення шумів із зображеннями RGB можна виконати шляхом створення глибшої моделі та більш відповідного розміру прихованого простору. Майбутні дослідження можуть включати створення моделі для усунення шумів на зображеннях вищого калібру та тих, які містять кілька каналів.

2.3 ASU-CNN: ефективна глибока архітектура для класифікації зображень і візуалізації функцій

Штучний інтелект здійснив технологічну революцію в дослідженнях і застосуванні в різних галузях науки і промисловості. Штучні нейронні мережі як інструменти штучного інтелекту замінюють або більш обережно обганяють людський інтелект і обов'язки в різних аспектах. ШНМ широко використовуються в чисельному моделюванні задач, що виникають у різних галузях прикладних наук і промислових досліджень. У сучасну епоху комп'ютерний зір є однією з найцікавіших областей досліджень, яка має справу з різними завданнями, включаючи розпізнавання зображень, мови та дій, головним чином для соціальної безпеки. Більшість програм комп'ютерного зору покладаються на інформацію на основі зображень або відео для цілей розпізнавання чи класифікації. CNN є найкращою архітектурою DNN для роботи з даними на основі зображень. Завдяки корисності та високим вимогам до виконання різних завдань, CNN були розвинені в більш надійні та тонкі версії, такі як ResNets. Основний фокус досліджень у цій галузі базується на модифікаціях CNN з новими та більш ефективними функціями активації, функціями втрат та оптимізаторами. Вибір функцій втрат в основному залежить від характеру завдання, яке потрібно виконати, наприклад, проблеми розпізнавання обличчя людини можуть бути правильно поставлені шляхом модифікації втрат softmax, таких як Sphreface, і підходів до різних адитивних параметрів. Функції активації відіграють вирішальну роль у визначенні продуктивності будь-якої архітектури DNN. У минулому були введені різні

					КВРКІ 200244.20.02.22 ПЗ	Арк. 36
Зм.	Арк.	№ докум.	Підпис	Дата		

монотонні функції активації, але більшість із них зіткнулися з недоліками на основі градієнта під час зворотного поширення, які потім були пом'якшені та немонотонними функціями активації, такими як Swish та Mish. Усі ці функції активації були неосциляційними за своєю природою і дослідники не звертали уваги на використання неосциляційних функцій для мереж. У випадку CNN ReLu вважається найефективнішим вибором для використання в згорткових шарах. У 2021 році впровадження GCU, коливальної немонотонної функції активації, розширило горизонт функцій активації, застосовних у різних архітектурах DNN. Функції коливальної активації можуть виконувати складні призначені завдання з меншою кількістю нейронів і, таким чином, є обчислювально дешевшими, ніж неосциляційні активації. Коливальна та немонотонна функція активації ASU спочатку була запропонована для DNN, що моделює нелінійні динамічні системи, і має потенційну обчислювальну перевагу для проблем нелінійної еволюції, мікроелектромеханічних систем (MEMS) та механічних коливань. ASU, заданий, перевершив GCU у відновленні періодичної та нелінійної динаміки MEMS(ASU)

У цій роботі буде пояснено інтеграцію ASU в CNN для завдань класифікації зображень. Була запропонована проста архітектура CNN, яка використовує ASU як активацію як згорткового, так і щільного шарів. Класифікація виконується на загальнодоступному наборі даних CIFAR-10, який містить 60 000 зображень, що належать до 10 різних класів. CIFAR-10 складається з кольорових зображень з роздільною здатністю 32x32 і містить 6000 зображень на клас. Будучи проблемою багатокласової класифікації, вихідний рівень мережі активується через softmax. Набір даних має неперекриваючі класи з десяти різних легко розпізнаваних об'єктів, тому в мережі використовується розріджена категоріальна втрата крос-ентропії. Щоб оптимізувати втрати, використовувався з відпалом швидкості навчання, де швидкості навчання спадають експоненціально. Реалізація виконується через фреймворк Tensorflow.

CNN вважаються найбільш пильними архітектурами для роботи з даними на основі зображень. Існує ієрархія згорткових шарів, що включає шари об'єднання,

					КВРКІ 200244.20.02.22 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

за якими слідує щільна мережа. Зображення надходять на перший згортковий рівень мережі, де за допомогою різних фільтрів виявляються численні характеристики вхідних зображень і цей процес триває по всій згортковій частині мережі. На рисунку 2.14 представлена загальна структура CNN, яка використовується в цій роботі для класифікації CIFAR-10.

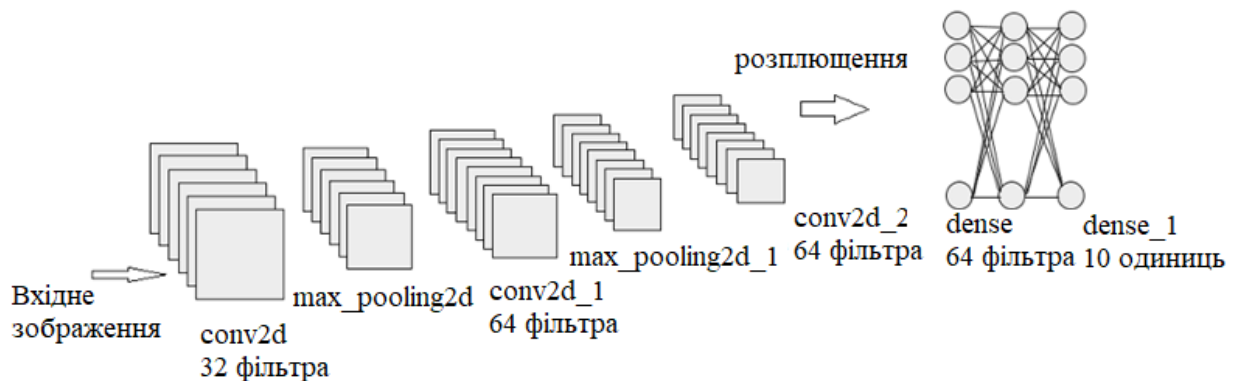


Рисунок 2.14 - CNN має три згорткові шари з різними фільтрами для вилучення особливостей вхідних зображень і щільну мережу в кінці для класифікації

Згортковий рівень складається з ряду фільтрів, відповідальних за виявлення різних типів ознак, присутніх у вхідних даних. Ці фільтри ковзають по вхідних даних для збору зваженої суми. Розроблена архітектура має три таких згорткових шару, тобто $l = 1, 2, 3$, де перший має 32 фільтри, а другий і третій шари мають 64 фільтри з дійсним заповненням і кроком 1. У цій роботі розроблений CNN застосовує ASU, який є лінійним виходом згортки. На рисунку 2.15 представлено ASU, його першу та другу похідні, які використовуються під час зворотного поширення.

ASU показано червоною кривою та імітує однакову коливальну поведінку як для строго позитивних, так і для строго негативних сторін домену і коливання мають тенденцію посилюватися для більших вхідних значень. Перша похідна, як показано зеленою кривою на рисунку 2.15, поводить себе точно протилежно вздовж обох сторін від нуля, і, просуваючись далі вздовж області, коливання знову мають тенденцію посилюватися. Друга похідна, як показано сивою кривою також

коливається та збільшується для вищих входів і поводить симетрично вздовж нуля області.

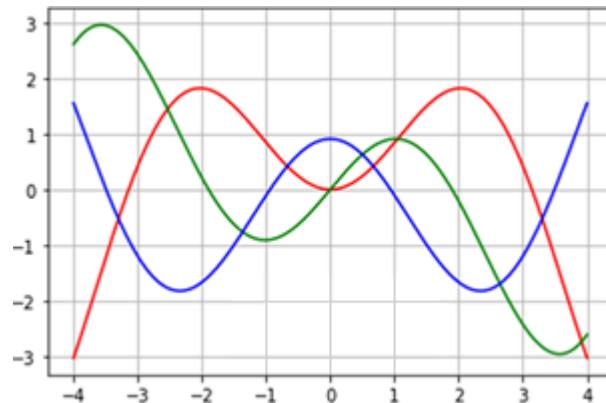


Рисунок 2.15 - Графіки ASU разом із першою та другою похідними

Шари об'єднання в основному використовуються для зменшення розміру карт функцій з метою зменшення витрат на обчислення. Максимальне об'єднання є загальноприйнятою практикою, яка вибирає найбільш помітне значення функції у вікні об'єднання. У цій роботі для перших двох згорткових рівнів мережі встановлюється максимальний розмір об'єднання (2, 2) із кроком 2. Завдання виділення ознак виконується ієрархічно на згорткових рівнях мережі. На рисунку 2.16 та рисунку 2.17 представлені карти функцій, отримані з різних рівнів розробленої архітектури CNN.

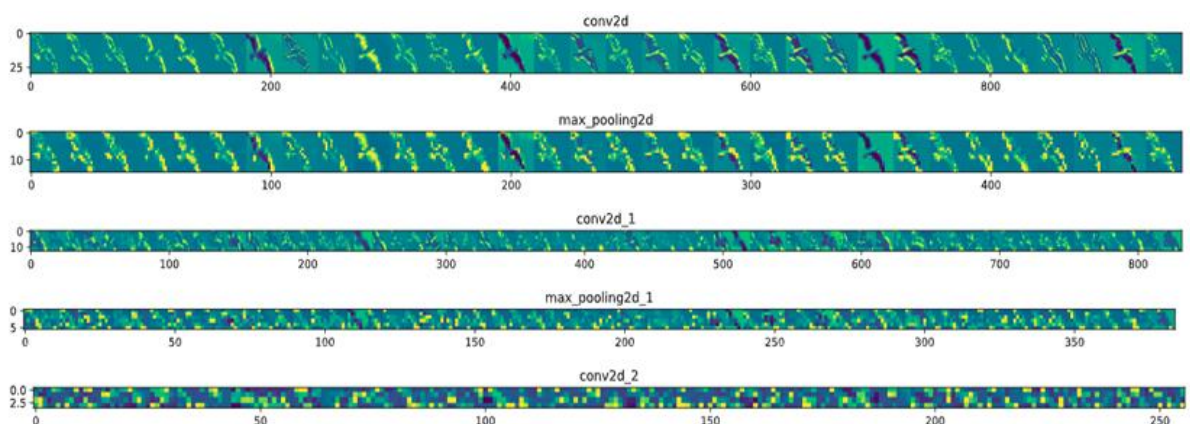


Рисунок 2.16 - Візуалізація карт функцій із різних шарів CNN, коли зображення птаха надходить у модель візуалізації

Перший згортковий шар, тобто conv2d, застосовує 32 фільтри, які виявляють основні особливості, такі як краї, горизонтальні або вертикальні візерунки, присутні на крилах птаха та його хвості. Заглиблюючись у conv2d_1, ми зосереджуємося на спостереженні відносно складних шаблонів, присутніх у вхідних даних. На 3-му рівні conv2d_2 мережі масштабують більше, щоб отримати навіть незначні деталі шаблонів, іншими словами, він зосереджується на виявленні особливостей високого рівня.

Така сама ієрархія виявлення ознак працює для введення зображення кота. Можна помітити, що початкові шари виділяють низькорівневі риси, зосереджуються на загальних шаблонах, присутніх на зображенні, таких як наявність хвоста, ніг, країв і вигинів тіла. Поглиблюючись до conv2d_2 у мережі, вилучення функцій стає складнішим і конкретнішим.

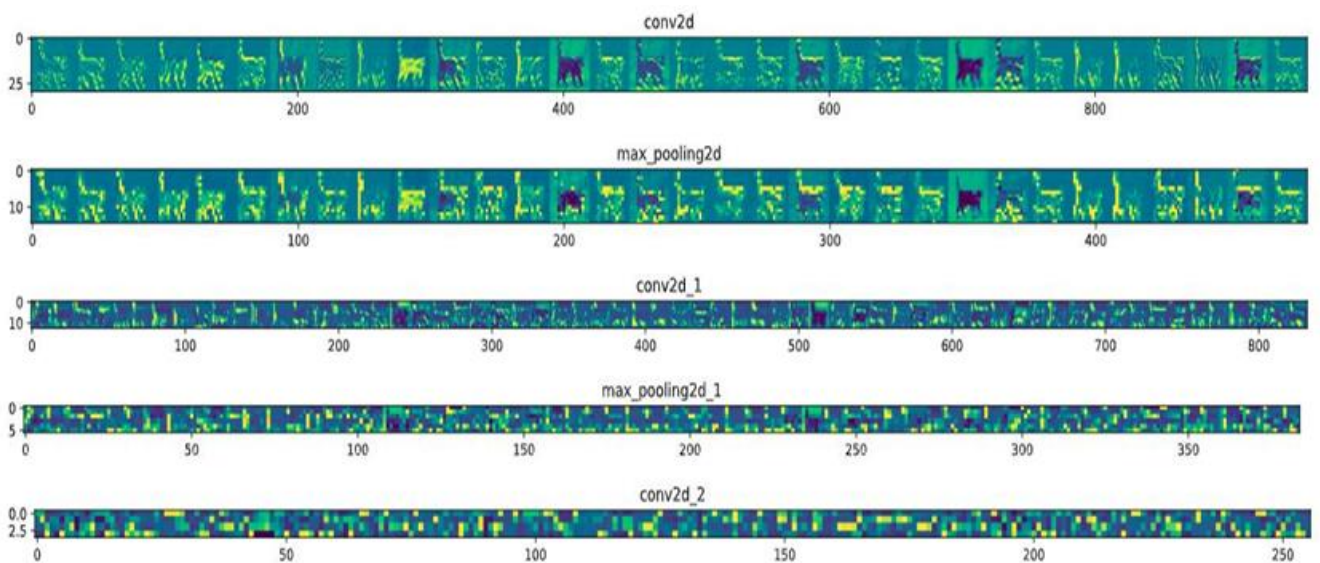


Рисунок 2.17 - У випадку зображення кота візуалізація карт функцій з різних згорткових шарів мережі

Це повністю зв'язані шари, що нагадують моделі багаторівневих перцептронів (MLP). Вихід з останнього згорткового шару вирівнюється та надсилається до щільних шарів, які працюють як стандартна мережа MLP для

класифікації. Спроектowana мережа має два щільних шари: один прихований шар із активацією ASU, а інший – вихідний рівень із функцією активації softmax.

