Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод нанесення цифрового водяного знаку на текстури тривимірних об’єктів

(тема)

Виконав:

студент II курсу, групи СПм-20-2

Смирнов В.О.
(прізвище, ініціали)

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

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

Освітня програма Системне програмування
(повна назва освітньої програми)

Керівник: доц. Мартовицький В.О.
(посада, прізвище, ініціали)

Допускається до захисту

В.о. зав. кафедри ЕОМ _____ Волк М.О.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Смирнову Владиславу Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Метод нанесення цифрового водяного знаку на текстури тривимірних об'єктів

затверджена наказом по університету від " 24 " березня 2022 р. № 413 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 18 травня 2022р.

3. Вхідні дані до роботи Тип обладнання – Nvidia Tesla P100
текстури тривимірних об'єктів, мова програмування – Python

4. Перелік питань, що потрібно опрацювати у роботі
аналіз різновидів систем водяних знаків
аналіз сфер використання систем водяних знаків
аналіз характеристик водяних знаків
аналіз існуючих методів вбудови водяних знаків в текстури та їх зчитування
створення систем водяних знаків та їх порівняння
аналіз впливу атак на текстури на зчитані з них водяні знаки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 14 слайдів

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	30.03.22-15.04.22	
2	Аналіз існуючих методів вбудови водяних	16.04.22-23.04.22	
	знаків в текстури та їх зчитування		
3	Створення систем водяних знаків	24.04.22-03.05.22	
4	Порівняння розроблених систем	04.05.22-09.05.22	
5	Аналіз впливу атак на текстури на зчитані	10.05.22-13.05.22	
	з них водяні знаки		
6	Оформлення матеріалів кваліфікаційної роботи		
7	Подання кваліфікаційної роботи на	14.05.22-18.05.22	
	рецензування		

Дата видачі завдання 28 березня 2022 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Мартовицький В.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 68 с., 20 рис., 5 табл., 2 дод., 20 джерел.

ЦИФРОВИЙ ВОДЯНИЙ ЗНАК, ТЕКСТУРИ, ТРИВИМІРНІ ОБ’ЄКТИ, НЕЙРОННА МЕРЕЖА, ДИСКРЕТНЕ КОСИНУСНЕ ПЕРЕТВОРЕННЯ, ДИСКРЕТНЕ ВЕЙВЛЕТНЕ ПЕРЕТВОРЕННЯ, KERAS, PYTHON, OPENCV

Метою кваліфікаційної роботи є дослідження систем для роботи з цифровими водяними знаками, а також порівняння різних методів вбудови та зчитування водяних знаків з текстур тривимірних об’єктів.

У ході виконання кваліфікаційної роботи були розглянуті існуючі методи вбудови водяних знаків, розроблено дві системи, що базуються на різних алгоритмах. Вони були порівняні між собою за точністю зчитування вбудованих цифрових водяних знаків, а також стійкістю до різноманітних атак, що погіршують точність роботи систем зчитування.

ABSTRACT

Master’s thesis: 68 pages, 20 figures, 5 tables, 2 appendices, 20 sources.

DIGITAL WATERMARK, TEXTURE, 3D OBJECTS, NEURAL NETWORK, DISCRETE COSINUS TRANSFORM, DISCRETE WAVELET TRANSFORM, KERAS, PYTHON, OPENCV

The major goal of this thesis is to study digital watermarking systems, as well as to compare different methods of embedding and extracting watermarks from the textures of 3D objects.

In order to investigate what watermarking system is better, methods of embedding watermarks were considered. Two systems based on different algorithms were developed. They were compared to each other in terms of accuracy of extraction of digital watermarks built into texture, as well as resistance to various attacks that degrade the accuracy of extraction systems.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 8
ВСТУП 9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 10
1.1 Основи стеганографії 10
1.2 Види систем вбудови цифрових водяних знаків 12
1.3 Застосування цифрових водяних знаків 13
1.4 Характеристики ЦВЗ 14
1.4.1 Непомітність 14
1.4.2 Надійність 15
1.4.3 Ємність 15
1.4.4 Безпечність 16
1.5 Атаки на ЦВЗ 17
1.6 JPEG стиснення 19
2 АНАЛІЗ МЕТОДІВ ТА АЛГОРИТМІВ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ 20
2.1 Методи просторової області 21
2.2 Методи частотної області 22
2.2.1 Дискретне косинусне перетворення 22
2.2.2 Дискретне вейвлетне перетворення 23
2.2.3 Дискретне перетворення Фур’є 25
2.2.4 Сингулярне розкладання значень 25
2.3 Опис використаних технологій 26
2.3.1 PyWavelets 26
2.3.2 Keras 27
2.3.3 OpenCV 28
3 СИСТЕМИ ДЛЯ РОБОТИ З ЦИФРОВИМИ ВОДЯНИМИ ЗНАКАМИ 34

