

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження використання автокодувальників для
знешумлення зображень
(тема)

Виконав:
студент 2 курсу, групи СШМ-22-2
Марченко М.Є.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

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

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник ст.викл Попов С.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

В.О. Філатов
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)
Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
(код і повна назва)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Марченку Максиму Євгеновичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження використання автокодувальників для знешумлення зображень

затверджена наказом університету від 1 квітня 20 24 р. № 260Ст

2. Термін подання студентом роботи до екзаменаційної комісії 12 червня 20 24 р.

3. Вихідні дані до роботи науково-технічні публікації, дані Інтернет-джерел, тематична література, документація до профільних бібліотек

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі та постановка задачі

2) Теоретичні дослідження

3) Використання автокодувальників для знешумлення зображень

РЕФЕРАТ

Пояснювальна записка: 54 с., 19 рис., 2 дод., 20 джерел.

АВТОКОДУВАЛЬНИКИ, ГЛИБИННЕ НАВЧАННЯ,
ЗНЕСУМЛЕННЯ ЗОБРАЖЕНЬ, КОМП'ЮТЕРНИЙ ЗІР, МАШИННЕ
НАВЧАННЯ, ШТУЧНІ НЕЙРОННІ МЕРЕЖІ.

Об'єкт дослідження – штучні нейронні мережі-автокодувальники.

Предмет дослідження – застосування автокодувальників для вирішення задачі знесумлення зображень.

Мета роботи – дослідження використання автокодувальників для знесумлення зображень.

Методи дослідження – опрацювання літературних та електронних джерел, розгляд будови та використання автокодувальників.

Досліджено використання для вирішення задачі знесумлення зображень штучних нейронних мереж-автокодувальників, розглянуто їх будову, особливості, процес навчання та використання, сильні та слабкі сторони. В ході проведення експериментальних досліджень було підтверджено спроможність автокодувальників до вирішення задачі знесумлення зображень. Були продемонстровані переваги та недоліки використання автокодувальників для вирішення зазначеної задачі. Було проведено порівняння різних моделей, орієнтованих на вирішення зазначеної задачі.

ABSTRACT

Master's thesis contains: 54 pp., 19 fig., 2 ann., 20 references.

**ARTIFICIAL NEURAL NETWORKS, AUTOENCODERS,
COMPUTER VISION, DEEP LEARNING, IMAGE DENOISING, MACHINE
LEARNING.**

The object of research is autoencoders – a kind of artificial neural networks.

The subject of the study is the use of autoencoders to solve the problem of images denoising.

The purpose of the work is to study the use of autoencoders for images denoising.

Research methods – study of literary and electronic sources, consideration of the structure and use of autoencoders.

The use of artificial neural networks-autoencoders to solve the problem of images denoising was studied, their structure, features, learning and use process, strengths and weaknesses were considered. In the course of experimental studies, the ability of autoencoders to solve the problem of images denoising was confirmed. The advantages and disadvantages of using autoencoders to solve the specified problem were demonstrated. A comparison of various models aimed at solving the specified problem was carried out.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної галузі та постановка задачі.....	10
1.1 Аналіз предметної галузі.....	10
1.2 Формальна задача знешумлення зображень.....	14
1.3 Постановка задачі.....	15
2 Теоретичні дослідження.....	17
2.1 Будова автокодувальника.....	17
2.2 Робота автокодувальника.....	18
2.3 Автокодувальники для знешумлення зображень.....	20
2.4 Сильні та слабкі сторони автокодувальників-знешумлювачів.....	22
3 Використання автокодувальників для знешумлення зображень.....	24
3.1 Інструменти, що використовувались.....	24
3.2 Автокодувальник для знешумлення зображень.....	25
3.3 Ускладнена модель автокодувальника для знешумлення зображень.....	31
3.4 Застосування автокодувальника для боротьби з різними видами цифрового шуму.....	35
3.5 Порівняння двох описаних моделей між собою.....	38
Висновки.....	41
Перелік джерел посилання.....	43
Додаток А Приклади програмних кодів.....	45
Додаток Б Відомість кваліфікаційної роботи.....	54

ВСТУП

В сучасному світі активно використовуються цифрові растрові зображення як один із способів збереження інформації. З'явившись у другій половині минулого сторіччя, вони міцно увійшли як у повсякденне життя, так і у значно вужчі сфери професійної та наукової діяльності. Однак, варто зазначити, що під час створення, передавання каналами зв'язку, обробки, збереження та інших маніпуляцій і процесів растрове зображення буде неминуче забруднюватись шумом. На це явище впливає низка факторів, як-то технічні параметри апаратури, за допомогою якої створюється зображення, обмеження каналів передачі, що використовуються, чи особливості роботи форматів зберігання.

Говорячи про шум в контексті цифрового зображення, варто пояснити, що це випадковий зсув інформації про колір чи яскравість певної частини зображення. На зображеннях шум є небажаним і негативним чинником для їх подальшої обробки чи використання як даних, бо чим більша кількість шумових даних на зображенні, тим складнішим буде виявлення корисної інформації з даного зображення. Подальше використання зображень, забруднених шумом, є ускладненим та може призвести до отримання результатів гіршої якості.

Через вищезазначені фактори, задача повного видалення чи часткового зменшення шуму на зображенні є важливою як для сучасних механізмів обробки візуальної інформації чи систем, що використовують зображення як дані, так і для використання зображень у повсякденних цілях як фінального продукту.

Дану задачу називають задачею знешумлення і вона полягає у тому, щоб усунути із забрудненого зображення шум, або принаймні зменшити інтенсивність його впливу та відновити вихідне зображення, не втративши корисну інформацію, яка на ньому збережена [1].

Виконання знешумлення зображень може бути етапом попередньої обробки у таких напрямках інтелектуальної обробки даних, як обробка медичних зображень чи системи машинного зору, або самостійним процесом під час фотозйомки за складних умов (з недостатньою освітленістю) чи астрофотографії.

Складність та відкритість задачі полягає в тому, що буває непросто виокремити на зображенні шумові елементи та відрізнити їх від корисної інформації, тобто, існує ризик втрати корисної інформації разом з видаленням шуму. Так, надмірне чи неякісне видалення шуму з зображення загрожує зниженням інформаційної цінності даного зображення, до прикладу, розмиття меж об'єктів чи втрати інформації зображеного матеріалу тощо. Очевидно, що часткова втрата інформації знижує цінність зображення як даних для подальшої інтелектуальної обробки [2]. Існують та продовжують з'являтися різні способи вирішення цієї задачі, що використовують різні підходи, але переслідують єдину мету – максимально ефективно очистити зображення від негативного впливу шуму та максимально зберегти при цьому корисну інформацію.

В межах даної роботи розглядатиметься спосіб вирішення задачі знешумлення зображення шляхом застосування інтелектуальних технологій, а саме автокодувальника.

Автокодувальник являє собою тип штучної нейронної мережі, що містить кодувальник, який стискає вихідні дані в певне представлення меншої розмірності, та декодувальник, який із вищезгаданого стиснутого представлення реконструює вихідні дані.

Ідея застосування автокодувальника полягає у тому, що кодувальник зможе уміло та ефективно відкинути шум, що є нерелевантною частиною, зберігши у стиснутому представленні якомога більше корисної інформації, а декодувальник, у свою чергу, якомога більш коректно реконструювати із стиснутого представлення зображення, наближене до вихідного, не

втрачаючи та не спотворюючи корисну інформацію, яку несе зображення [3].

В рамках даної кваліфікаційної роботи розглядатиметься використання автокодувальників для вирішення задачі знешумлення зображень, зокрема, будуть розглянуті теоретичні основи та аспекти будови та використання автокодувальника, будуть проведені експериментальні дослідження, що матимуть на меті установити сильні та слабкі сторони такого підходу.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної галузі

Шум на зображенні є розповсюдженою проблемою, що виникає, першочергово, через технічні обмеження оптики чи сенсора, що використовуються для створення зображення, або несприятливі умови навколишнього середовища.

Зашумленість зображення може варіюватися від майже невидимих плямок на фотографії, знятій при хорошому природньому денному освітленні, які можливо помітити лише прискіпливо роздивляючись зображення, до зображень, майже повністю забруднених шумом, з яких можна отримати порівняно невеликий обсяг корисної інформації, використовуючи такі зображення як дані для використання у інтелектуальних методах обробки.

Шум є вкрай небажаним явищем, оскільки може заважати виокремити чи ідентифікувати об'єкти на зображенні.

В цифрових зображеннях серед джерел шуму можна виокремити:

- складні умови зйомки, такі як низька освітленість під час створення зображення;
- температуру сенсора чи використання сенсора поза його робочим температурним діапазоном;
- потрапляння сторонніх об'єктів (наприклад, пилу) на оптичну систему;
- завади чи обмеження каналів передачі зображення.