Розроблена CNN, яка показана на рисунку 2.14, складається з трьох згорткових шарів, за якими слідує щільна мережа з одним прихованим шаром і вихідним шаром з 10 нейронів. Ваги мережі ініціалізуються уніфікованим ініціалізатором Xavier. Для навчання мережа використовує Sparse Categorical Cross-entropy loss. Мережа налаштована на навчання протягом 20 епох.

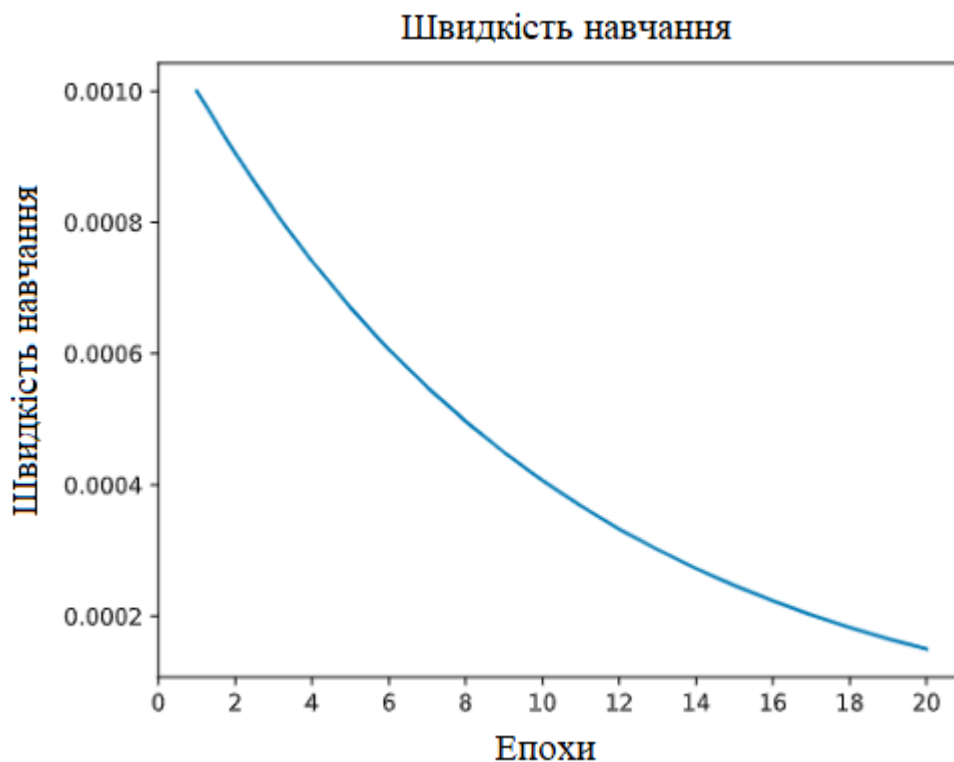


Рисунок 2.18 - Експоненціальний спад швидкості навчання, починаючи з 0,001 зі швидкістю спаду 0,1, досягаючи 0,00015 на 20-й епі

Початкова швидкість навчання встановлена на 0,001 з експоненціальною швидкістю спаду 0,1, показана на рисунку 2.18.

Експериментальні результати запропонованої мережі, що використовує функцію активації ASU для класифікації набору даних CIFAR-10, узагальнені у цій роботі. Мережа була оцінена на основі її продуктивності з точки зору часу навчання, точності даних навчання, історії втрат під час навчання та точності даних

тестування. Під час одного прогону, що складається з 20 епох, мережа завершила тренувальний процес за 3 хвилини 2 секунди. Детальна шарова структура розробленої архітектури CNN представлена на рисунку 2.19.

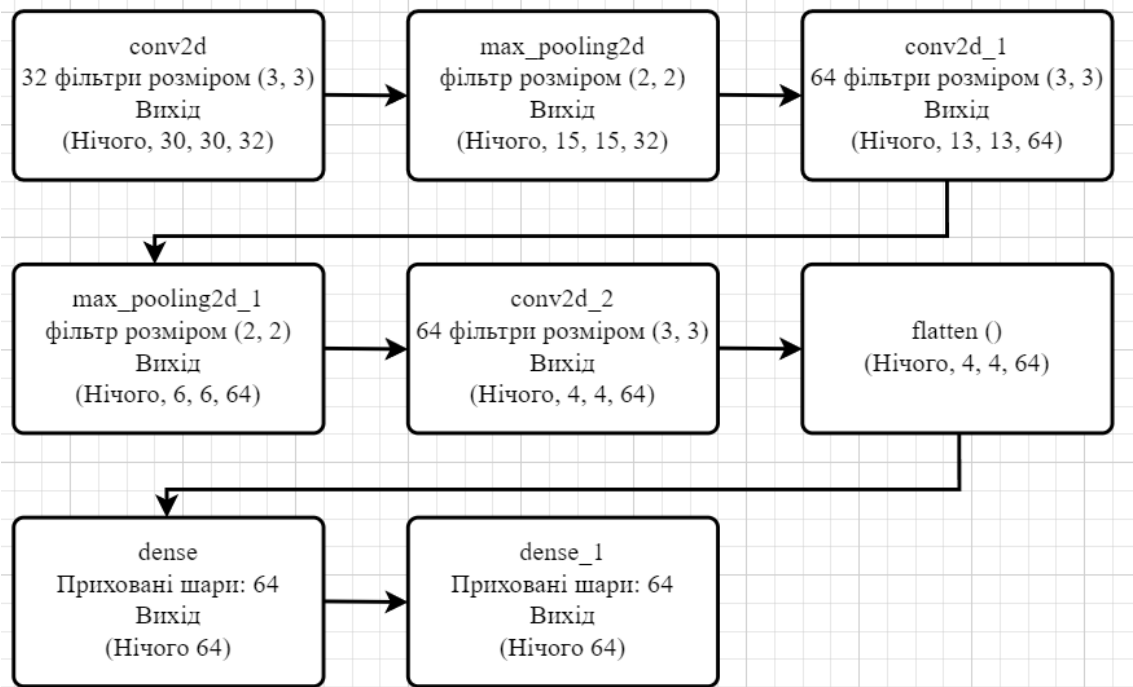


Рисунок 2.19 - Експериментальна побудова CNN з назвами шарів, фільтрами та вихідними розмірами карт функцій, що виходять із кожного згорткового шару

На рисунку 2.20 показано історію втрат мережі під час процесу навчання. З графіка видно, що зі збільшенням кількості епох втрати при навчанні поступово зменшувалися. Це означає, що мережа покращила свою здатність мінімізувати розбіжності між прогнозованими та фактичними значеннями, оскільки вона проходила більше ітерацій навчання. Крім того, мережа підтримувала досить низькі втрати перевірки, що вказує на її здатність до узагальнення невидимих даних. Графік ілюструє прогресування обох точностей протягом усього процесу навчання. Примітно, що мережа постійно покращувала свою точність даних навчання, досягнувши високого значення 90,15%. Тим часом точність перевірки також продемонструвала задовільну продуктивність, що свідчить про те, що мережа може ефективно узагальнювати свої знання для нових екземплярів. При

оцінці даних тестування мережа досягла точності 70,15%. Це вказує на його здатність правильно класифікувати раніше невидимі зразки з набору даних CIFAR-10, хоча й з відносно нижчою точністю порівняно з навчальними даними. Можна виконати подальший аналіз і оптимізацію, щоб підвищити продуктивність мережі на даних тестування та потенційно підвищити їх точність. Коротше кажучи, запропонована мережа, що використовує функцію активації ASU, продемонструвала багатообіцяючу продуктивність у класифікації набору даних CIFAR-10. Вона продемонструвала ефективне навчання, про що свідчить зменшення втрат під час навчання та підвищення її точності. Крім того, мережа продемонструвала хороші можливості узагальнення з розумною втратою перевірки. Хоча точність даних тестування була нижчою, ніж даних навчання, вона все ж досягла гідної поваги точності 70,15%, що свідчить про його здатність класифікувати невидимі зразки. Мережа з ASU потребувала менших обчислювальних витрат, оскільки існує лише одна трансцендентна функція та потрібне одне обчислення множення. Таким чином, навчання мережі з ASU було набагато швидшим, а результуюча точність дуже приваблива.

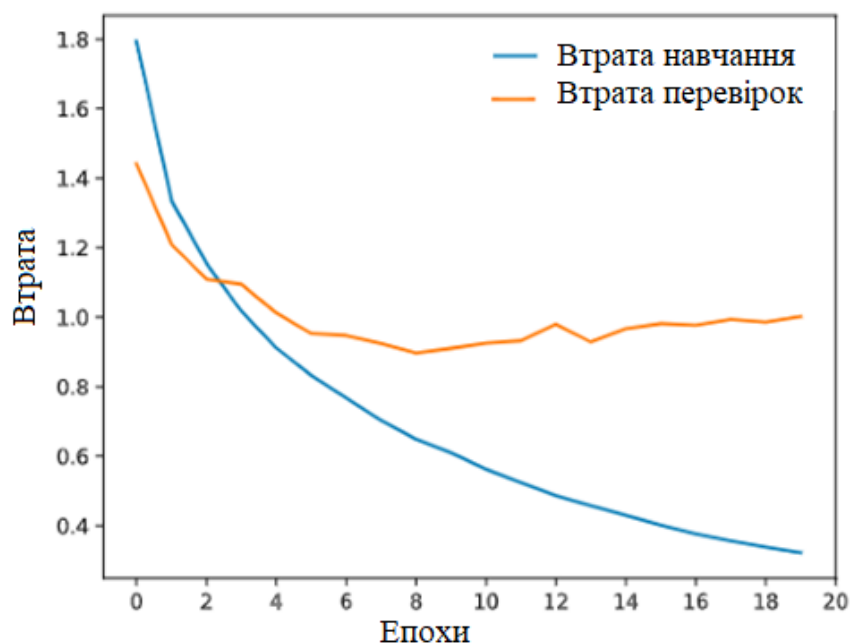


Рисунок 2.20 - Історія втрат навчання та перевірки на CIFAR-10 у разі активації ASU за 20 епох

На рисунку 2.21 представлено графічну ілюстрацію навчання та перевірки точності CNN на CIFAR-10.

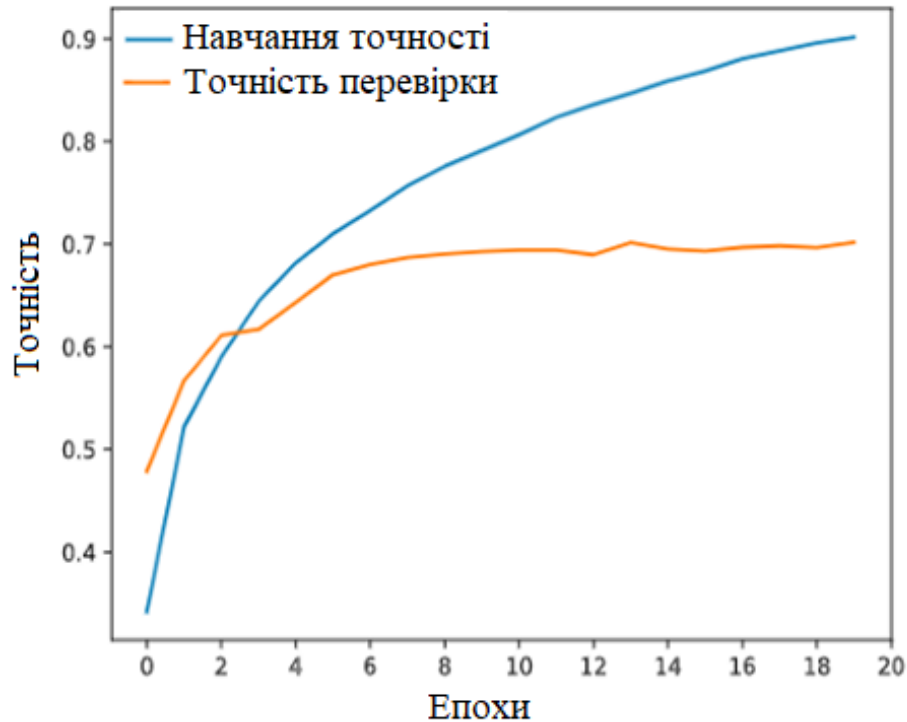


Рисунок 2.21 - Графічна ілюстрація навчання та перевірки точності CNN на CIFAR-10

Розроблена CNN, що містить ASU на всіх рівнях, була оцінена для CIFAR-10 і продемонструвала багатообіцяючу продуктивність. Візуалізація карт функцій з усіх шарів згортки, швидкість навчання та обчислювальна можливість ASU також засвідчують пильність, розсудливість та ефективність запропонованої архітектури. Результати оцінювання виявили багатообіцяючу продуктивність, вказуючи на те, що ASU має потенціал для підвищення точності та ефективності моделей CNN для завдань класифікації зображень. У майбутньому цю функцію активації можна перевірити для інших програм комп'ютерного зору. Отже, навчання мережі з ASU було набагато швидшим, а результуюча точність дуже приваблива та мережа продемонструвала хороші можливості узагальнення з розумною втратою перевірки.