На рисунку 1.1 наведено ілюстративний приклад зображення, що забруднене шумом, в порівнянні з тим же зображенням, що не забруднене шумом.



Рисунок 1.1 – Приклад зображення, забрудненого шумом, (праворуч) у порівнянні із чистим зображенням

Формально проблему забруднення зображення шумом можна представити за допомогою математичної рівності (1.1):

$$y = x + n \quad (1.1)$$

де y представляє собою вихідне зображення, забруднене шумом,
 x – невідоме бажане зображення, чисте від шумової компоненти,
 n – шумова компонента,

Говорячи про власне шумову компоненту, варто відмітити, що вона може мати різну природу походження [4], наприклад:

- гаусів шум – має густину ймовірності таку ж саму, як у нормального розподілу. Виникає в електронних приладах, таких, як підсилювачі сигналу і є наслідком теплових коливань електронів та інших випадкових процесів. Приклад зображення із шумом Гауса наведено на рисунку 1.2;



Рисунок 1.2 – Приклад зображення із шумом Гауса [5]

– імпульсний шум (також відомий як «сіль з перцем») – являє собою випадкові пікселі на зображенні, що приймають значення яскравих («сіль») чи темних («перець») плям. Приклад зображення з шумом «сіль з перцем» наведено на рисунку 1.3;



Рисунок 1.3 – Приклад зображення з шумом «сіль з перцем» [6]

– дробовий або пуассонівський шум – описується розподілом ймовірностей Пуассона, що описує ймовірність виникнення випадкової події на інтервалі часу. Цей різновид шуму виникає через статистичну природу фіксації частинок (фотонів), де кількість зафіксованих кожним пікселем подій варіюється згідно із розподілом Пуассона. Цей тип шуму є адаптивним і залежить від рівня сигналу. Приклад зображення з дробовим шумом наведено на рисунку 1.4;



Рисунок 1.4 – Приклад зображення з дробовим шумом [7]

– спекл-шум – виникає через інтерференцію хвиль. Цей різновид шуму зустрічається на зображеннях, отриманих за допомогою ультразвукової, лазерної чи радіолокаційної системи. Відбиті від різних точок об'єкта хвилі накладаються одна на одну, створюючи варіації інтенсивності. Приклад зображення зі спекл-шумом наведено на рисунку 1.5.

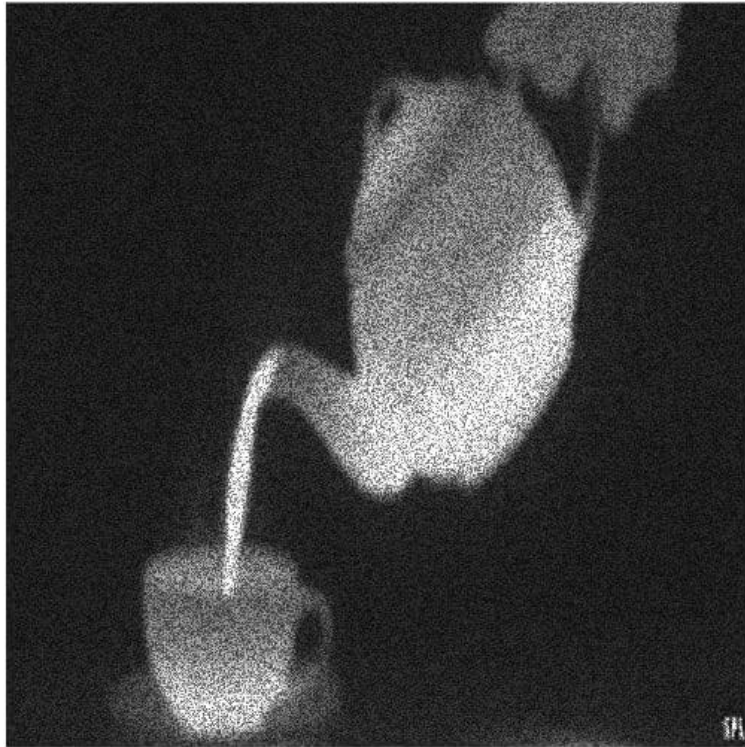


Рисунок 1.5 – Приклад зображення зі спекл-шумом [8]

1.2 Формальна задача знешумлення зображень

Задача знешумлення зображень полягає у відновленні оригінального зображення шляхом видалення або значного зменшення випадкових варіацій кольору чи яскравості, що не несуть корисної інформації та можуть з'являтися на зображенні внаслідок низки чинників, серед яких несприятливе освітлення, висока чутливість чи інші конструктивні обмеження сенсора, який використовується для створення зображень, висока температура, обмеження каналів передачі, через які проходить зображення тощо [17].

Вирішення задачі знешумлення зображень полягає у створенні та застосуванні інструментів, які дозволяють мінімізувати частку шуму, та водночас зберегти важливі структурні характеристики зображення, що

являють собою корисну інформацію, заради збереження, обробки та передачі якої зображення і створюється.

Задача знешумлення зображень є важливою складовою галузі інтелектуальних технологій обробки даних в цілому, оскільки її вирішення дозволяє поліпшити якість даних, які в подальшому можуть бути використані в якості вхідних для інших інтелектуальних систем, таких як розпізнавання облич, обробка медичних зображень, машинний зір. Точність та продуктивність перерахованих систем напряду залежить від якості вхідних даних, що будуть застосовуватись для навчання та відпрацювання роботи систем і які зазвичай представлені у вигляді цифрових зображень.

1.3 Постановка задачі

В ході дослідження обраної предметної галузі було виявлено, що задача знешумлення зображень є відкритою, актуальною, затребуваною і як повністю самостійна задача, і як складова частина різних сфер інтелектуальної обробки візуальних даних. В ході вирішення цієї задачі, зокрема, застосовуються методи машинного навчання та глибокого машинного навчання.

Мета знешумлення зображень – зменшення шуму на зображенні з мінімальною втратою корисної інформації, зокрема, варто відмітити наступне:

- рівні ділянки мають залишитись гладкими;
- краї об'єктів мають лишитись чіткими, не піддатися розмиттю;
- фактура чи текстура матеріалу має бути збережена;
- необхідно уникнути утворення на зображенні нових сторонніх артефактів.

Задача даної кваліфікаційної роботи полягає у дослідженні використання автокодувальників як одного зі способів вирішення задачі знешумлення зображень, що включає в себе:

- дослідження теоретичних основ автокодувальників;
- вивчення архітектури їх побудови;
- розгляд властивостей їхньої роботи із зображеннями;
- виокремлення сильних та слабких сторін застосування автокодувальників для знешумлення зображень;
- прикладне застосування автокодувальників для вирішення задачі знешумлення зображень.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1 Будова автокодувальника

Автокодувальник являє собою особливий тип нейронної мережі з прямим зв'язком, де вхідні дані співпадають з вихідними. Автокодувальник перетворює вхідні дані в стиснуте представлення, яке ще називають кодом, з якого відновлюються вихідні дані.

Автокодувальники, на відміну від класичних способів стиснення даних, є чутливими до самих даних, тобто, якщо модель, збудована за архітектурою автокодувальника, буде натренована для роботи з фотографіями рукописного тексту, вона не зможе адекватно працювати із пейзажними фотографіями [9].

Будь-яка мережа-автокодувальник складається з трьох головних компонентів: кодувальника, вищезгаданого кода (інколи його можуть називати найвужчим місцем за аналогією з англ. *bottleneck*) та декодувальника. Кодувальник та декодувальник є повнозв'язними мережами прямого поширення [10]. Код являє собою одношарову мережу. Розглянемо призначення кожного із елементів:

- кодувальник – модуль, що відповідає за перетворення вихідних даних (набору даних) у стиснуте представлення, що містить основні характеристики вхідного зображення та має розмірність меншу, ніж початкові дані;

- код – модуль, що зберігає стиснуте представлення, відповідно, є найважливішим. Завдяки обмеженню потоку інформації від кодувальника до декодувальника забезпечується уловлювання важливої інформації та відкидання незначущої;

- декодувальник – модуль, задачею якого є реконструювати вихідний результат із стиснутого представлення якомога ближче до вхідних даних шляхом інвертування операцій, проведених кодувальником. На

цьому шарі проводиться поступове підвищення дискретизації представлення і зрештою формується нове зображення, аналогічне за розмірністю до вхідного [19].

Узагальнену архітектуру автокодувальника зображено на рисунку 2.1.