2.4 Архітектура, що поєднує згорткову нейронну мережу (CNN) і машина опорних вікторів (SVM) для класифікації зображень

Низка досліджень із застосуванням підходів до глибокого навчання засвідчила найсучаснішу продуктивність у значній кількості завдань. Вони включають, але не обмежуються цим, класифікацію зображень, цей підхід, а також обмеження на бінарну класифікацію. Оскільки SVM прагне визначити оптимальну гіперплощину, що розділяє два класи в наборі даних, мультиноміальний випадок, здається, ігнорується. З використанням SVM у мультиноміальній класифікації випадок стає «один проти всіх», у якому позитивний клас представляє клас з найвищим балом, тоді як решта представляють негативний клас. У цій роботі просимульовано архітектуру, яка поєднує згорточну нейронну мережу (CNN) і лінійну SVM для класифікації зображень. Однак CNN, що використовується в цьому дослідженні, є простим 2-згортковим рівнем із моделлю Max Pooling, на відміну від відносно більш складної моделі та попередньої обробки (Google TensorFlow використовувався для реалізації алгоритмів глибокого навчання в цьому дослідженні).

MNIST — це стандартний набір даних класифікації рукописних цифр, який широко використовується для порівняльного аналізу моделей глибокого навчання. Це класифікаційна задача з 10 класів, яка містить 60 000 навчальних прикладів і 10 000 тестових випадків – усі у відтінках сірого. Проте стверджується, що набір даних MNIST є «занадто простими» і «надмірно використовуваними», і «вони не можуть представляти сучасні завдання CV [Computer Vision]». Було запропоновано набір даних Fashion-MNIST. Зазначений набір даних складається із зображень статей Zalando, які мають такий же розподіл, таку ж кількість класів і той самий колірний профіль, що й MNIST.

Обидва набори даних використовувалися без попередньої обробки, наприклад, нормалізації чи зменшення розмірності, які показані нижче.

					КВРКІ 200244.20.02.22 ПЗ	Арк. 45
Зм.	Арк.	№ докум.	Підпис	Дата		

Розповсюдження набору даних для MNIST і FashionMNIST вказані в таблиці 2.4.

Таблиця 2.4— Розповсюдження набору даних для MNIST і FashionMNIST

Набір даних	MNIST	Fashion-MNIST
Навчання	60, 000	10, 000
Тестування	60, 000	10, 000

Конволюційна нейронна мережа (CNN) — це клас штучних нейронних мереж із глибоким передаванням, який зазвичай використовується у проблемах комп’ютерного зору, таких як класифікація зображень. Відмінністю CNN від «простої» мережі багатошарового персептрона (MLP) є використання згорткових шарів, об’єднання та нелінійності, такі як tanh, сідmoid і ReLU. Згортковий шар (позначається CONV) складається з фільтра, наприклад, $5 \times 5 \times 1$ (5 пікселів для ширини та висоти та 1, оскільки зображення мають градації сірого). Інтуїтивно кажучи, рівень CONV використовується для «ковзання» по ширині та висоті вхідного зображення та обчислення скалярного добутку області введення та параметрів навчання ваги. Це, у свою чергу, створить двовимірну карту активації, яка складається з відповідей фільтра в заданих регіонах. Отже, рівень об’єднання (позначений як POOL) зменшує розмір вхідних зображень відповідно до результатів фільтра CONV. У результаті кількість параметрів у моделі також зменшується, що називається зменшенням вибірки. Крім того, порівняно з об’ємними обчисленнями, які вимагають tanh і сідmoid, ReLU реалізується шляхом простого встановлення порогових значень матриці на нуль. Найпоширенішою функцією активації в наші дні є функція ReLU. ReLU зазвичай використовується над tanh і сідmoid, оскільки було виявлено, що вона значно прискорює конвергенцію стохастичного градієнтного спуску порівняно з двома іншими функціями. На рисунку 2.22 представлено функцію активації Rectified Linear Unit (ReLU), яка створює 0 як вихідний сигнал, коли $x < 0$, а потім виробляє лінійне значення з нахилом 1, коли $x > 0$.

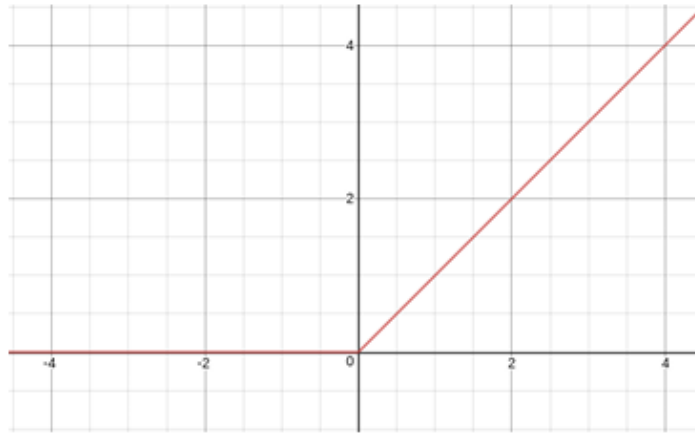


Рисунок 2.22 - Функція активації Rectified Linear Unit (ReLU), яка створює 0 як вихідний сигнал, коли $x < 0$, а потім виробляє лінійне значення з нахилом 1, коли $x > 0$

У цій роботі модель BaseC реалізована із такою архітектурою:

1. INPUT: $32 \times 32 \times 1$
2. CONV5: 5×5 size, 32 filters, 1 stride
3. ReLU: $\max(0, h^\theta(x))$
4. POOL: 2×2 size, 1 stride
5. CONV5: 5×5 size, 64 filters, 1 stride
6. ReLU: $\max(0, h^\theta(x))$
7. POOL: 2×2 size, 1 stride
8. FC: 1024 Hidden Neurons
9. DROPOUT: $p = 0.5$
10. FC: 10 Output Classes

На 10-му рівні CNN замість традиційної функції softmax із функцією перехресної ентропії (для обчислювальних втрат) реалізовано L2-SVM.

Експерименти для цього дослідження склалися з двох частин: перший етап навчання та другий тестовий приклад. Моделі CNN-SVM і CNN-Softmax використовувалися на MNIST і Fashion-MNIST. У цьому дослідженні розглядалися лише точність навчання, втрати під час навчання та точність тесту. Усі

експерименти в цьому дослідженні проводилися на портативному комп'ютері з процесором Intel Core(TM) i5-6300HQ @ 2,30 ГГц x 4, 16 ГБ оперативної пам'яті DDR3 і графічним процесором NVIDIA GeForce GTX 960M 4 ГБ DDR5.

Гіперпараметри, що використовуються для Cnn-Softmax у моделях CNN SVM вказані в таблиці 2.5.

Таблиця 2.5– Гіперпараметри, що використовуються для Cnn-Softmax у моделях CNN SVM.

Гіперпараметри	Cnn-Softmax	CNN SVM
Розмір партії	128	128
Відсоток відсіву	0,5	0,5
Швидкість навчання	1e-3	1e-3
Кроки	100000	100000
SVM C	N/A	N/A

Гіперпараметри, які вказано у таблиці 2.5, були призначені вручну та використовувалися для експериментів як у MNIST, так і в FashionMNIST.

На рисунку 2.23 представлено точність навчання класифікації CNN-Softmax і CNNSVMonimage за допомогою MNIST, а на рисунку 2.24 представлено архітектуру, що поєднує згорткову нейронну мережу (CNN) і опорну векторну машину (SVM) для класифікації зображень. Нижчу будуть наведені графіки навчання точності, які було зроблено за допомогою MNIST та Fashion – MNIST та буде показано, що модель CNN-Softmax мала середню точність навчання 98,47% і середню втрату навчання 0,136794931, а модель CNN-SVM мала середню точність навчання 97,67 % і середню втрату навчання 0,13%, разом ці моделі зробили 10000 кроків.



Рисунок 2.23 – Навчання точності CNN-Softmax і CNN-SVM на класифікації зображень за допомогою MNIST

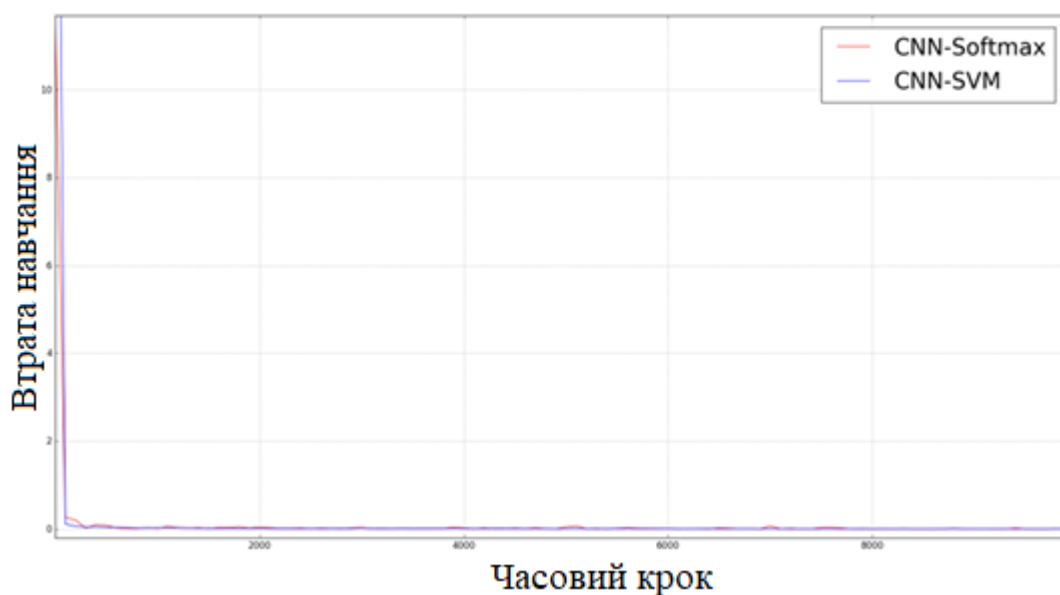


Рисунок 2.24 – Втрати навчання CNN-Softmax і CNN-SVM на класифікацію зображень

Їхня втрата навчання, зробивши 10 000 кроків, обидві моделі змогли завершити тренування за 4 хвилини 16 секунд. Модель CNN-Softmax мала середню точність навчання 98,4765625% і середню втрату навчання 0,136794931, тоді як модель CNN-SVM мала середню точність навчання 97,671875% і середню втрату

навчання 0,268976859. На рисунку 2.25 представлено навчання точності CNN-Softmax і CNN-SVM на класифікації зображень за допомогою Fashion-MNIST.



Рисунок 2.25 – Навчання точності CNN-Softmax і CNN-SVM на класифікації зображень за допомогою Fashion-MNIST

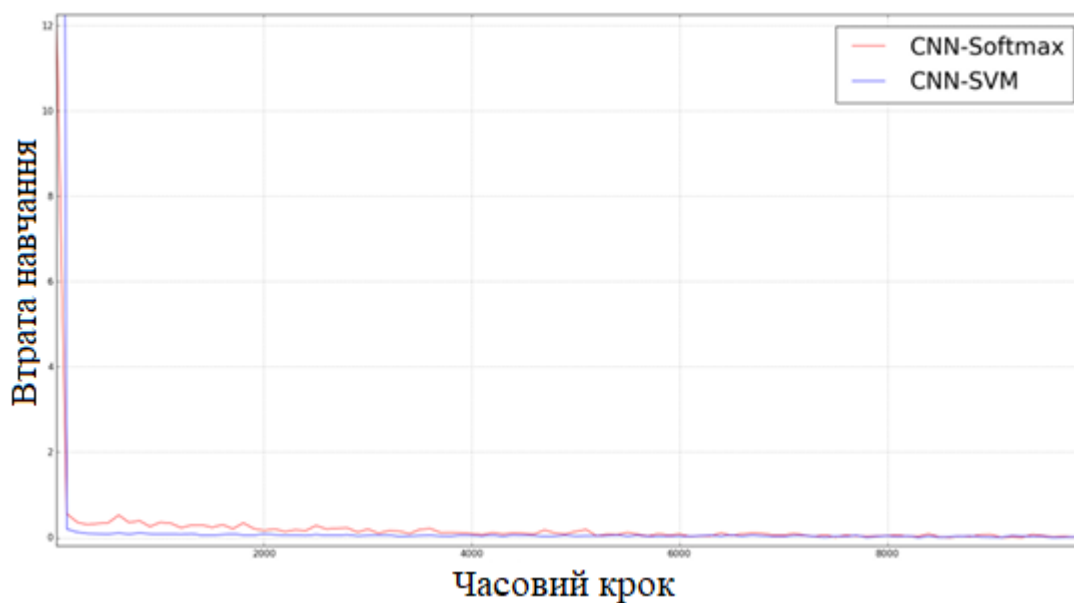


Рисунок 2.26 – Втрата навчання CNN Softmax і CNN-SVM на класифікацію зображень за допомогою Fashion MNIST

На рисунку 2.23 показано точність навчання класифікації CNN-Softmax і CNNSV Monimage за допомогою MNIST, тоді як на рисунку 2.26 показано втрати їх навчання. Зробивши 10 000 кроків, модель CNN-Softmax змогла завершити навчання за 4 хвилини 47 секунд, тоді як модель CNN-SVM могла завершити навчання за 4 хвилини 29 секунд. Модель CNN-Softmax мала середню точність навчання 94% і середню втрату навчання 0,259750089, тоді як модель CNNSVM мала середню точність навчання 90,15% і середню втрату навчання 0,793701683.

Перевірка точності класифікації зображень CNN-Softmax і CNN-SVM за допомогою MNIST і Fashion-MNIST вказані в таблиці 2.6.

Таблиця 2.6 – Перевірка точності класифікації зображень CNN-Softmax і CNN-SVM за допомогою MNIST і Fashion-MNIST

Набір даних	CNN-Softmax	CNN-SVM
MNIST	99,23%	99,04%
FASHION-MNIST	91,86%	90,72%

Після 10 000 кроків навчання обидві моделі були перевірені на тестових прикладах кожного набору даних. Як показано з таблиці 2.6, обидва набори даних містять по 10 000 тестів кожен та показано точність тестування CNNSoftmax і CNN-SVM на класифікацію зображень за допомогою MNIST і Fashion-MNIST. Точність тесту на наборі даних MNIST не підтверджує висновки, оскільки саме CNN-Softmax мав кращу точність класифікації, ніж CNN-SVM. Цей результат можна пояснити тим фактом, що попередньої обробки даних не було, ніж у наборі даних MNIST. Крім того, маємо відносно більш складну модель і методологію, ніж проста процедура, яка виконана в цьому дослідженні. З іншого боку, точність тесту моделі CNN-Softmax відповідає результатам, оскільки обидві методології не передбачали попередньої обробки даних Fashion-MNIST.

Результати цього дослідження вимагають удосконалення методології для подальшої перевірки його перегляду на запропонованому CNN-SVM. Незважаючи на суперечність із висновками, кількісно кажучи, точність тестів CNN-Softmax і CNN-SVM майже однакова з відповідним дослідженням. Існує гіпотеза, що з попередньою обробкою даних і відносно більш складною базовою моделлю CNN результати повинні бути відтворені.

2.5 Висновки

У другому розділі було досліджено архітектуру та реалізацію UNet для сегментації зображень. Розглянуто поняття шум та як відбувається усунення шумів зображення за допомогою згорткового автокодувальника. Було розглянуто операцію згортки та застосування операції деконволюції. Проаналізовано використання згорткового автоенкодера та ReLU. Також було досліджено та проаналізовано архітектуру, яка поєднує CNN та SVM для класифікації зображень та ASU-CNN, як ефективну глибоку архітектуру для класифікації зображень та візуалізації функцій.

					КВРКІ 200244.20.02.22 ПЗ	Арк. 52
Зм.	Арк.	№ докум.	Підпис	Дата		

3 ВИКОРИСТАННЯ АВТОЕНКОДЕРУ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ У ФОТО

3.1 Навчений компресор зображень на основі автокодувальника

Останніми роками стиснення зображень із втратами є складною темою в області глибокого навчання. Дотепер було запропоновано декілька методів стиснення зображень за допомогою методів глибокого навчання, наприклад, наскрізні навчені нейронні мережі, нейронні мережі з постобробкою з ненеуронними кодеками або вдосконалення традиційних кодеків із використанням нейронних мереж. Структура автокодувальника була популярним вибором для наскрізного стиснення зображення. Частина кодера вчиться генерувати компактне представлення вхідного зображення, а частина декодера реконструює зображення, близьке до вхідного, на основі компактного представлення. Подібно до традиційних методів, продуктивність схеми стиснення зображення з втратами залежить від того, наскільки добре схема квантування та керування швидкістю змішана з цією структурою автокодера. Вимоги до пікселів при досягненні хорошого пікового співвідношення сигнал-шум (PSNR) і суб'єктивної якості. Було використано архітектуру автокодувальника від Менцера. Архітектура від Mentzer складається з набору симетричних шарів згортки та шарів залишкових блоків для стиснення та розпакування зображення. Кодер зменшує розмірність зображення до кількох карт функцій. Карти функцій містять щільно упаковані значення кольорів із зображення та можуть бути точно реконструйовані у вихідне значення за допомогою відповідного декодера. Кожна карта ознак додатково модулюється за допомогою карти важливості, яка контролює розподіл бітів між зразками ознак у карті функцій, критично визначаючи кінцеву швидкість стиснення вихідного зображення. Карти важливості генеруються набором навчених шарів згортки, які можуть ідентифікувати зразки важливих ознак зображення та створюють набір масок важливості для скорочення глибини карт функцій. Таким чином, остаточні карти функцій зберігають лише найбільш значущу інформацію про представлення

					КВРКІ 200244.20.02.22 ПЗ	Арк. 53
Зм.	Арк.	№ докум.	Підпис	Дата		

зображення з певною швидкістю. По суті, вони є компактним представленням вхідного зображення. Щоб подолати проблему навчання через операцію квантування, було застосовано техніку м'якого квантування і реалізовано пряму оцінку швидкості на основі значень масок важливості для цілей контролю швидкості. Ентропійний кодер, який стискає усічені карти функцій, розроблено на основі стандарту JPEG2000. Контекстна модель була налаштована відповідно до характеристик карти функцій, яка в більшості випадків містить усі нулі на найменшій бітовій площині. Відповіддю на цей виклик із стиснення навченого зображення (CLIC) на CVPR 2018 є наскрізний автокодер із залишковими блоками та картами важливості, який можна навчати. Зокрема, система має на меті відповідати 0,15 біт-пер. Відповідно до вказівок щодо тестування, PSNR використовується для вимірювання якості вихідного зображення. Його було порівняно із популярними стандартами стиснення зображень, такими як JPEG і BPG. Крім того, було використано MS-SSIM, який імітує суб'єктивну оцінку якості людини, як другий показник якості зображення.

Частина автокодувальника для створення карти ознак і реконструкції зображення адаптована. Входом для кодера є зображення розміром $B \times H$ у форматі RGB. Верхня частина представляє кодер E , а нижня частина представляє декодер D . Усі рівні в кодері — це операції лінійної згортки та нелінійної активації, а декодер шари реалізують зворотні операції. Маючи вхідне зображення x , кодер створює набір карт ознак. Нормалізація в кодері обмежує вхідне значення зображення $[-1, 1]$. Вихідний сигнал z кодера, обмежений діапазоном від 0 до 1, надходить до квантувача та мережі важливості. Квантувач Q створює представлення фіксованою комою, зберігаючи перші d бітів після коми. Зокрема, значення карт важливості є позитивними цілими числами в діапазоні від 1 до d , що вказує, скільки (найбільш значущих) бітів квантованої ознаки q слід зберегти. На рисунку 3.1 представлено карти важливості для вхідного зображення.

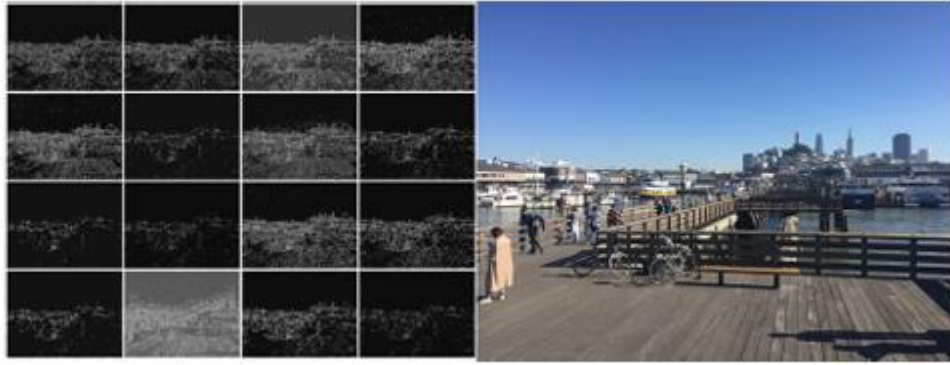


Рисунок 3.1 – Карти важливості (ліворуч) для вхідного зображення (праворуч)

Квантувач є критичним елементом у стисненні зображення з втратами, який розташований на кінці кодера. Крім того, квантовані значення були збережені з серією цілих індексів, щоб заощадити місце під час зберігання кількох квантованих значень. Ці індекси можуть бути представлені у двійковій формі, щоб відповідати операції маскування, описаній у наступному підрозділі.

Карта важливості розглядає проблему різної складності вмісту в різних частинах зображення. Крім того, деякі зображення легше стискати, ніж інші. Це тісно пов'язане з вимірюванням складності зображення. Карта важливості, створена нейронною мережею, зможе навчитися позначати складність різних областей зображення. Таким чином, при заданому бітрейті система стиснення виділяє більше бітів для складних областей.

Сітка важливості складається із залишкових блоків, за якими слідує квантувач для створення цілочисельної карти важливості imp для кожної карти функцій. Imp діє як маска для встановлення нульових бітів квантованої ознаки, які є менш значущими для загальної якості реконструкції і показано на рисунку 3.2. Карты важливості мають роздільну здатність і номер каналу, ідентичні картам функцій. Варто відзначити, що вони виробляються на основі одного і того ж виходу кодера. У процесі навчання квантувач у сітці важливості замінено на м'яке квантування. На рисунку 3.2 представлено двійкове арифметичне кодування карт ознак.

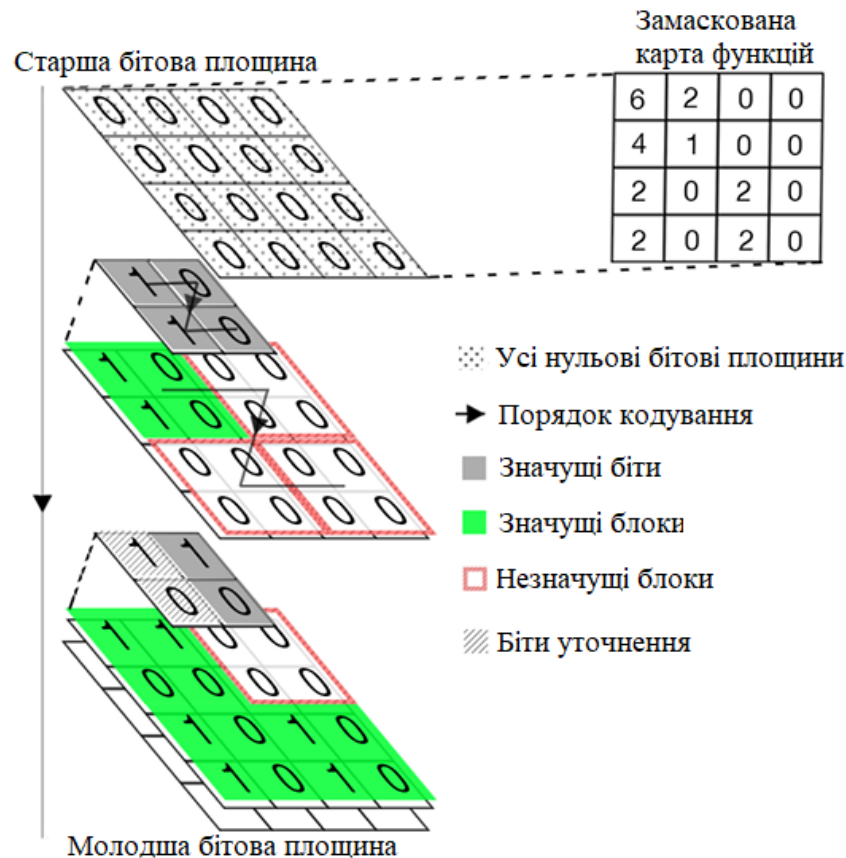


Рисунок 3.2 – Двійкове арифметичне кодування карт ознак

Втрата швидкості була оцінена, підсумовуючи всі значення карт важливості, $H(\text{imp})$. Для викривлення було обчислено як середню квадратичну помилку (MSE), так і індекс багатомасштабної структурної подібності (MS-SSIM) між оригінальним зображенням x і реконструйованим зображенням x .

Ентропійне кодування ґрунтується на стандарті JPEG2000. Він кодує двійкове представлення окремих остаточних карт ознак від старшої бітової площини (MSB) до найменш значимої (LSB). Крім того, на кожній бітовій площині біти кодування класифікуються на значущі та уточнюючі типи та кодуються поблочно за допомогою контекстно-адаптивного бінарного арифметичного кодера. Параметр розміру блоку є експериментальним. Як і у випадку з JPEG2000, окремі контекстні моделі використовуються для різних типів бітів, причому ймовірності контексту постійно оновлюються. Крім того, вводяться прапорці як на бітовій площині, так і на рівні блоку, щоб вказати всі нульові випадки. Ентропійний

декодер декодує стислий бітовий потік і відновлює двійкові карти ознак. Закодований бітовий потік також містить додаткову інформацію, необхідну декодеру. Рисунок 3.3 представляє двійкове арифметичне кодування карт ознак.

```

1: Initialize final feature maps  $Z$ 
2: for each feature maps do
3:   Initialize Block Flag  $B_{m,n}$  to 0
4:   Initialize Sample Flag  $S_{m,n,i,j}$  to 0
5:   Initialize All Zero Bitplane Flag  $AZB$  to 1
6:   for each bitplane  $k = \text{MSB to LSB}$  do
7:     if  $AZB == 1$  then
8:       Set  $AZB$  as per all zero bitplane state
9:       Encode  $AZB$ 
10:    if  $AZB == 0$  then
11:      for each block  $m, n$  do
12:        if  $B_{m,n} == 0$  then
13:          Set  $B_{m,n}$  as per significant block state
14:          Encode  $B_{m,n}$ 
15:        if  $B_{m,n} == 1$  then
16:          for each sample  $i, j$  do
17:            if  $S_{m,n,i,j} == 0$  then
18:              Set  $S_{m,n,i,j}$  as per significance state
19:              Encode significant bit
20:            else
21:              Encode refinement bit

```

Рисунок 3.3 – Двійкове арифметичне кодування карт ознак

Навчальний набір даних містить 10 000 зображень із конкурсу ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2012. Для навчання було випадково обрізано фрагменти розміром 128x128 із цих зображень як вхідні дані для автокодера. Для тестування та підтвердження був використаний набір даних Kodak PhotoCD.

Налаштування параметрів вказані в таблиці 3.1.

					КВРКІ 200244.20.02.22 ПЗ	Арк. 57
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.1 – Налаштування параметрів

Параметри	Значення
Розмір партії	10
Бітова глибина	4
Розмір блока	4 x 4
Швидкість навчання	[1e-4, 1e-6]
Номер каналу	16

Порівняння ефективності стиснення вказане в таблиці 3.2.

Таблиця 3.2 – Порівняння ефективності стисненн

	Біти/піксель	PSNR	MS-SSIM
JPEG	0, 155	22, 1	0,75668
BPG	0, 153	29, 2	0,9425
Дана система	0, 126	26, 3	0,9242

Навчання автокодеру та мережі важливості виконується в три послідовні фази. На першому етапі попереднього навчання автокодер навчається самостійно без мережі важливості. Це досягається лише мінімізацією спотворення реконструкції. На другому етапі сітка важливості включається в процес навчання, щоб мінімізувати цільову функцію, яка включає як спотворення, так і оцінку швидкості. Однак на цьому етапі попередньо навчений автокодер не оновлюється. На останньому етапі тонкого налаштування автокодер і мережа важливості оновлюються спільно. У таблиці 3.1 наведено детальні налаштування параметрів.