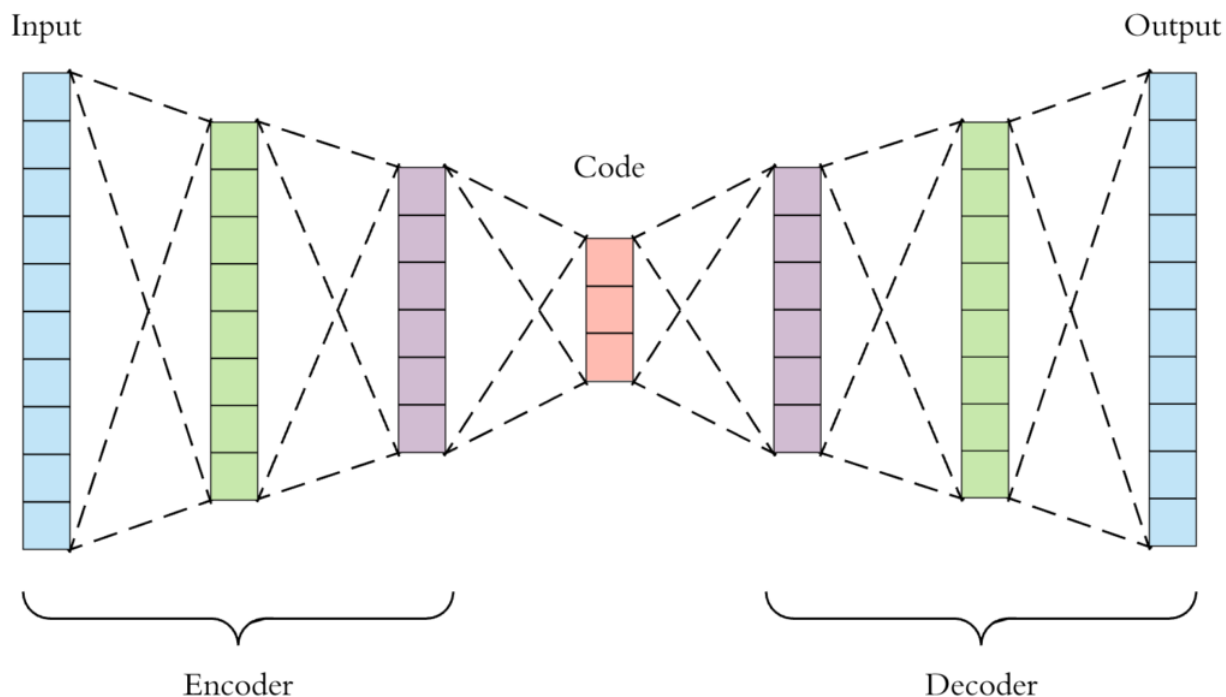


Рисунок 2.1 – Ілюстрація архітектури автокодувальника

2.2 Робота автокодувальника

Кодувальник, отримуючи на вхід дані, продукує стиснуте представлення, з якого декодувальник, структура якого дзеркально подібна до кодувальника, реконструює нове вихідне зображення, базуючись лише на стисненому представленні. Метою є отримання вихідного зображення якомога більш наближеного до вхідного зразка.

Очевидно, що для того, аби вихідний результат був якомога ближчим до вхідного, модель має на етапі створення стисненого представлення провести певну інтелектуальну роботу з виокремлення особливостей і закономірностей, що формують цінну частину зображення як даних та

їхнього збереження у стисненому представленні. Якщо шаблони, закономірності та особливості будуть виокремлені неякісно чи некоректно, то результатом обробки зображення із використанням такої моделі стане часткова втрата чи спотворення корисної інформації, що робить марним використання такої моделі для інтелектуальної обробки даних у вигляді зображень.

Принцип роботи мережі-автокодувальника ілюстративно зображено на рисунку 2.2.

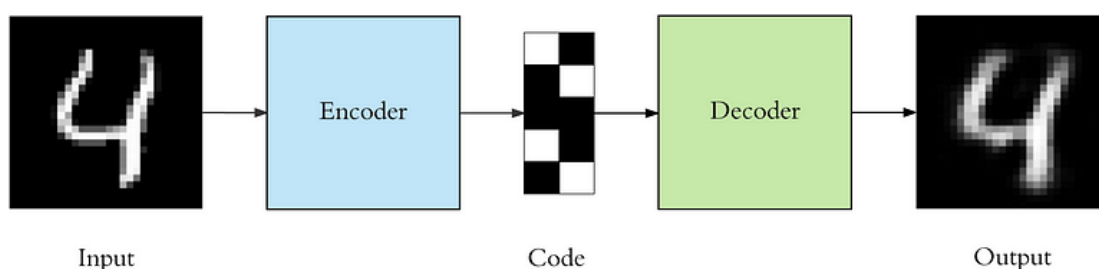


Рисунок 2.2 – Ілюстрація роботи мережі-автокодувальника [9]

Для навчання автокодувальників задаються чотири гіперпараметри, а саме [11]:

- розмір кода – найважливіший із параметрів, що визначає ступінь стиснення даних. Надто сильне стиснення створює ризик втрати корисної інформації і надмірно спрощеного представлення, недостатнє може призвести до явища перенавчання (англ. *overfitting*), тобто збереження некорисної частини даних разом із корисною. В такому випадку модель фактично не проводить жодної інтелектуальної роботи і більшою мірою копіює вхідні дані на вихід;
- глибина – кількість шарів кодувальника та декодувальника. Як і в будь-яких штучних нейронних мережах, є важливим параметром, що визначає складність та продуктивність моделі;

– кількість вузлів на кожному шарі – зазвичай, архітектуру автокодувальника можна неформально охарактеризувати як «пісковий годинник», тобто на кожному шарі кодувальника кількість вузлів поступово зменшується, на кожному шарі декодувальника навпаки, зростає. З точки зору структури шарів, декодувальник зазвичай є симетричним до кодувальника, оскільки має виконувати зворотні операції;

– функція втрат при реконструкції – залежить від типу вхідних даних, але можуть застосовуватись середньоквадратична помилка (MSE) чи бінарна крос-ентропія. Значення функції втрат, власне, є кількісною характеристикою роботи нейронної мережі.

Навчання автокодувальників відбувається за процедурою навчання без учителя із використанням зворотного поширення.

Оскільки розмірність стисненого представлення є меншим за розмірність вхідних даних, то модель вимушена виокремлювати в даних найважливішу та корисну інформацію, а не просто копіювати вхідні дані на вихід. Надто велика розмірність стисненого представлення призведе до явища, відомого у галузі інтелектуальної обробки даних, як перенавчання, коли модель фактично просто створюватиме копію вхідного зображення, не проводячи власне жодної інтелектуальної роботи.

2.3 Автокодувальники для знешумлення зображень

В контексті розглянутої в попередньому розділі задачі знешумлення зображень можна використовувати штучні нейронні мережі-автокодувальники.

Автокодувальники, за своєю суттю, призначені для виявлення і виокремлення значущих ознак та закономірностей у вхідних даних, дозволяючи при цьому відкидати менш важливу інформацію зради того, аби створити стиснене представлення (код).

Якщо вхідні дані мережі-автокодувальника являють собою забруднені шумом зображення, то, очевидно, перед мережею постане задача відокремити корисні дані від некорисних, аби створити представлення зменшеної розмірності. Мережа навчається виокремлювати корисну інформацію на зображенні (до прикладу, межі об'єктів, фактура чи колір матеріалу тощо) і зберігати її в представленні меншої розмірності, ігноруючи шум, який в даному випадку з точки зору моделі є менш значущою частиною даних, а тому відкидається і не зберігається у стисненому представленні.

Очевидно, що якщо під час створення представлення меншої розмірності шум відкидався як неважлива частина даних, то при роботі декодувальника нове реконструйоване зображення не матиме шуму, оскільки декодувальник реконструює нове зображення, базуючись лише на наявному всередині моделі стисненому представленні, куди раніше були допущені лише ті дані, які були визнані значущими. Як результат, зображення, отримане внаслідок роботи автокодувальника-знешумлювача, є близьким за змістом до оригінального вхідного зображення, проте позбавлене тої частини візуальної інформації, що була визначена як шум і відкинута [12].

Процес перетворення забрудненого шумом зображення на чисте ілюстративно зображено на рисунку 2.3.

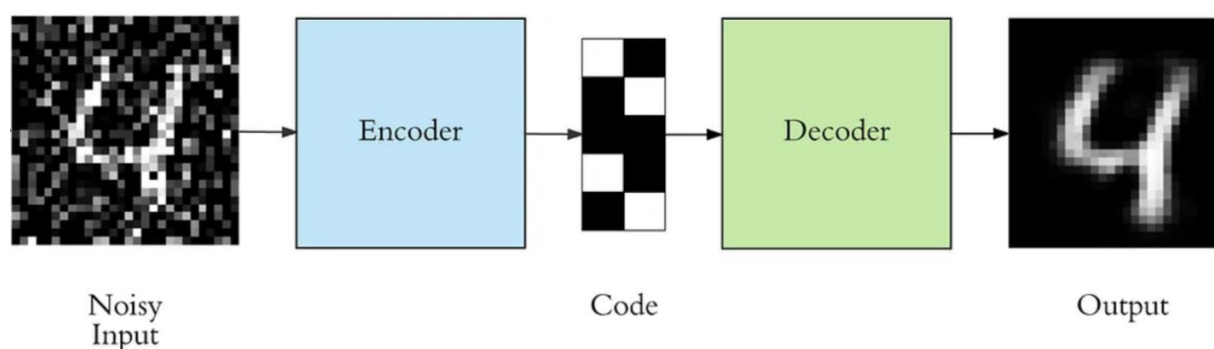


Рисунок 2.3 – Ілюстрація роботи автокодувальника для знешумлення зображень

2.4 Сильні та слабкі сторони автокодувальників-знешумлювачів

В рамках використання штучних нейронних мереж-автокодувальників для вирішення задачі знешумлення зображень можна виокремити наступні сильні сторони цієї технології:

- ефективність видалення різних видів шуму – так, завдяки інтелектуальному виокремленню закономірностей і властивостей у візуальних даних, автокодувальники здатні ефективно боротися із шумом різних типів та різної природи виникнення;

- здатність до відтворення зображень – автокодувальники можуть достатньо точно реконструювати зображення, наближене до початкового, не втрачаючи та не спотворюючи корисну інформацію, що зберігається у даному зображенні;

- здатність до узагальнення – такі нейронні мережі мають змогу виявляти доволі складні структурні залежності чи шаблони у візуальних даних, що допомагає ефективно працювати з різними типами зображень та ефективно виокремлювати корисну інформацію від небажаної шумової компоненти;

- автоматизація процесу – автокодувальники використовують технологію навчання без учителя, тому все, що необхідно для роботи з ними, це набір даних.

Однак, можна виокремити і деякі слабкі сторони:

- необхідність значної вибірки даних для якісного навчання – як і у випадку з майже будь-якими штучними нейронними мережами, для якісного виокремлення залежностей чи закономірностей, моделі необхідний великий обсяг навчальних даних для тренування. У випадку із навчанням моделі на замалій вибірці, можливе неефективне виокремлення шуму на зображеннях;

- підбір гіперпараметрів на етапі проектування та реалізації моделі може бути складним, у випадку застосування мережі із невірною

заданими гіперпараметрами є ризик перенавчання, тобто «адаптації» моделі під конкретні дані чи навпаки недостатня її точність і втрата корисної інформації;

– в залежності від параметрів конкретної мережі, її навчання та застосування можуть вимагати значних обчислювальних та часових ресурсів, що може поставити під сумнів доцільність застосування такого способу знешумлення в деяких випадках.

3 ВИКОРИСТАННЯ АВТОКОДУВАЛЬНИКІВ ДЛЯ ЗНЕСУМЛЕННЯ ЗОБРАЖЕНЬ

3.1 Інструменти, що використовувались

Дослідження використання автокодувальників для вирішення задачі знесумлення зображень виконувались із використанням мови програмування Python, яка є звичним, гнучким, потужним та безкоштовно поширюваним інструментом для розв'язання різних задач зі сфери машинного навчання та інтелектуальної обробки даних. Використовувалась 64-бітна версія Python версії 3.10.

Також було використано перелічені нижче розширення:

- NumPy – розширення, що додає підтримку значної кількості математичних сутностей та операцій, як-то багатовимірні масиви і матриці. В даній роботі використовувалось для створення цифрового шуму на зображеннях із набору даних [13];

- Matplotlib – розширення для візуалізації даних двовимірною графікою. В даній роботі використовувалось для роботи із візуалізацією зразків із набору даних [14];

- Tensorflow та Keras – розширення, розроблені для задоволення потреб у побудові й тренуванні нейронних мереж. В даній роботі використовувалось для роботи із функціоналом створення та навчання нейронних мереж [15].

Оскільки далі в тексті даного розділу фігуруватимуть параметри та час, що було витрачено на навчання моделей, варто зазначити, що в усіх випадках навчання моделей відбувалось на особистому комп'ютері форм-фактору ноутбук, що має двоядерний процесор із тактовою частотою 2.3 ГГц (повне маркування «Intel(R) Pentium(R) CPU 4415U @ 2.30GHz»), обсяг оперативної пам'яті складає 4 Гб, та керується 64-бітною операційною системою Windows 10.

В якості вихідних даних розглядався набір даних MNIST, що являє собою колекцію зображень рукописних цифр від 0 до 9 та постачається разом із розширенням Tensorflow. Приклад зображень із набору даних наведено на рисунку 3.1.

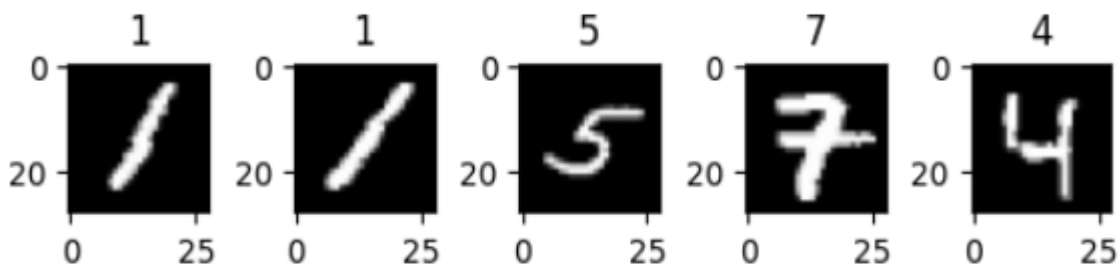


Рисунок 3.1 – приклад даних із набору даних MNIST

Оскільки надані у наборі даних зображення є незабрудненими цифровим шумом, необхідно штучним чином додати до них шум, перед тим як переходити до розгляду власне вирішення задачі знешумлення на прикладі цих даних.

3.2 Автокодувальник для знешумлення зображень

Пропонується розглянути автокодувальник, що прийме на вхід зображення розміром $28 * 28$ та перетворюватиме його на стиснуте представлення у вигляді вектора із 128 елементів.

Перша частина, а саме кодувальник, складається із серії згорткових та повнозв'язних шарів. Ця частина відповідає за перетворення вхідного зображення, забрудненого шумом, на стиснуте представлення в прихованому просторі, що містить основні характеристики вхідного зображення, фіксують основні шаблони та структурні особливості. У якості функції активації використовується ReLU, додаючи на рівні кодувальника нелінійність шляхом встановлення нульових значень для негативних

виходів та збереження значень позитивних виходів. Це дозволяє кодувальнику вивчати представлення даних та виявляти важливі характеристики.

Наступна частина моделі, а саме декодувальник, відповідає за реконструкцію зображень із стиснутого представлення в прихованому просторі. Ця частина складається із декількох повнозв'язних та транспонованих згорткових шарів. Шари, з яких складається декодувальник, виконують роботу, обернену до тої, що виконується кодувальником на попередньому етапі. Поступово підвищуючи розмірність представлення, декодувальник, зберігаючи раніше виокремлені характеристики, відновлює зображення, що за своїми розмірами відповідатиме вхідному. На рівні декодувальника використовуються функції активації Sigmoid та ReLU. Сигмоїдна активуюча функція відображає вихід кожного нейрона на ймовірнісне значення в діапазоні $[0, 1]$.

В якості функції втрат розглядається середня квадратична помилка (MSE).

В якості оптимізатора використовується Adam.

Для отримання вхідних даних до зображень з набору даних додається випадковий гаусівський шум [16].

Модель передбачає наступні шари для кодувальника:

- вхідний шар з указаним параметром `input_shape = (28 * 28 * 1)`, що відповідає розмірам вхідного зображення. Даний параметр визначає форму вхідних даних, а саме дозволяє моделі обробляти зображення розміром $28 * 28$ з одним каналом (відтінки сірого);

- два згорткових шари із кількістю фільтрів 32 та 64 відповідно, розміром ядра 3×3 , кроком 2, функцією активації ReLU. Перший із цих шарів з 32 фільтрами виявляє базові шаблони на зображенні, в той час другий, що має удвічі більшу кількість фільтрів, відповідає за виявлення

локальних патернів. Таке поєднання дозволяє ефективно виявляти на забрудненому шумом зображенні корисну інформацію, яку варто зберегти;

- шару Flatten, що перетворює результат роботи згорткових шарів на одновимірний вектор. Таке перетворення необхідно для подальшого введення даних в повнозв'язні шари;

- повнозв'язний Dense-шар виконує остаточне стискання даних у вектор із 128 елементів, зберігаючи основні характеристики вхідного зображення.

А також наступні шари для декодувальника:

- dense-шар із нейронами $7 * 7 * 64$, що відповідає останній формі представлення перед перетворенням на одновимірний вектор. Така операція необхідна для підготовки даних до подальшого розгортання в початкові розміри;

- reshape, що перетворює стиснуте представлення у багатовимірне, що необхідно для подальших транспонованих згорткових операцій;

- два транспонованих згорткових шари, один із яких має 32 фільтри та функцію активації ReLU, а інший 1 фільтр та функцію активації Sigmoid. Розмір ядра 3×3 , крок 2 в обох шарах. Перший із описаних шарів відповідає за збільшення розмірів представлення до розмірів початкового зображення, задача ж другого, що має один фільтр, в остаточному відновлення зображення та нормалізації значень пікселів.