У цій роботі порівнюється продуктивність стиснення системи з продуктивністю JPEG і BPG на рівні 0,15 біт/с за PSNR і MS-SSIM. Тестовий набір даних взято з CLIC 2018. Варто зазначити, що числа PSNR обчислюються за

методом CLIC 2018: тобто, середня квадратична помилка для цих 287 тестових зображень спочатку обчислюється перед перетворенням PSNR. Для MS-SSIM були отримані числа шляхом усереднення значень MS-SSIM цих тестових зображень, причому MS-SSIM кожного тестового зображення отримано шляхом усереднення трьох кольорових компонентів. З рисунка 3.4 видно, що обидві системи VPGeandour значно перевершують JPEG з точки зору PSNR. Хоча VPG на 3 дБ кращий за точки зору PSNR, їх MS-SSIM близькі один до одного. На рисунку 3.7 наведено декілька прикладів зображень для порівняння суб'єктивної якості цих методів.

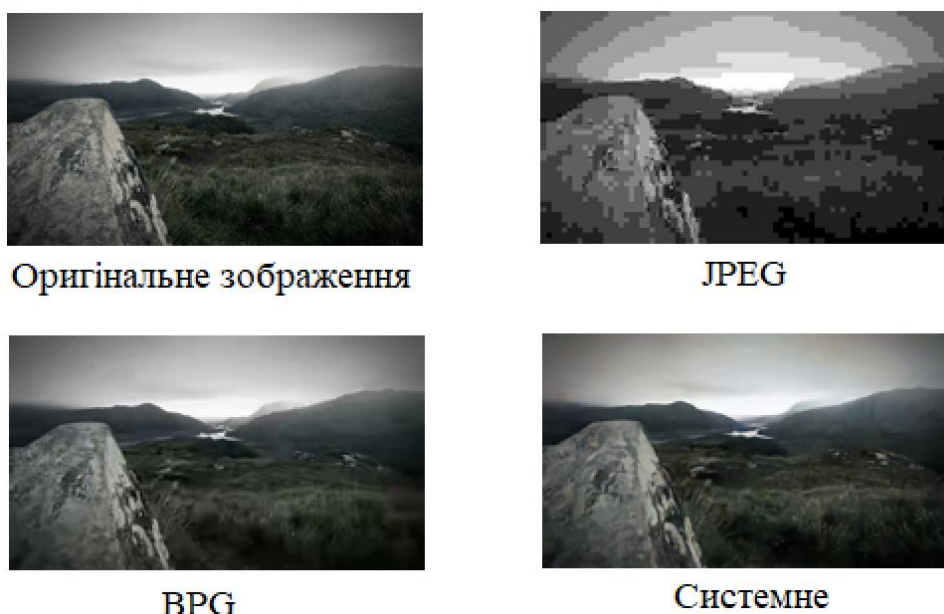


Рисунок 3.4 – Оригінальне зображення і реконструйовані зображення, створені JPEG, VPG та цією системою

Ця робота описує реакцію NCTU на CLIC2018. Вона представляє компресор навчених зображень на основі автокодувальника з поняттям карт важливості для розподілу бітів і керування швидкістю. Незважаючи на те, що вона значно перевершує JPEG, все ще є розрив у продуктивності порівняно з VPG, що свідчить про достатньо можливостей для подальшого вдосконалення. Серед іншого, вплив карт важливості на суб'єктивну якість заслуговує на подальше дослідження. У цьому випадку це спричиняє кольорові смуги на кількох зображеннях.

3.2 Swapping Autoencoder для глибокої обробки зображень

Глибокі генеративні моделі стали дедалі ефективнішими у створенні реалістичних зображень із випадково відібраних вихідних даних, але використання таких моделей для керованого маніпулювання існуючими зображеннями залишається складним завданням. Swapping Autoencoder - глибока модель, розроблену спеціально для маніпулювання зображеннями, а не для випадкової вибірки. Ключова ідея полягає в тому, щоб закодувати зображення на два незалежних компоненти та забезпечити, щоб будь-яка замінена комбінація відображалася на реалістичне зображення. Компоненти представляють структуру та текстуру, змушуючи один компонент кодувати статистику одночасних патчів у різних частинах зображення. Оскільки цей метод навчається за допомогою кодера, пошук прихованих кодів для нового вхідного зображення стає тривіальним, а не громіздким. У результаті цей метод дозволяє маніпулювати реальними введеними зображеннями різними способами, включаючи заміну текстур, локальне та глобальне редагування та векторну арифметику латентного коду. Експерименти на кількох наборах даних показують, що ця модель дає кращі результати та значно ефективніша порівняно з останніми генеративними моделями. На рисунку 3.5 представлено глибоку модель Swapping Autoencoder, яка вчиться відокремлювати текстуру від структури для завдань редагування зображень.

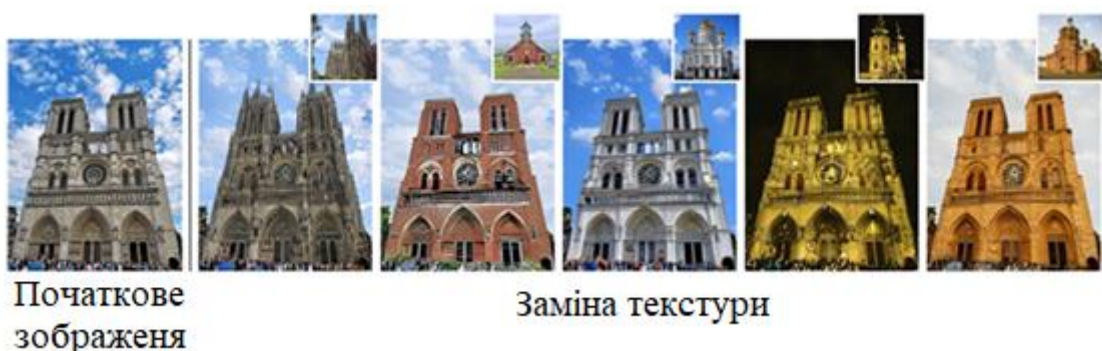


Рисунок 3.5 – Swapping Autoencoder, яка вчиться відокремлювати текстуру від структури для завдань редагування зображень

Нижче представлено на рисунку 3.6 глибокий автоенкодер, у якого 3 приховані шари та рисунок 3.7, на якому теж представлено глибокий автоенкодер, але у нього уже 5 прихованих шарів.

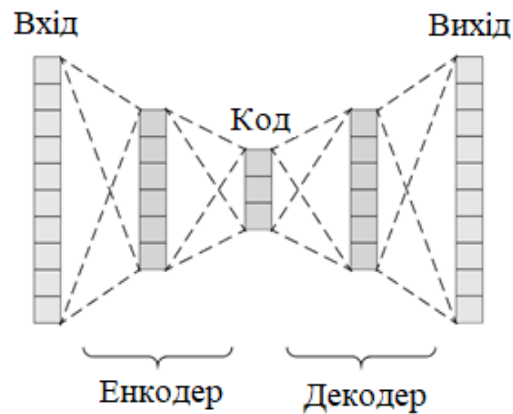


Рисунок 3.6 – Глибокий автоенкодер (3 прихованих шарів)

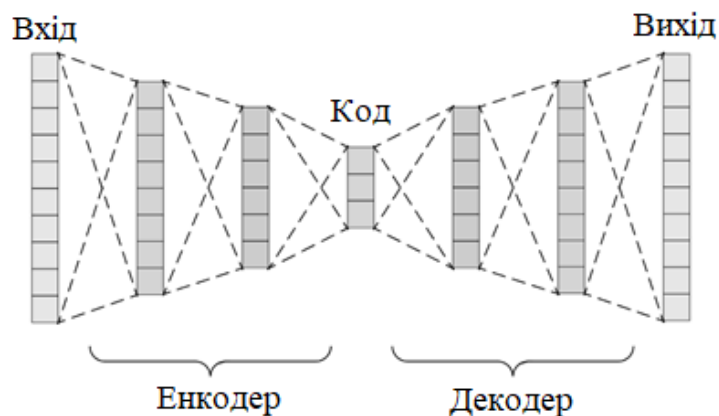


Рисунок 3.7 – Глибокий автоенкодер (5 прихованих шарів)

Swapping Autoencoder складається з операції автоматичного кодування Зверху та заміни знизу. Вгорі кодер E вбудовує вхід (Notre-Dame) у два коди. Структурний код є тензором з просторовими розмірами; код текстури є 2048-вимірним вектором. Декодування за допомогою генератора G має створювати реалістичне зображення (закріплене дискримінатором D), що відповідає вхідному сигналу (втрати при реконструкції). За ідеальної конвергенції навчання Swapping Autoencoder заохочує кілька бажаних властивостей вивченого простору

вбудовування. Функція кодування E оптимізована в бік ін'єкції через втрати реконструкції, оскільки різні зображення відображаються на різні приховані коди. Крім того, вибір дизайну сприяє тому, щоб різні коди створювали різні виходи через G : код текстури повинен фіксувати розподіл текстури, тоді як код структури повинен фіксувати інформацію про місце розташування вхідних зображень. Нарешті, спільний розподіл двох кодів зображень, згенерованих обміном, враховується конструкцією, оскільки коди структури поєднуються з кодами випадкової текстури. На рисунку 3.8 представлено схему роботи Swapping Autoencoder.

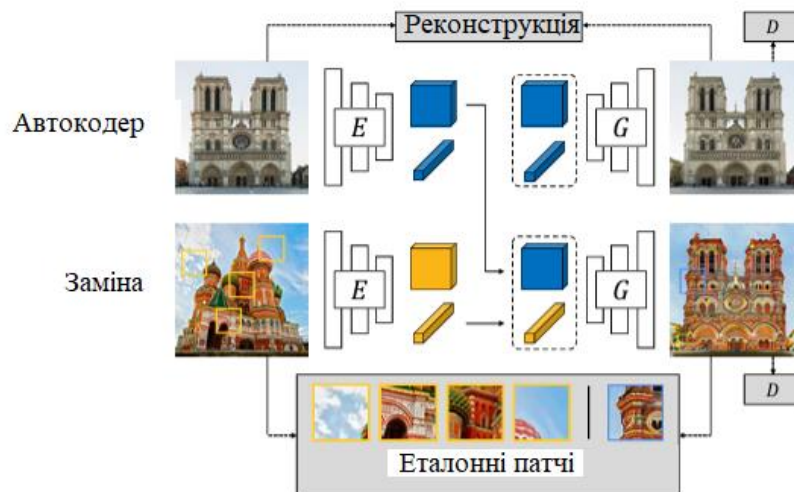


Рисунок 3.8 – Схема роботи Swapping Autoencoder

На рисунку 3.9 представлені приклади вбудовування та якість конструкції.

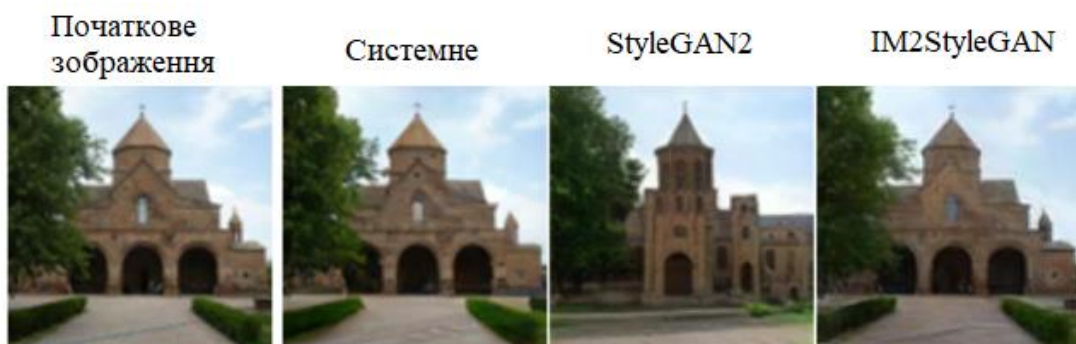


Рисунок 3.9 – Приклади вбудовування та якість конструкції

Проектування зображення у простори для вбудовування методу та базових моделей GAN, Im2StyleGA та StyleGAN2. Ці еконструкції краще зберігають деталізований контур, ніж StyleGAN2 і виглядають чіткішими, ніж Im2StyleGAN. Це перевірено в середньому за допомогою показника LPIPS. Ці методи також створюють зображення набагато швидше, ніж останні генеративні моделі, які використовують повторювану оптимізацію.

Першим кроком маніпулювання зображенням за допомогою генеративної моделі є проектування його в його латентну лопату. Якщо вхідне зображення не можна проектувати з високою точністю, вбудований вектор не можна використовувати для редагування, оскільки користувач редагував би інше зображення. Рисунок 3.10 ілюструє як приклад реконструкції, так і кількісне вимірювання якості реконструкції.



Рисунок 3.10 – Заміна зображень

Кожен рядок показує результат поєднання структурного коду крайнього лівого зображення з кодом текстури верхнього зображення (навчання на LSUN Church and Bedroom). Ця модель створює реалістичні зображення, які зберігають текстуру (наприклад, матеріал будівлі або малюнок простирадла) і структуру (контури предметів).

Метод туру точно зберігає візерунок дверей (верхній) і риси обличчя (нижній) без розмитості забезпечує кращу якість реконструкції, ніж базові лінії. Виняток становить набір даних Church, де Im2StyleGA отримує кращу оцінку реконструкції. Важливо, оскільки цей метод розроблено з урахуванням часу вбудовування, він вимагає лише одного проходу вперед, принаймні в 1000 разів швидше, ніж базові лінії, які вимагають сотні тисяч кроків оптимізації. Далі ми досліджуємо, наскільки корисним є вбудовування, досліджуючи маніпуляції з отриманим кодом. На рисунку 3.8 показано приклади гібридних зображень за нашим методом, отриманих шляхом поєднання структури та коди текстур з різних зображень. Текстури верхнього ряду зображень є незмінними переданий: візерунки неба, фасаду та вікон відображаються у відповідних областях конструкції образи на церквах, і так само на простирадлах.