Загальну будову моделі, а саме кількість та типи шарів, з яких вона складається, а також кількість параметрів, із якими вона працюватиме, зображено на рисунку 3.2.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 28, 28, 1)]	0
conv2d (Conv2D)	(None, 14, 14, 32)	320
conv2d_1 (Conv2D)	(None, 7, 7, 64)	18496
flatten (Flatten)	(None, 3136)	0
dense (Dense)	(None, 128)	401536
dense_1 (Dense)	(None, 3136)	404544
reshape (Reshape)	(None, 7, 7, 64)	0
conv2d_transpose (Conv2DTranspose)	(None, 14, 14, 32)	18464
conv2d_transpose_1 (Conv2DTranspose)	(None, 28, 28, 1)	289
=====		
Total params: 843,649		
Trainable params: 843,649		
Non-trainable params: 0		

Рисунок 3.2 – Загальна будова моделі автокодувальника для знешумлення зображень

Навчання моделі відбувається у 10 епох із розміром пакета 128. В результаті навчання моделі із заданими параметрами зайняло близько 11 хвилин, а помилка на тестовій частині набору даних склала 0.0104, що є прийнятним результатом. Оскільки використовується механізм ранньої зупинки, що зупиняє навчання, якщо значення функції втрат не зменшується протягом п'яти епох поспіль, навчання зайняло усього лише вісім епох із зазначених у первинно зазначених параметрах десяти. Детальніше

результати навчання моделі із зазначеними параметрами наведено на рисунку 3.3.

```

epochs = 10
batch_size = 128

history = autoencoder.fit(x_train_noisy, x_train, validation_data=(x_test_noisy, x_test),
                           epochs=epochs, batch_size=batch_size, callbacks=[early_stopping])

```

[11] ✓ 11m 2.2s

```

... Epoch 1/10
469/469 [=====] - 65s 133ms/step - loss: 0.0090 - val_loss: 0.0103
Epoch 2/10
469/469 [=====] - 68s 144ms/step - loss: 0.0088 - val_loss: 0.0103
Epoch 3/10
469/469 [=====] - 70s 149ms/step - loss: 0.0087 - val_loss: 0.0103
Epoch 4/10
469/469 [=====] - 69s 148ms/step - loss: 0.0086 - val_loss: 0.0103
Epoch 5/10
469/469 [=====] - 79s 169ms/step - loss: 0.0085 - val_loss: 0.0103
Epoch 6/10
469/469 [=====] - 118s 252ms/step - loss: 0.0084 - val_loss: 0.0104
Epoch 7/10
469/469 [=====] - 88s 187ms/step - loss: 0.0084 - val_loss: 0.0104
Epoch 8/10
469/469 [=====] - 98s 208ms/step - loss: 0.0083 - val_loss: 0.0104

```

Рисунок 3.3 – Результати навчання запропонованої моделі

Наочну демонстрацію роботи вищеописаної штучної нейронної мережі-знешумлюючого автокодувальника наведено на рисунку 3.4, де у верхньому рядку наведено приклади зображень із початкового набору даних, у середньому – ці ж зображення, штучно забруднені гауссовим шумом перед побудовою нейромережі власне для проведення експериментальних досліджень, у нижньому рядку – результат знешумлення цих же зображень, отриманий із використанням вищеописаної нейронної мережі.

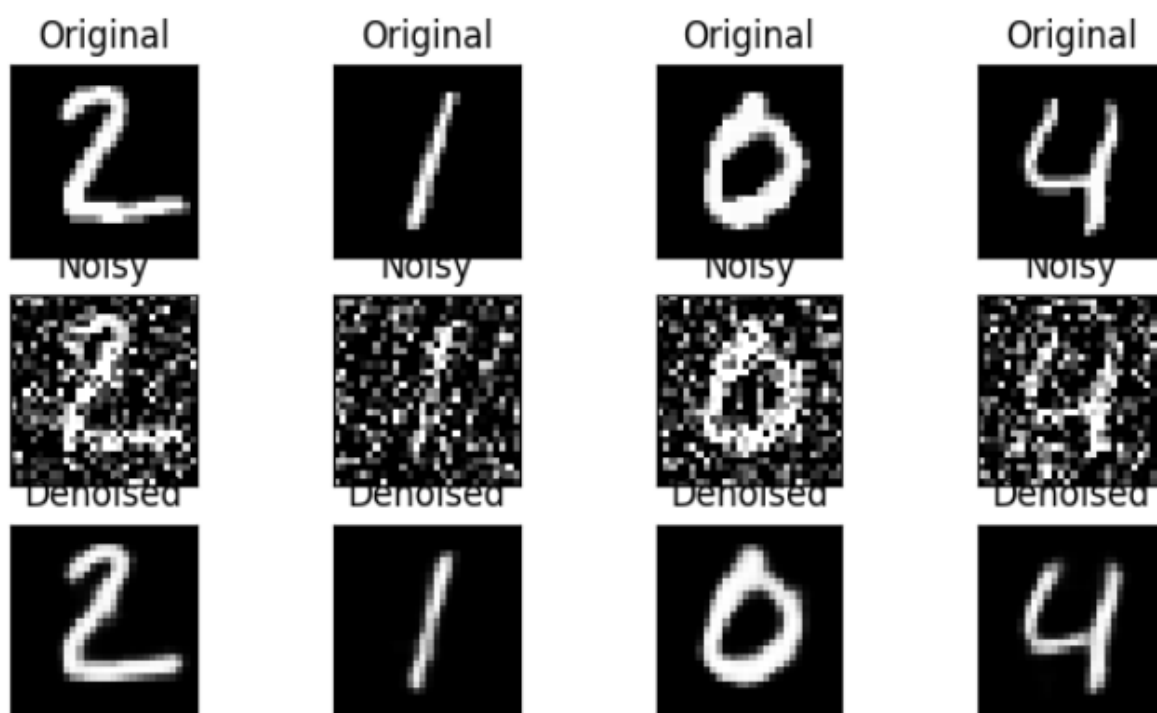


Рисунок 3.4 – Демонстрація знешумлення зображень із використанням автокодувальника

На рисунку 3.4 можна візуально оцінити роботу запропонованої моделі для знешумлення зображень. Моделі, що найголовніше, вдалося відновити цінну інформацію на зображеннях (у даному випадку це силует рукописної цифри на кожному із них), не спотворивши його і не втративши його елементів. Це означає, що модель підтвердила здатність коректно виокремлювати та зберігати основні характеристики зображення, що обробляється.

Одночасно з цим вдалося візуально очистити зображення від шумової компоненти, інтенсивність якої на забруднених зразках видно неозброєним оком. Отже, шляхом експериментального дослідження автокодувальник здатен ефективно відфільтрувати шум і відновлювати зображення до стану, наближеному до початкового.

В цілому, прикладне застосування автокодувальника для знешумлення зображень підтвердило описані у підрозділі 2.4 даної роботи

переваги автокодувальника як знешумлювача, а саме здатність до узагальнення та інтелектуального виявлення корисної інформації, що зберігається у зображенні, та здатність до достатньо точного відтворення початкового зображення без втрати чи спотворення вищезгаданою корисної інформації, а отже і до коректного та ефективного очищення зображень від цифрового шуму.

3.3 Ускладнена модель автокодувальника для знешумлення зображень

Для вирішення тієї ж задачі знешумлення зображень, забруднених цифровим шумом, пропонується нейронна мережа дещо ускладненої будови, ніж у попередньому підрозділі, що у якості вхідних даних отримуватиме зображення розмірністю $28 * 28$.

Перші декілька шарів нейронної мережі, що виконують роль кодувальника, складаються зі згорткових шарів (Convolutional layers), за якими ідуть агрегувальні шари (MaxPooling layers).

Згорткові шари на даному етапі використовуються для створення ядра згортки, завдяки чому на оброблюваному зображенні виділяються важливі ознаки, такі як межі об'єкта.

Щодо ж агрегувальних шарів, то їхньою задачею є поступове зменшення розмірності представлення по мірі проходження від поточного етапу до наступного, беручи та зберігаючи максимальне значення у вікні введення.

Як уже було сказано у попередньому розділі, шари кодувальника, зменшуючи роздільну здатність зображення, створюють тимчасове представлення (код), у якому зберігається лише виокремлена кодувальником корисна частина інформації (features).

Отримане з шарів-кодувальника представлення (код) надалі передається у шари, які разом можна назвати декодувальником. Їхньою

задачею є реконструювати із стисненого представлення зображення розмірністю такою, що збігатиметься із розмірністю вихідного зображення. Декодувальник складається із декількох транспонованих згорткових шарів (Transposed Convolutional layers) та шарів збільшення розмірності (UpSampling layers).