Реалістичність гібридів зображень. Для гібридів зображень додатково було порівняно з SOTA методи передачі стилю, хоча вони безпосередньо не застосовуються для контрольованого редагування вбудовування зображень. Справжнє та згенероване зображення послідовно відображаються Amazon протягом однієї секунди кожне Механічні туркери (АМТ), які вибирають, що вони вважають підробкою. Алгоритм, який генерує абсолютно правдоподібні зображення, досягне 50%-го рівня обману. Було зібрано 15 000 суджень, по 1000 для кожного алгоритму та набору даних. Цей метод досягає більш реалістичні результати для всіх наборів даних. Найближчим конкурентом є метод WCT2, який є призначений для фотореалістичної передачі стилю. Усереднене для трьох наборів даних цей метод досягає найвищий відсоток обману (31:0 1:4%), з WCT2, що слідує за статистичною значущістю (28:6 1:3%). Якісні приклади ілюстровано на рисунку 3.9. На рисунку 3.10 візуалізовано результат 3750 суджень користувачів щодо чотирьох базових ліній і трьох наборів даних, які показують, що цей метод перевершує всі базові методи зі статистичною значущістю щодо збереження стилю. Що стосується збереження вмісту, цей метод лише відстає від WCT2, який є фотореалістичним методом стилізації, який лише незначно змінює колір вхідних

					КВРКІ 200244.20.02.22 ПЗ	Арк. 64
Зм.	Арк.	№ докум.	Підпис	Дата		

даних. Найважливіше те, що цей метод досягає найкращої продуктивності зі статистичною значущістю як стилю, так і вмісту серед моделей, які можуть вставляти зображення, що потрібно для інших форм редагування зображень. На рисунку 3.11 представлені порівняння іміджевих гібридів.



Рисунок 3.11 – Порівняння іміджевих гібридів

На рисунку 3.12 представлена безперервна інтерполяція.

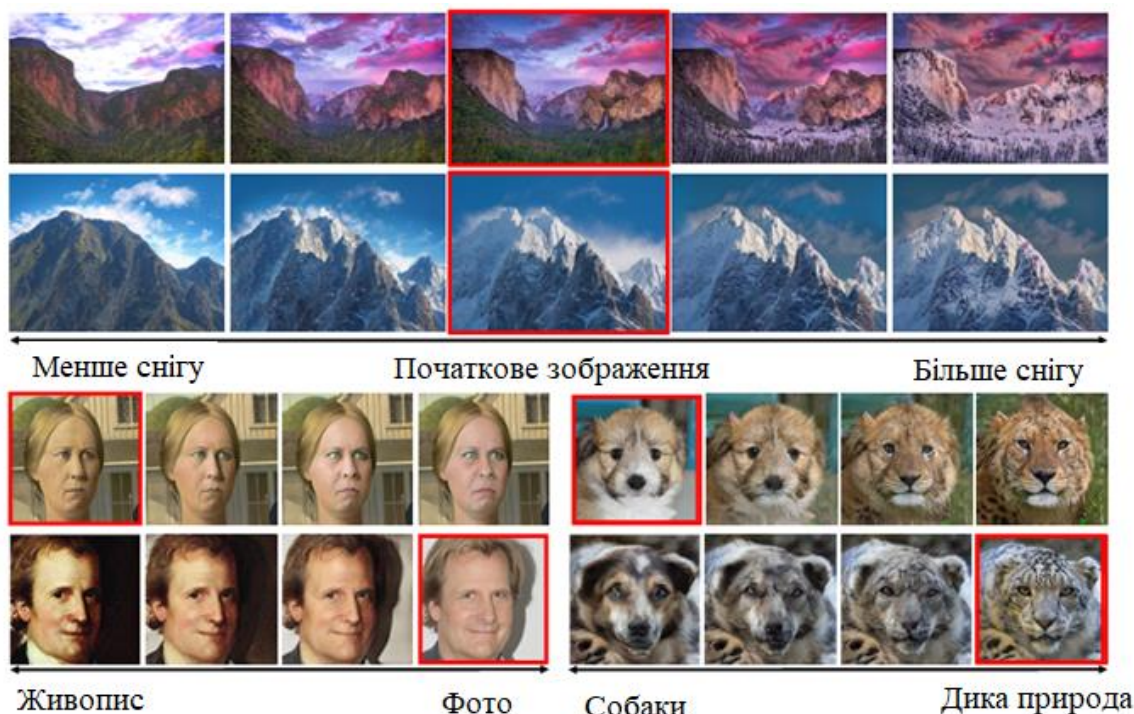


Рисунок 3.12 – Безперервна інтерполяція

Вгорі вектор маніпуляцій для снігу було виявлено, коли було зроблено різницю між 10 зібраними користувачами фотографіями засніжених і літніх гір. Вектор просто додається до коду текстури вхідного зображення (червоного) з деяким посиленням. Внизу багатодоменне безперервне перетворення досягається шляхом застосування середньої векторної різниці між кодами текстур двох доменів на основі анотацій із навчальних наборів. Навчання було на наборах даних Portrait2FFHQ і AFHQ.

3.3. Автокодер із збереженням конфіденційності для спільного виявлення об'єктів

Результати останніх програм штучного інтелекту, таких як автономне водіння, візуальне спостереження та Інтернет речей, обсяг даних, що передається між «розумними» периферійними пристроями (наприклад, камерами) і хмарними сервісами, становить швидко зростає. У цих програмах висновок виконується спільно між периферійним пристроєм, який часто має обмежені обчислювальні можливості та більш потужним обчислювальним об'єктом, який зазвичай називають хмарою. Враховуючи швидке зростання кількості таких міжмашинних (M2M) з'єднань, існує нагальна потреба в ефективній методології стиснення даних, адаптованій до автоматизованого машинного аналізу. JPEG AI і MPEG-VCM — це дві відомі групи стандартизації, які активно працюють у цій галузі, щоб задовольнити нагальну потребу у стисненні даних для машин. У рамках цілей дослідження у цій роботі було прогнано покращити ефективність стиснення для таких спільних систем аналізу M2M. У той же час конфіденційність завжди була головною проблемою в передачі даних. Це занепокоєння стає все більшим, який виконує атаку інверсії моделі, критично важливу, оскільки обсяг даних, які збираються та передаються між периферійними пристроями та хмарою, продовжує зростати. На рисунку 3.13 представлена модель виявлення об'єктів, розгорнута в рамках спільної розвідки з супротивником, який виконує атаку інверсії моделі.

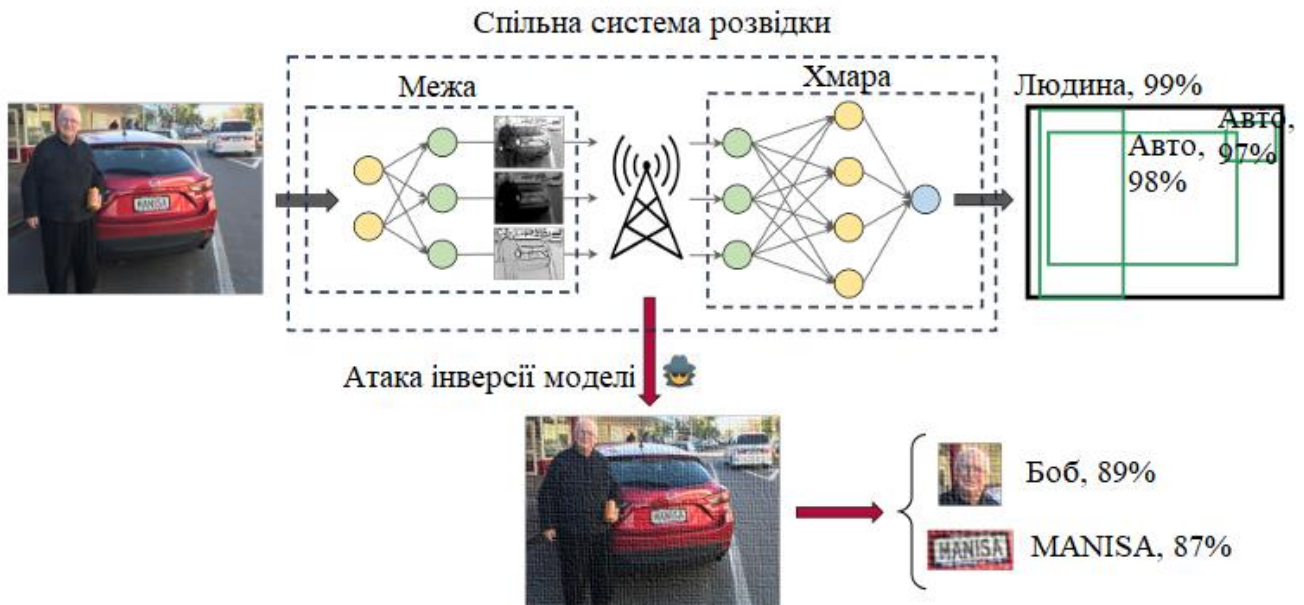


Рисунок 3.13 – Модель виявлення об’єктів, розгорнута в рамках спільної розвідки з супротивником, який виконує атаку інверсії моделі

Було розглянуто сценарій, коли зловмисник отримує несанкціонований доступ до даних, зібраних камерами дорожнього руху та іншими датчиками, які використовуються для моніторингу дорожнього руху. Якщо вони зможуть ідентифікувати людей за такими даними, вони зможуть відстежувати їх, визначати їхній спосіб життя, а потім використовувати ці знання для різних незаконних і шкідливих цілей. Подібні проблеми існують у програмах для розумного дому, незалежному житті тощо. Схеми криптографії, такі як шифрування даних або секретний обмін пропонують одне з можливих рішень для захисту даних, хоча вони пов’язані з власними ризиками та проблемами, як потреба в довіреній стороні. Спільний інтелект має потенціал для пом’якшення ризику розкриття приватної інформації, оскільки дані, що передаються між периферійним пристроєм і хмарою, є не вихідними вхідними даними, а деякими проміжними характеристиками моделі DNN проведення запланованого аналізу. У цій структурі перші кілька шарів моделі DNN виконуються на межі, а результуючі функції кодуються та передаються в хмару, де розгортаються інші шари. Розподіл обчислень DNN між периферією та хмарою може забезпечити затримку та енергоефективність. Хоча цей підхід

розкриває лише приховане представлення вхідних даних, він залишається вразливим до атаки інверсії моделі, за допомогою якої вхідні дані можна відновити певною мірою. Щоб протистояти цій атаці, одна зі стратегій полягала б у видаленні інформації, яка не має відношення до завдання, а також, при цьому, приватної інформації, зберігаючи в переданих даних лише інформацію, що стосується завдання. Однак розрізнення між релевантною для завдання, нерелевантною для завдання та приватною інформацією залишається проблемою, особливо у візуальних даних реального світу.

Виявлення об'єктів за допомогою YOLOv5, як завдання машинного зору, хоча загальна структура спільного інтелекту також застосовна до інших моделей і завдань DNN. Існують різні версії YOLOv5 з різними рівнями складності та точності було обрано модель YOLOv5m як основу для експериментів. У запропонованій структурі було дотримано двох ключових цілей: стійкість до MIA та підвищення ефективності стиснення. Для досягнення цих цілей навчався автокодер, вставлений у модель машинного зору, разом із допоміжною мережею (ResNet), спрямованою на реконструкцію вхідних даних у змагальній манері. Варто зазначити, що автокодер навчається в автономному режимі перед розгортанням. На етапі висновку інтерфейс YOLO та «AE» розгортаються на периферійному пристрої разом із кодувальником VVC, тоді як декодер VVC, «AD» і серверна частина YOLO розгортаються в хмарі. «ResNet» використовується лише під час навчання для покращення стійкості автокодувальника до змагань і не є частиною розгорнутого конвеєра. Ці компоненти докладніше описано нижче.

Для спільного розгортання периферійної хмари модель YOLOv5m має бути розділена на дві частини: фронтенд, який буде розгорнуто на периферійному пристрої і серверну частину, яка буде знаходитись у хмарі. Глибша точка розділення є більш бажаною з точки зору як ефективності стиснення, так і конфіденційності. Однак обмежена обчислювальна потужність і енергоспоживання крайніх пристроїв сприяють вибору меншої точки розділення. Вибір точки розділення глибше цього рівня вимагатиме кодування та передачі кількох наборів

тензорів ознак, що збільшить складність і зробить стиснення більш складним. Таким чином, було обрано найглибшу розумну точку розколу для моделі YOLOv5m.

У точці розділення автокодер інтегрується та навчається з метою зменшення розмірності вихідного латентного простору та створення вузьких місць із покращеною стисливістю та стійкістю до МІА. Це стратегія plug-and-play, і її можна легко застосувати до інших моделей і завдань комп'ютерного зору без повторного навчання оригінальної моделі зору. Роль АЕ полягає в тому, щоб перетворити власний латентний простір YOLO в точці розщеплення (зі 192 каналами) у вузькі місця меншого розміру (тут, з 64 каналами). Просторові розміри залишаються незмінними для підтримки просторової точності, що є вирішальним для подальшого виявлення об'єктів. Отриманий тензор ознак вузького місця розміщується на мозаїку та попередньо квантується до 8 бітів на елемент, а потім кодується за допомогою VVC-Intra. На стороні хмари закодований бітовий потік декодується декодером VVC і AD, а потім передається на серверну частину YOLOv5m для виявлення об'єктів.

Навчання проводиться в автономному режимі. Було прийнято спеціальну стратегію навчання для створення вузьких місць, що стискаються, які є стійкими до МІА, але можуть підтримувати завдання виявлення об'єктів. Це робиться за допомогою допоміжної моделі DNN, яка називається Reconstruction Network (RecNet), метою якої є імітація супротивника, який виконує МІА, і спроба реконструювати вхідне зображення з ознак вузького місця. Архітектура RecNet приблизно відображає архітектуру YOLOv5m інтерфейсу та АЕ. Це відповідає атаці білого ящика, коли супротивник знає архітектуру атакованої моделі. Автокодер і RecNet навчаються разом. Протягом усього процесу навчання RecNet намагається якнайкраще відновити вхідне зображення з вузьких місць, тоді як АЕ намагається порушити продуктивність RecNet, маніпулюючи згенерованими вузькими місцями. Водночас як АЕ, так і AD працюють для підтримки високої точності виявлення об'єктів. Крім того, функції вузьких місць поступово

оптимізуються для ефективності стиснення за допомогою функції втрати стисливості під час навчання. На рисунку 3.14 представлена якість зображення, яка може бути недостатнім показником конфіденційності.



Рисунок 3.14 – Якість зображення може бути недостатнім показником конфіденційності

На рисунку 3.14 показано оригінальне зображення, яке розкриває деяку приватну інформацію (обличчя та номерний знак), на зображенні у низькій якості шум додається до всього зображення, за винятком області обличчя та номерного знака, створюючи низький PSNR, але приватна інформація все ще повністю відкрита, а на зображенні вищої якості шум додається лише до області обличчя та номерного знака, що призводить до зображення зі значно вищим значенням PSNR, але з приватною інформацією, яка тепер закрита шумом. Тому в цьому прикладі PSNR є поганим проксі для захисту конфіденційності. З цієї причини ми пропонуємо систему оцінювання, у якій конфіденційність вимірюється більш безпосередньо, замість того, щоб покладатися на проксі-сервери, такі як PSNR і SSIM. Зокрема, наша оцінка ґрунтується на точності розпізнавання обличчя та номерного знака, отриманій на зображеннях, відновлених за допомогою MIA на вузьких місцях.

Обличчя містять важливу інформацію, за допомогою якої можна розкрити особу людини. Щоб кількісно визначити рівень конфіденційності запропонованого методу, було виміряно, наскільки точно модель розпізнавання обличчя може ідентифікувати обличчя на зображеннях, відновлених MIA з вузьких місць.

Доступні різні моделі розпізнавання обличчя, особливо на основі DNN. MagFace — добре відома модель розпізнавання обличчя на основі DNN із високою точністю. MagFace досяг 99,83% точності перевірки набору даних LFW. Моделі розпізнавання обличчя часто оцінюють під час завдання верифікації, мета якого полягає в тому, щоб визначити, чи два різних зображення показують ту саму особу. Однак конфіденційність насамперед стосується точності, з якою людину можна розпізнати на зображенні. Модель розпізнавання обличчя має відповідати зображенням у наборі зондів із відповідними зображеннями в галереї, на яких показано тих самих людей. Набір даних LFW складається з 13 233 зображень, що належать 5 749 особам, 1 680 з яких мають принаймні два доступних зображення. З цих 1 680 осіб одна з їхніх фотографій призначається до набору галереї, а решта призначаються до набору зондів. Далі будуть збиратися вихідні вектори ознак, згенеровані MagFace для всіх зображень, і буде вчислятись відстань між усіма парами елементів із галереї та наборів зондів. Зображення в галереї з найменшою віддаленістю від досліджуваного зображення зонда вибирається як збіг. Зрештою, точність ідентифікації розраховується за кількістю правильно виявлених збігів.

Дані номерних знаків — це ще одна форма конфіденційної інформації, яку можна використати для відстеження конкретних транспортних засобів із потенційно зловмисною метою за допомогою міських камер моніторингу руху. Точність розпізнавання номерних знаків можна вважати ще одним прямим показником конфіденційності. Для цієї мети було використано LPRNet, модель розпізнавання номерних знаків на основі DNN. Загальний підхід до звітності про точність методів розпізнавання номерних знаків полягає в тому, щоб вважати результат моделі правильним лише коли всі символи номерного знака розпізнаються правильно. Однак, з точки зору конфіденційності, навіть частково успішне розпізнавання може вважатись витоком конфіденційності. Тому було взято за критерій точність розпізнавання символів. У цих експериментах ця точність розраховується на наборі з 5 000 зображень із набору даних CCPD. На рисунку 3.15 представлені приклади зображень, відновлених за допомогою MIA.

					КВРКІ 200244.20.02.22 ПЗ	Арк. 71
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВОК

У результаті дослідження було виявлено, що аналіз нейромережі автоенкодер є важливим етапом у дослідженні областей застосування штучного інтелекту. Дослідження також підтвердило потенціал використання автоенкодеру для забезпечення безпеки при розпізнаванні об'єктів на фотографіях. Порівняльний аналіз алгоритмів методів роботи автоенкодеру підкреслив його ефективність та потенційні області застосування. Отже, дане дослідження сприяє розробці ефективних систем розпізнавання зображень на основі нейромереж автоенкодерів, що може мати значний вплив на розвиток штучного інтелекту у різних сферах життя. Об'єктом дослідження є процес розпізнавання зображень заснованого на автоенкодері, а предметом дослідження є методи розпізнавання зображень, які засновані на базі автоенкодеру.

					КВРКІ 200244.20.02.22 ПЗ	Арк. 73
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Архітектура штучних нейронних мереж. URL:<https://library.ztu.edu.ua/e-copies/books/kononuk1.pdf>(дата звернення: 09.02.2024).
2. Системи неконтрольованого розпізнавання зображень. URL:<https://scholarscompass.vcu.edu/etd/7482/>(дата звернення: 09.02.2024).
3. Метод перевірки зображення глибокого навчання без нагляду. URL:<https://arxiv.org/abs/2312.14395>(дата звернення: 10.02.2024).
4. Кластеризація зображень за допомогою обмеженої машини Больцмана. URL:<https://arxiv.org/abs/2312.13845>(дата звернення: 10.02.2024).
5. Метод розпізнавання обличчя на основі глибокого навчання з використанням сіамської мережі. URL: <https://arxiv.org/abs/2312.14001>(дата звернення: 10.02.2024).
6. Використання автоенкодерів для класифікації. URL:https://www.researchgate.net/publication/339945889_Autoencoders(дата звернення: 15.02.2024).
7. Аналіз систем розпізнавання зображень на прикладі обличчя. URL:<https://arxiv.org/pdf/2312.14301>(дата звернення: 15.02.2024).
8. Глибоке навчання на основі автокодерів для усунення шумів. URL:https://www.sciencedirect.com/science/article/pii/S1877050920311431?ref=pdf_download&fr=RR-2&rr=894d22973c23bfdc(дата звернення: 19.02.2024).
9. TensorFlow. URL:<https://www.tensorflow.org/>(дата звернення: 19.02.2024).
10. Згорткові нейронні мережі. URL:<https://itmaster.biz.ua/programming/vision/cnns1.html>(дата звернення: 19.02.2024).
11. Двовимірне графічне середовище. URL:<https://ieeexplore.ieee.org/document/4160265>(дата звернення: 22.02.2024).
12. Прогнозна технологічна модель. URL:<https://ptm.asu.edu/>(дата звернення: 22.02.2024).
13. Реалізація UNet для сегментації зображень. URL:<https://arxiv.org/html/2403.02863v1>(дата звернення: 23.02.2024).

					КВРКІ 200244.20.02.22 ПЗ	Арк. 74
Зм.	Арк.	№ докум.	Підпис	Дата		

14. Автоматизована система розпізнавання об'єктів на основі згорткового автокодера. URL:

https://www.researchgate.net/publication/344371110_Automated_Object_Recognition_System_based_on_Convolutional_Autoencoder(дата звернення: 23.02.2024).

15. Структура обробки зображень на основі автокодувальника для модифікації зовнішнього вигляду об'єкта. URL:

<https://link.springer.com/article/10.1007/s00521-020-04976-7>(дата звернення: 26.02.2024).

16. Опорно-векторні мережі. URL:

<https://link.springer.com/article/10.1007/BF00994018>

17. DNN. URL: <https://arxiv.org/pdf/1701.05549>(дата звернення: 27.02.2024).

18. Усунення шумів зображення за допомогою згорткового автокодувальника. URL: <https://arxiv.org/pdf/2207.11771>(дата звернення: 02.03.2024).

19. Вступ до згорткових нейронних мереж. URL:

<https://arxiv.org/abs/1511.08458>(дата звернення: 03.02.2024).

20. ASU-CNN: ефективна глибока архітектура для класифікації зображень і візуалізації функцій. URL: <https://arxiv.org/pdf/2305.19146>(дата звернення: 03.03.2024).

21. CNN архітектура. URL: <https://arxiv.org/pdf/2401.13575>(дата звернення: 03.03.2024).

22. YOLOv5m. URL:https://www.researchgate.net/figure/Performance-of-YOLOV5s-YOLOV5m-YOLOV5l-and-YOLOV5x_tbl2_362855802(дата звернення: 05.03.2024).

23. Зручна диференціальна бібліотека конфіденційності. URL: <https://arxiv.org/abs/2109.12298>(дата звернення: 05.03.2024).

24. Архітектура, що поєднує згорткову нейронну мережу (CNN) і машина опорних вікторів (SVM) для класифікації зображень. URL: <https://arxiv.org/pdf/1712.03541>(дата звернення: 08.03.2024).

					КВРКІ 200244.20.02.22 ПЗ	Арк. 75
Зм.	Арк.	№ докум.	Підпис	Дата		

25. Навчений компресор зображень на основі автокодувальника. URL: <https://arxiv.org/pdf/1902.07385>(дата звернення: 10.03.2024).
26. Штучний інтелект. URL: https://jai.in.ua/index.php/en/issues?paper_num=1504(дата звернення: 12.03.2024).
27. Вивчення впливу архітектур автокодерів, що знешумлюють, на пошук зображень. URL: <https://www.sciencedirect.com/science/article/pii/S1877050924009177>(дата звернення: 12.03.2024).
28. Ентропійне кодування. URL: https://speechprocessingbook.aalto.fi/Transmission/Entropy_coding.html
29. Сегментація зображень. URL: <https://www.ibm.com/topics/image-segmentation>(дата звернення: 12.03.2024).
30. Декілька типів архітектури штучних нейронних мереж, які вам слід знати. URL: <https://dedekurniawann.medium.com/several-types-of-artificial-neural-networks-architecture-that-you-should-know-c169a5e22ec7>(дата звернення: 14.03.2024).
31. Ефективність згорткових автокодерів на персоналізованих системах рекомендацій на основі зображень. URL: <https://arxiv.org/pdf/2003.06205>(дата звернення: 14.03.2024).
32. Навчання обернених нейромереж як автокодерів. URL: <https://arxiv.org/pdf/2303.11239>(дата звернення: 14.03.2024).
33. Автоенкодер. URL: <https://en.wikipedia.org/wiki/Autoencoder>(дата звернення: 14.03.2024).
34. Автокодувальники та функція усунення шумів: від теорії до практики. URL: <https://towardsdatascience.com/autoencoders-and-the-denoising-feature-from-theory-to-practice-db7f7ad8fc78>(дата звернення: 15.03.2024).
35. Розблокування високоточної диференціально приватної класифікації зображень. URL: <https://arxiv.org/abs/2204.13650>(дата звернення: 09.02.2024).
36. Максисетний підхід моделі гаусового шуму. URL: <https://hal.science/hal-00004331/>(дата звернення: 15.03.2024).

					КВРКІ 200244.20.02.22 ПЗ	Арк. 76
Зм.	Арк.	№ докум.	Підпис	Дата		

37. База даних mnist рукописних цифр. URL: <http://yann.lecun.com/exdb/mnist/>
38. Swapping Autoencoder для глибокої обробки зображень. URL: <https://arxiv.org/pdf/2007.00653>(дата звернення: 17.03.2024).
39. Автокодер із збереженням конфіденційності для спільного виявлення об'єктів. URL: <https://arxiv.org/pdf/2402.18864>(дата звернення: 17.03.2024).
40. Штучна нейронна мережа. URL:<https://library.ztu.edu.ua/e-copies/books/kononuk1.pdf>
41. Не глибокий автоенкодер. URL: <https://dsplab.feri.um.si/ocenjevanje-zaupanja-v-globokih-nevronskih-mrezah/>
42. Використання автоенкодерів для класифікації. URL: https://www.researchgate.net/figure/An-illustration-for-using-autoencoders-as-regularization-for-supervised-models-Given-the_fig5_339945889
43. Аналіз систем розпізнавання зображень на прикладі обличчя. URL: <https://arxiv.org/pdf/2312.14301>
44. Реалізація UNet для сегментації зображень. URL: <https://arxiv.org/html/2403.02863v1>
45. Усунення шумів зображення за допомогою згорткового автокодувальника. URL: <https://arxiv.org/pdf/2207.11771>
46. ASU-CNN: ефективна глибока архітектура для класифікації зображень і візуалізації функцій. URL: <https://arxiv.org/pdf/2305.19146>
47. Архітектура, що поєднує згорткову нейронну мережу (CNN) і машина опорних вікторів (SVM) для класифікації зображень. URL: <https://arxiv.org/pdf/1712.03541>
48. Навчений компресор зображень на основі автокодувальника. URL: <https://arxiv.org/pdf/1902.07385>
49. Swapping Autoencoder для глибокої обробки зображень. URL: <https://arxiv.org/pdf/2007.00653>
50. Глибокий автоенкодер з трьома прихованими шарами. URL: <https://www.mdpi.com/2075-163X/11/10/1089>