Транспоновані згорткові шари на етапі декодування виконують роботу, зворотню до власне згорткових шарів, що використовувались на етапі кодування, тобто дозволяють, зберігаючи раніше виявлений шаблон зв'язності, перейти від деякого представлення (згортки) до вихідних даних, що матимуть форму, аналогічну вхідним.

Щодо ж шарів збільшення розмірності, їхня задача впливає із їхньої назви, тобто отримання із стисненого представлення вихідного зображення, розміри якого будуть не меншими за розміри вхідних даних, які отримувала модель [18].

В якості функції активації на усіх шарах як кодувальника, так і декодувальника використовується ReLU [12]. Функція ReLU встановлює нульові значення для негативних виходів, лишаючи позитивні виходи незмінними. Це допомагає моделі навчатися нелінійним залежностям у даних.

Узагальнену будову моделі, а саме кількість та типи шарів, з яких складається модель, а також кількість параметрів, із якими вона працюватиме, зображено на рисунку 3.5.

Також на вищезазначеному рисунку можна побачити, що на відміну від моделі, описаної в попередньому підрозділі, розмір прихованого представлення удвічі більший, кількість шарів та параметрів також більша.

На рисунку 3.6 наведено результати тренування моделі за наступних умов – модель навчається протягом 10 епох із використанням пакету із 64 зображень, а перевіряється на тестовій частині набору даних.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 28, 28, 1)]	0
conv2d_3 (Conv2D)	(None, 24, 24, 64)	1664
max_pooling2d_3 (MaxPooling 2D)	(None, 12, 12, 64)	0
conv2d_4 (Conv2D)	(None, 10, 10, 128)	73856
max_pooling2d_4 (MaxPooling 2D)	(None, 5, 5, 128)	0
conv2d_5 (Conv2D)	(None, 3, 3, 256)	295168
max_pooling2d_5 (MaxPooling 2D)	(None, 1, 1, 256)	0
conv2d_transpose_4 (Conv2DTranspose)	(None, 3, 3, 256)	590080
conv2d_transpose_5 (Conv2DTranspose)	(None, 5, 5, 128)	295040
up_sampling2d_2 (UpSampling 2D)	(None, 10, 10, 128)	0
conv2d_transpose_6 (Conv2DTranspose)	(None, 12, 12, 64)	73792
up_sampling2d_3 (UpSampling 2D)	(None, 24, 24, 64)	0
conv2d_transpose_7 (Conv2DTranspose)	(None, 28, 28, 1)	1601
=====		
Total params: 1,331,201		
Trainable params: 1,331,201		
Non-trainable params: 0		

Рисунок 3.5 – Узагальнена будова пропонованої моделі

```

autoencoder.fit(x_train_noisy, x_train, epochs=10, batch_size=64, validation_data=(x_test_noisy, x_test))
✓ 106m 14.2s

Epoch 1/10
938/938 [=====] - 643s 678ms/step - loss: 0.0300 - val_loss: 0.0174
Epoch 2/10
938/938 [=====] - 662s 706ms/step - loss: 0.0162 - val_loss: 0.0152
Epoch 3/10
938/938 [=====] - 694s 740ms/step - loss: 0.0144 - val_loss: 0.0140
Epoch 4/10
938/938 [=====] - 624s 666ms/step - loss: 0.0135 - val_loss: 0.0136
Epoch 5/10
938/938 [=====] - 616s 657ms/step - loss: 0.0129 - val_loss: 0.0133
Epoch 6/10
938/938 [=====] - 631s 673ms/step - loss: 0.0125 - val_loss: 0.0130
Epoch 7/10
938/938 [=====] - 627s 668ms/step - loss: 0.0121 - val_loss: 0.0129
Epoch 8/10
938/938 [=====] - 620s 661ms/step - loss: 0.0118 - val_loss: 0.0130
Epoch 9/10
938/938 [=====] - 639s 681ms/step - loss: 0.0116 - val_loss: 0.0128
Epoch 10/10
938/938 [=====] - 614s 654ms/step - loss: 0.0114 - val_loss: 0.0129

<keras.callbacks.History at 0x29abd05f070>

```

Рисунок 3.6 – Результати тренування моделі

Для оцінки точності використовується середня квадратична помилка (MSE).

Для налаштування вагових коефіцієнтів було використано оптимізатор Adam.

Після навчання моделі із вищезазначеними параметрами точність на тестовій частині набору даних склала 0,0129. Навчання ускладненої моделі зайняло набагато більше часу – за такої ж кількості епох, як і у моделі з попереднього підрозділу, процес навчання розтягнувся більше ніж на півтори години, а саме 106 хвилин, тобто 1 годину та 46 хвилин.

Приклад роботи вищеописаної моделі наведено на рисунку 3.7. Як і в попередньому підрозділі, у верхньому рядку наведено приклади початкових зображень із набору даних, у середньому рядку наведено ці ж зображення, забруднені цифровим шумом, в нижньому рядку наведено результати знешумлення цих зображень із використанням ускладненої моделі.

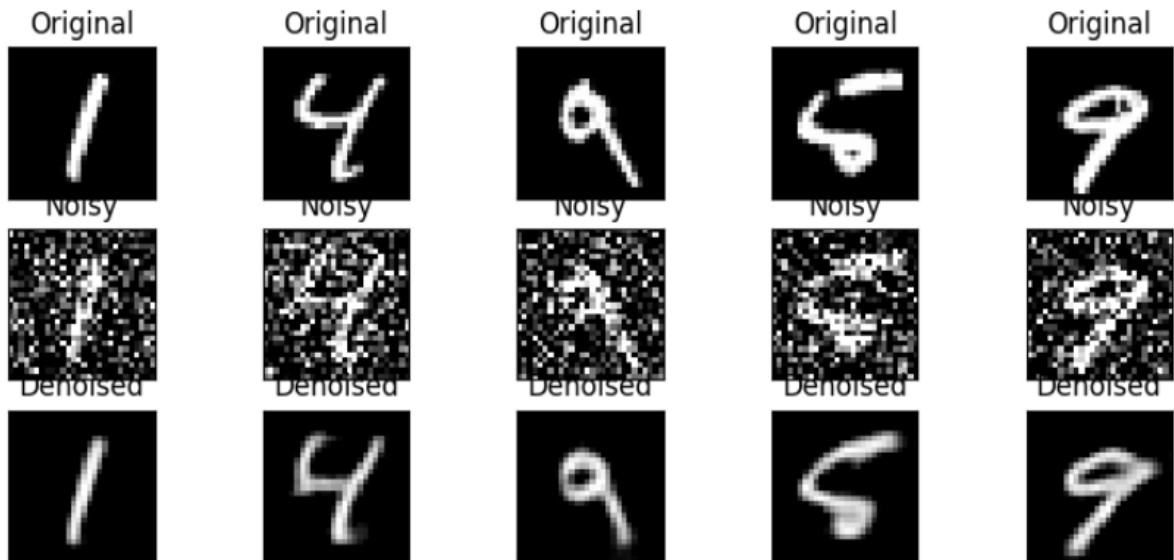


Рисунок 3.7. – Демонстрація знешумлення зображень із використанням ускладненої моделі автокодувальника

3.4 Застосування автокодувальника для боротьби з різними видами цифрового шуму

В рамках ускладненої моделі також розглядалися декілька видів цифрового шуму, що забруднює зображення, а саме наступні:

- гаусів шум
- імпульсний шум (також відомий як «сіль з перцем»)
- пуасонівський шум
- спекл-шум

Про особливості та відмінності одне від іншого видів цифрового шуму детальніше розповідалося у першому розділі даної роботи.

На рисунку 3.8 зображено приклади даних, ушкоджених вищезазначеними різними виглядами шуму.