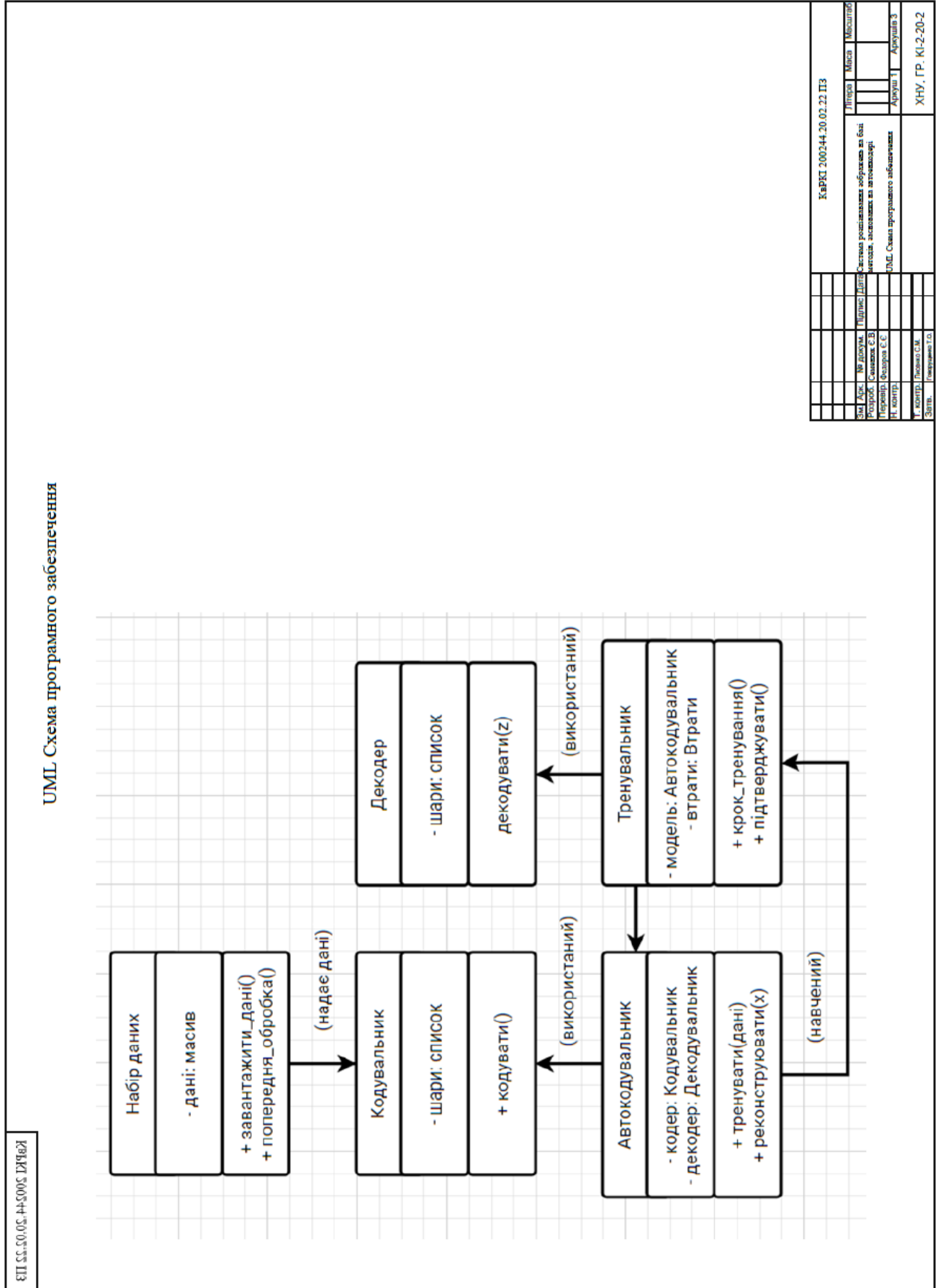
51. Глибокий автоенкодер з п'ятьма прихованими шарами. URL: <https://towardsdatascience.com/word-vectors-and-decoding-autoencoder-for-dimensionality-reduction-407815ead4b7>

52. Автокодер із збереженням конфіденційності для спільного виявлення об'єктів URL: <https://arxiv.org/pdf/2402.18864>

					КВРКІ 200244.20.02.22 ПЗ	Арк. 78
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток А (обов'язковий)

Копія креслення «UML Схема програмного забезпечення»

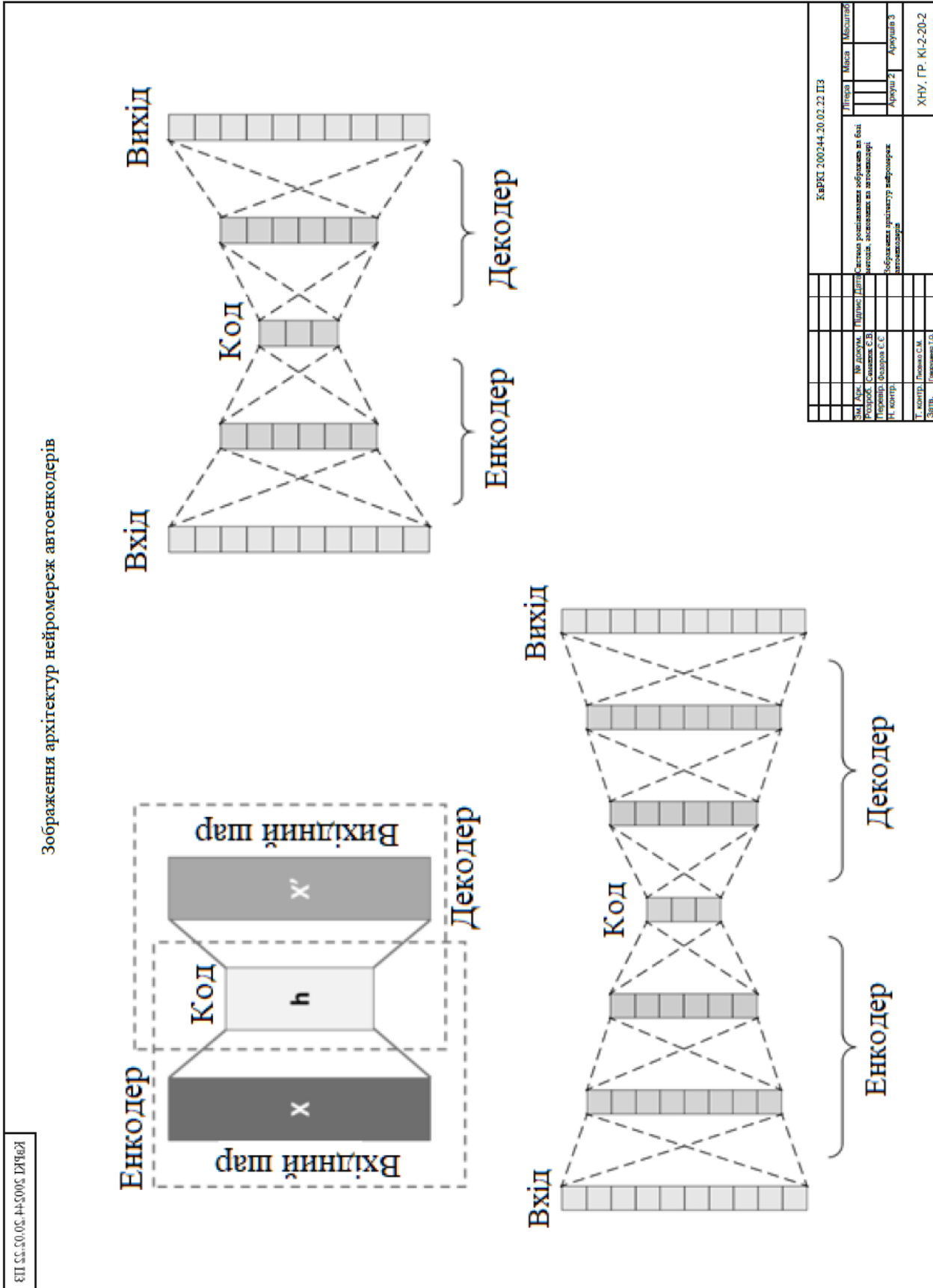


Знак	Док.	М. добуток	Підпис	Датум	Літера	Маса	Масштаб
Розроб.	Самозах. Є.В.						
Н. контр.	Федоренко Є.Є.				Архив 1	Архив 2	Архив 3
І. контр.	Голованюк С.М.						
Затв.	Голованюк С.М.						
	Голованюк С.М.						

ХНУ. ГР. КІ-2-20-2

Додаток Б
(обов'язковий)

Копія креслення «Зображення архітектур нейромереж автоенкодерів»



Додаток В (обов'язковий)

Копія креслення «Двійкове арифметичне кодування карт ознак»

ЕПІ 55.50.05.4450005 ДІВІВІ	<p style="text-align: center;">Двійкове арифметичне кодування карт ознак</p> <ol style="list-style-type: none"> 1: Initialize final feature maps Z 2: for each feature maps do 3: Initialize Block Flag $B_{m,n}$ to 0 4: Initialize Sample Flag $S_{m,n,i,j}$ to 0 5: Initialize All Zero Bitplane Flag AZB to 1 6: for each bitplane $k = \text{MSB}$ to LSB do 7: if $AZB == 1$ then 8: Set AZB as per all zero bitplane state 9: Encode AZB 10: if $AZB == 0$ then 11: for each block m, n do 12: if $B_{m,n} == 0$ then 13: Set $B_{m,n}$ as per significant block state 14: Encode $B_{m,n}$ 15: if $B_{m,n} == 1$ then 16: for each sample i, j do 17: if $S_{m,n,i,j} == 0$ then 18: Set $S_{m,n,i,j}$ as per significance state 19: Encode <i>significant bit</i> 20: else 21: Encode <i>refinement bit</i>
-----------------------------	---

КАРКИ 200344.30.02.22.ІІЗ			
ЗМ. Лист	№ докум.	Держав.	Місця
Розроб.	Складено	Затверд.	Використ.
Проек.	Використ.	Датум	Архив
Г. Київ	Г. Київ	Г. Київ	Г. Київ
Р. Сімонов	Р. Сімонов	Р. Сімонов	Р. Сімонов
СІПБ	СІПБ	СІПБ	СІПБ
ХНУ, ГР. КІ-2-20-2			

Ім'я користувача:
Кафедра КІ

ID перевірки:
1016375325

Дата перевірки:
19.06.2024 14:07:41 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
19.06.2024 14:09:51 EEST

ID користувача:
100005591

Назва документа: Семенюк_Система розпізнавання зображень на базі методів, заснованих на автоенкодері
Кількість сторінок: 82 Кількість слів: 13189 Кількість символів: 102792 Розмір файлу: 6.13 MB ID файлу: 1016183217

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

8.04% Схожість

Найбільша схожість: 6.32% з Інтернет-джерелом (<http://library.ztu.edu.ua/e-copies/books/kononuk1.pdf>)

7.85% Джерела з Інтернету 70 Сторінка 84

0.81% Джерела з Бібліотеки 137 Сторінка 84

0% Цитат

Не знайдено жодних цитат

Посилання 1 Сторінка 84

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 2

Підозріле форматування 16 сторінок

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 4.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 13%

ID: 131544 Назва: БКР Система розпізнавання зображень на базі методів, заснованих на автоенкодері Додано в БД: 2024-06-19 Автор: С. В. Семешко Керівник: С. С. Федоров Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	93681	763	4210 (4%)	43 (6%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Семенюк Євгеній Валерійович

Тема: Система розпізнавання зображень на базі методів, заснованих на автоенкодері.

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 70

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є дослідження та аналіз системи розпізнавання зображень на базі методів, заснованих на автоенкодері.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи:

У першому розділі розглянуто поняття ШНМ та її архітектуру. Розглянуто переваги та недоліки використання автоенкодера у системах розпізнавання зображень та використання автоенкодерів для класифікації. Зроблено аналіз систем розпізнавання зображень на прикладі облич та методів напівконтрольованого та неконтрольованого навчання в контексті глибоких нейронних мереж. У другому розділі було досліджено архітектуру та реалізацію UNet для сегментації зображень. Розглянуто поняття шум та як відбувається усунення шумів зображення за допомогою згорткового автокодувальника. Було розглянуто операцію згортки та застосування операції декоінволюції. Проаналізовано використання згорткового автоенкодера та ReLU. Також було досліджено та проаналізовано архітектуру, яка поєднує CNN та SVM для класифікації зображень, та ASU-CNN, як ефективну глибоку архітектуру для класифікації зображень та візуалізації функцій. У третьому розділі в результаті дослідження було виявлено, що аналіз нейромережі автоенкодер є важливим етапом у дослідженні областей застосування штучного інтелекту.

Дослідження також підтвердило потенціал використання автоенкодеру для розпізнавання об'єктів на фотографіях. Порівняльний аналіз алгоритмів методів роботи автоенкодеру підкреслив його ефективність та потенційні області застосування. Отже, дане дослідження сприяє розробці ефективних систем розпізнавання зображень на основі нейромереж автоенкодерів, що може мати значний вплив на розвиток штучного інтелекту у різних сферах життя. Об'єктом дослідження є процес розпізнавання зображень заснованого на автоенкодері, а предметом дослідження є методи розпізнавання зображень, які засновані на базі автоенкодеру.

4. Позитивні сторони роботи: система може детектувати об'єкти на зображеннях, а також вона дозволяє автоматизувати цей процес з високою ймовірністю класифікації зображень.

5. Негативні сторони роботи: Не проведено дослідження для визначення кількості шарів автоенкодеру.

Не проведено дослідження для визначення залежності ймовірності класифікації від рівня шуму.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження:

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Кирило Дмитро Павлович, завідувач кафедрою інформатики,
К.Т.К. 9252117

"20" 06 2024 р.

 (підпис)

Завідувачу кафедри КІС
д-р.техн.наук, проф. Говорушенко Т. О.

Семенюка Євгенія Валерійовича

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-20-2

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

22 квітня 2024 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система розпізнавання зображень на базі методів, заснованих на автоенкодері

Автор: Семенюк Євгеній Валерійович

Спеціальність: 123- Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Федоров Євген Євгенович, д.т.н. професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 137 джерелами на один фрагмент речення;
- 4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 8.04% і адресується до 70 першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



Є. Є. Федоров

С. М. Лисенко

Т. О. Говорушенко