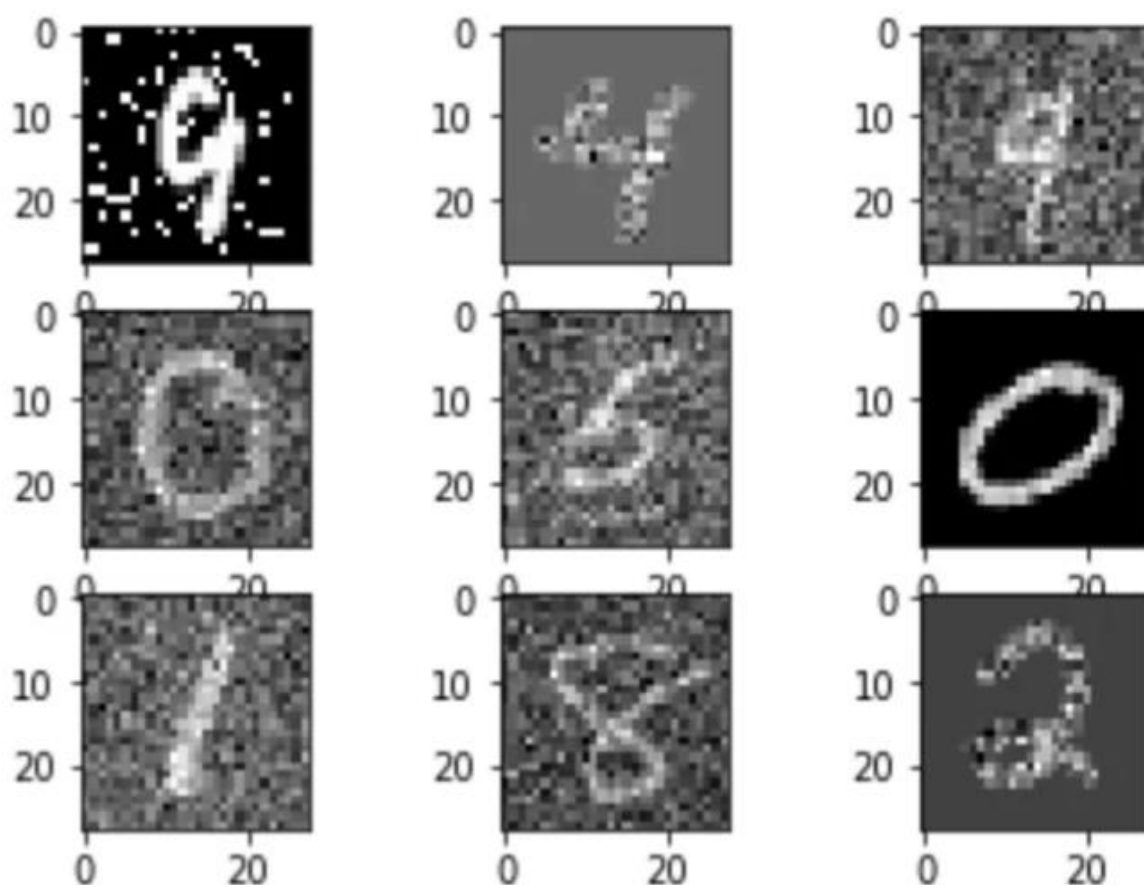


Рисунок 3.8 – Приклади даних, забруднених різними видами цифрового шуму

Будова моделі не зазнала змін і описана в попередньому підрозділі.

На рисунку 3.9 наведено результати тренування моделі за наступних умов – модель навчається протягом 10 епох із використанням пакету із 64 зображень, а перевіряється на тестовій частині набору даних.

Як видно з рисунку 3.9, час, потрібний для навчання моделі, принципових змін не зазнав. Після навчання моделі із вищезазначеними параметрами середня квадратична помилка на повному наборі даних склала 0.0233. Дана зміна відносно результатів роботи моделі, описаної в попередньому підрозділі, є очікуваною, враховуючи ускладнення забруднення вхідних даних різними видами шуму, та не має принципово

негативно відобразитись на здатності моделі проводити роботу зі знешумлення зображень.

На рисунку 3.10 наведено приклад зображень, забруднених цифровим шумом – саме такі зображення є вхідними даними для автокодувальника-знешумлювача.

```

autoencoder.fit(train_noisy, train_clean, epochs=10, batch_size=64, validation_data=(test_noisy, test_clean))
[18] ✓ 105m 48.1s
... Epoch 1/10
938/938 [=====] - 653s 691ms/step - loss: 0.0392 - val_loss: 0.0284
Epoch 2/10
938/938 [=====] - 636s 678ms/step - loss: 0.0270 - val_loss: 0.0262
Epoch 3/10
938/938 [=====] - 631s 673ms/step - loss: 0.0252 - val_loss: 0.0253
Epoch 4/10
938/938 [=====] - 644s 687ms/step - loss: 0.0244 - val_loss: 0.0246
Epoch 5/10
938/938 [=====] - 639s 682ms/step - loss: 0.0238 - val_loss: 0.0241
Epoch 6/10
938/938 [=====] - 637s 679ms/step - loss: 0.0235 - val_loss: 0.0242
Epoch 7/10
938/938 [=====] - 624s 665ms/step - loss: 0.0232 - val_loss: 0.0237
Epoch 8/10
938/938 [=====] - 622s 664ms/step - loss: 0.0229 - val_loss: 0.0236
Epoch 9/10
938/938 [=====] - 628s 669ms/step - loss: 0.0227 - val_loss: 0.0237
Epoch 10/10
938/938 [=====] - 631s 672ms/step - loss: 0.0226 - val_loss: 0.0233

```

Рисунок 3.9 – Результати тренування моделі

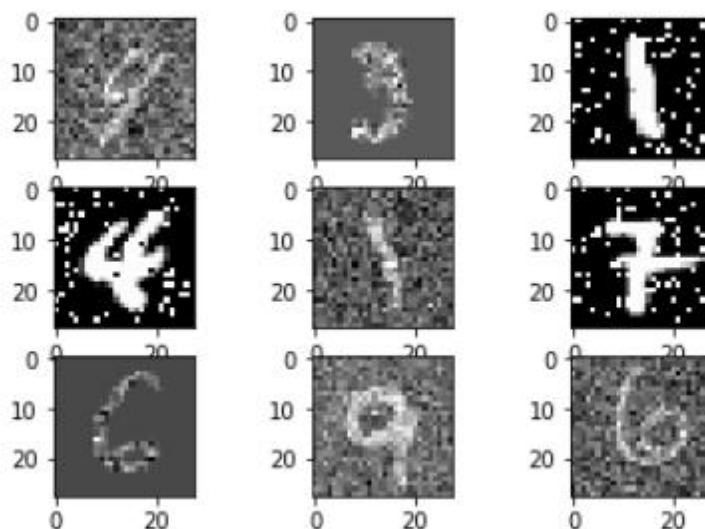


Рисунок 3.10 – Приклади зашумлених зображень

На рисунку 3.11 зображено результат знешумлення вищенаведених зображень із застосуванням вищеописаної ускладненої моделі автокодувальника. На рисунку можна побачити, що зображення вдалося візуально очистити від шуму, який перед проведенням знешумлення можна було побачити неозброєним оком. Можна наочно побачити, що знешумлення із використанням автокодувальника виявилось ефективним проти різних видів цифрового шуму, що підтверджує одну із наведених у підрозділі 2.4 переваг використання автокодувальників для знешумлення зображень, а саме здатність ефективно видаляти шум різних типів та різної природи походження.

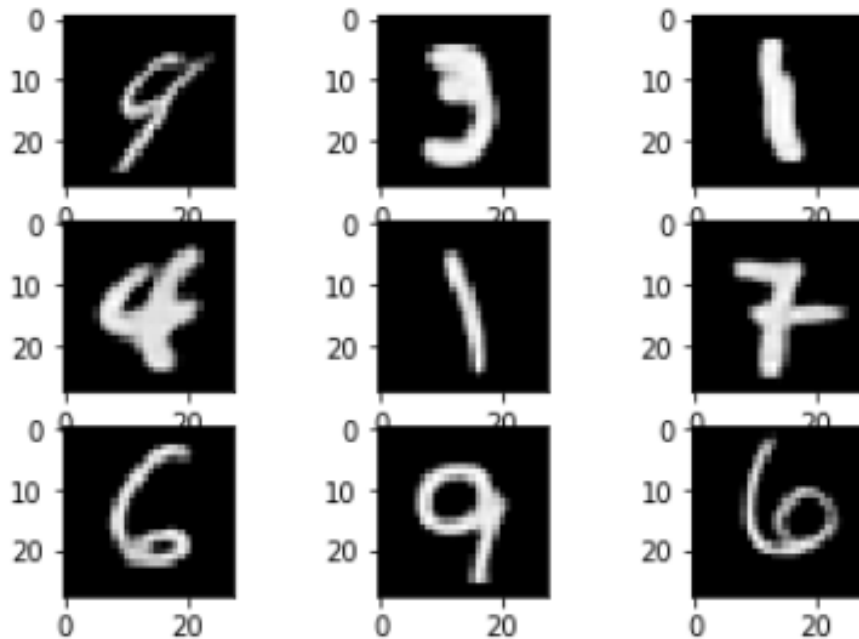


Рисунок 3.11 – Результат знешумлення зображень із застосуванням автокодувальника

3.5 Порівняння двох описаних моделей між собою

У трьох попередніх підрозділах було детально описано дві моделі автокодувальників, створених із однією і тією ж метою – знешумлення

зображень. Зазначені моделі мали різну кількість та різні типи шарів, різну розмірність прихованого представлення, але працювали на одному і тому самому початковому наборі даних MNIST, що постачається разом із розширенням Tensorflow. Це дає можливість порівняти отримані результати наочним чином, side-by-side.

Ускладнена модель з підрозділу 3.3 мала більшу кількість шарів, а саме 11 проти 9 у моделі з підрозділу 3.2. Також в ускладненій моделі, на відміну від попередньої, представлені шари MaxPooling. Ускладнена модель має на 58% більше параметрів за ту, що наведена у підрозділі 3.2. Ускладнена модель мала удвічі більший розмір вектора, що є стиснутим представленням у прихованому просторі. Та яким чином описана кількісна та якісна різниця характеристик проявилась у прикладній площині?

Для оцінки точності обох моделей використовувались одні і ті ж самі методи, а саме середня квадратична помилка з оптимізатором Adam. Помилка в ускладненій моделі після 10 епох навчання склала 0.0129. Помилка в попередній моделі з підрозділу 3.2 склала 0.0104. Видно, що ускладнена модель проявила дещо більше значення функції втрат після тренування на тому ж наборі даних із застосуванням аналогічного шуму, що неприємно здивувало, хоча і не стало критичною проблемою.

Втім, окрім власне значення функції втрат є ще і інший кількісний параметр – час, який було витрачено на навчання моделей. Модель з підрозділу 3.2 для процесу навчання з 10-ти епох витратила дещо більше 11 хвилин, ускладнена ж модель навчалась близько 105-ти хвилин за ідентичного значення кількості епох. Різниця більше ніж у 10 разів, що є суттєвим.

Тренування ускладненої моделі для знешумлення зображень, забруднених різними видами цифрового шуму, описане в підрозділі 3.4, показало незначне зростання середньої квадратичної помилки, а саме з 0.0129 до 0.0233, що саме по собі не виглядає критичною проблемою та є очікуваним, виходячи з більшої складності вхідних даних. Втім, проблемою

залишилися значні часові витрати, що їх потребувала ускладнена модель для навчання, особливо в порівнянні з базовою моделлю, описаною в підрозділі 3.2

Таким чином, експериментальні дослідження показали, що принцип «чим більше, тим краще» не завжди можна застосовувати в проектуванні та реалізації інтелектуальних систем. Наочно наведено приклад, коли збільшення кількості шарів, збільшення розмірності стиснутого представлення, збільшення параметрів не тільки не призвели до поліпшення точності моделі, навпаки, призвели до хай незначного, але збільшення помилки на тестовому наборі даних, але, що найголовніше, призвели до великого (більше, ніж удесятеро) зростання часу, що потребує модель для навчання. Така різниця у витраті часових ресурсів вкупі з відсутністю виграшу в точності показали недоцільність прикладного використання ускладненої моделі, описаної в підрозділах 3.3 та 3.4, для вирішення задачі знешумлення зображень.

Втім, описані експериментальні дослідження очікувано підтвердили і сильні сторони використання автокодувальників для знешумлення зображень, що були описані в підрозділі 2.4. Моделі підтвердили здатність видаляти із зображень різні види цифрового шуму, ефективно виокремлюючи при цьому корисну інформацію з забруднених зображень, та точно відтворювати зображення у наближеному до первісного вигляді без спотворень чи втрат корисної інформації.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було виконано усі поставлені завдання.

Було проведено роботу із попереднього аналізу предметної галузі, а саме розгляду поняття цифрового шуму на зображеннях, його видів, природи та причин його виникнення. Було проведено роботу з розгляду сучасного стану задачі знешумлення зображень як самостійної задачі та як складової частини інших задач машинного зору та інтелектуального аналізу даних.

Було проведено теоретичне дослідження, що включало у себе розгляд будови нейронних мереж-автокодувальників, призначення та особливості складових частин їхньої архітектури. Дослідження проводилося шляхом, зокрема, опрацювання джерел, а саме статей, публікацій, літератури. Було сформульовано можливі сильні та слабкі сторони застосування автокодувальників для вирішення задачі знешумлення зображень.

Було опрацьовано практичне використання нейронних мереж-автокодувальників як способу вирішення задачі знешумлення зображень. Зокрема, було розглянуто дві різні за будовою моделі, обидві з яких реалізують боротьбу із цифровим шумом на зображеннях, було порівняно результати їхньої роботи, зокрема, точність та час, що витрачається на тренування моделей.

Шляхом експериментальних досліджень було підтверджено раніше сформульовані теоретично сильні сторони використання автокодувальників для вирішення задачі знешумлення зображень. Окрім сильних сторін, проявилися і слабкі, а саме, одна з розглянутих моделей, як виявилось, вимагає суттєво більших часових витрат на тренування в порівнянні з іншою.

Втім, було наочно доведено можливість використання штучних нейронних мереж-автокодувальників для вирішення задачі знешумлення зображень.

В ході виконання даної кваліфікаційної роботи було використано і закріплено знання та навички, отримані з вивчення попередніх дисциплін, зокрема, таких як обчислювальний інтелект, основи інтелектуального аналізу даних, комп'ютерний зір, когнітивні системи та моделі, глибинне навчання нейронних мереж, основи наукових досліджень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Марченко М. Використання автокодувальників для знешумлення зображень. *Радіоелектроніка та молодь у XXI столітті. Т. 6 : Конференція "Інформаційні інтелектуальні системи"*: матеріали 28-го Міжнар. молодіж. форуму, м. Харків, 16 квіт. 2024 р. Харків, 2024. С. 115–116.
2. Brief review of image denoising techniques / L. Fan et al. *Visual Computing for Industry, Biomedicine, and Art*. 2019. Vol. 2, no. 1. URL: <https://doi.org/10.1186/s42492-019-0016-7> (date of access: 10.05.2024).
3. Elad M., Kawar B., Vaksman G. Image Denoising: The Deep Learning Revolution and Beyond.
4. Swain A. Noise in digital image processing. 2018. URL: <https://medium.com/@anishaswain/noise-in-digital-image-processing-55357c9fab71> (дата звернення 03.04.2024).
5. Lenzen, Frank. Statistical Regularization and Denoising.
6. Deepti G. P., Borker M. V., Sivaswamy J. Impulse Noise Removal from Color Images with Hopfield Neural Network and Improved Vector Median Filter. *Image Processing (ICVGIP)*, Bhubaneswar, India, 16–19 December 2008. 2008. URL: <https://doi.org/10.1109/icvgip.2008.75> (date of access: 10.05.2024).
7. The curious case of poisson noise and MATLAB ‘imnoise’ command. 2016. URL: <https://ruiminpan.wordpress.com/2016/03/10/the-curious-case-of-poisson-noise-and-matlab-imnoise-command/> (дата звернення 05.04.2024).
8. Buenestado P., Acho L. Image Segmentation Based on Statistical Confidence Intervals. *Entropy*. 2018. Vol. 20, no. 1. P. 46. URL: <https://doi.org/10.3390/e20010046> (date of access: 10.06.2024).
9. Dertat A. Applied deep learning - part 3: autoencoders. 2019. URL: <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798> (дата звернення 14.04.2024).

10. Jordan J. Introduction to autoencoders. 2018. URL: <https://www.jeremyjordan.me/autoencoders/> (дата звернення: 16.04.2024).
11. Bandyopadhyay H. Autoencoders in deep learning: tutorial & use cases. 2021. URL: <https://www.v7labs.com/blog/autoencoders-guide> (дата звернення 18.04.2024).
12. Chaudhary K. Denoising AutoEncoders can reduce noise in images. 2023. URL: <https://medium.com/game-of-bits/denoising-autoencoders-can-reduce-noise-in-images-5b74753eaf97> (дата звернення 19.04.2024).
13. NumPy documentation URL: <https://numpy.org/doc/stable> (дата звернення 02.05.2024)
14. Matplotlib 3.9.0 documentation URL: https://matplotlib.org/stable/api/matplotlib_configuration_api.html (дата звернення 02.05.2024)
15. Module: tf.keras | Tensorflow v2.16.1 URL: https://www.tensorflow.org/api_docs/python/tf/keras (дата звернення 02.05.2024)
16. Keshavarz A. Image denoising using autoencoders (improved version). 2023. URL: <https://medium.com/@a.keshavarz/image-denoising-using-autoencoders-improved-version-5f8a90019971> (дата звернення 05.05.2024).
17. Boncelet C. Chapter 7 - image noise models. *The essential guide to image processing*. 2009. P. 143–167.
18. Kılıç İ. What is max pooling and why do we need max pooling?. 2023. URL: <https://medium.com/@ilyurek/what-is-max-pooling-and-why-do-we-need-max-pooling-57247a3fbca9> (дата звернення 07.05.2024).
19. What are autoencoders and why are they used?. *Buff ML*. URL: <https://buffml.com/what-are-autoencoders-and-why-are-they-used/> (дата звернення 09.05.2024).
20. What is image denoising?. *Saiwa*. 2023. URL: <https://medium.com/@saiwa.dotai/what-is-image-denoising-2c27c97caf29> (дата звернення 12.05.2024).