3.1 Опис детермінованого алгоритму 35
3.1.1 Алгоритм вбудови водяного знаку 36
3.1.2 Алгоритм зчитування водяного знаку 38
3.2 Опис розробленої нейронної мережі 39
3.3 Вимоги до текстур та водяних знаків 44
4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ 46
ВИСНОВКИ 56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 57
ДОДАТОК А Графічний матеріал кваліфікаційної роботи 60
ДОДАТОК Б Загальна схема розробленої нейронної мережі 68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN – Convolutional Neural Network

ЦВЗ – цифровий водяний знак

ДКП – дискретне косинусне перетворення

ДВП – дискретне вейвлетне перетворення

ДПФ – дискретне перетворення Фур’є

НМ – нейронна мережа

СРЗ – сингулярне розкладання значень

ВСТУП

Розвиток нових технологій і зростаючий обсяг даних в цифровому середовищі змінюють екосистему сучасних медіа. Будь-які дані можуть бути з легкістю продубльовані. Це приносить зручність в повсякденне життя, роботу, наукові дослідження. Але з цим все більшої актуальності набирає тема захисту інтелектуальної власності.

Щосекунди через інтернет відбувається обмін величезними обсягами даних. Частина цих даних передається у відкритому вигляді й може бути отримана сторонніми особами. У такому вигляді, зазвичай, поширюються різноманітні зображення, відео та інший контент, який в тому числі може бути платним.

Виставляючи свій витвір в інтернет творці розраховують знайти потенційного покупця чи просто похизуватися створеним. Або, наприклад, у видавничій індустрії мультимедіа дослідники, науковці та інженери-практики надають великого значення поширеним досягненням, оскільки несанкціоноване маніпулювання та необмежене відтворення оригінальних цифрових мультимедіа можна легко поширювати через канали зв’язку, такі як Інтернет. Отже, існує реальна потреба в ефективних методах для забезпечення захисту від несанкціонованого доступу та запобігання завданню шкоди власникам цифрового контенту. Через можливі проблеми з авторським правом авторські права та інтелектуальна власність цифрових мультимедійних даних мають бути захищені від незаконного володіння, тиражування та поширення. Для цього використовуються такі засоби як шифрування, стеганографія та водяні знаки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основи стеганографії

Стеганографія – спосіб передачі або зберігання інформації з урахуванням збереження в таємниці самого факту такої передачі (зберігання). На відміну від криптографії, що приховує вміст таємного повідомлення, стеганографія приховує факт його існування.

Перевага стеганографії над чистою криптографією у тому, що повідомлення не привертають себе уваги. Повідомлення, факт шифрування яких не прихований, викликають підозру. Таким чином, криптографія захищає зміст повідомлення, а стеганографія – сам факт наявності будь-яких прихованих послань від викриття. Основні поняття стеганографії:

- стеганографічна система (стегосистема) – об’єднання методів і засобів, що використовуються для створення прихованого каналу для передачі інформації;
- повідомлення – це термін, що використовується для загальної назви прихованої інформації, що передається, будь то лист або цифровий файл;
- контейнер – так називається будь-яка інформація, яка використовується для приховування таємного повідомлення. Порожній контейнер – контейнер, який не містить секретного послання. Заповнений контейнер (стегоконтейнер) – контейнер, що містить таємне послання;
- стегоканал – канал передачі стегоконтейнера;
- ключ (стегоключ) – секретний ключ, необхідний для приховування стегоконтейнера. Ключі в стегосистемах бувають двох типів: секретні та відкриті. Якщо стегосистема використовує секретний ключ, то він має бути створений або до початку обміну повідомленнями, або переданий захищеним каналом. Стегосистема, що використовує відкритий ключ, має бути такою, щоб неможливо було отримати з нього закритий ключ. У цьому випадку

відкритий ключ може передаватись незахищеним каналом.

Цифрова стеганографія – це напрямок класичної стеганографії, заснований на приховуванні або впровадження додаткової інформації в цифрові об’єкти, викликаючи деякі зміни цих об'єктів. Але, як правило, ці речі є мультимедіа-об’єктами (текстури, відео, аудіо, текстури, 3D-об’єкти) і внесення модифікацій, які будуть непомітні для середньо-статистичної людини, не призводить до значних змін цих об’єктів. Крім того, в оцифрованих об'єктах, що мають аналогову природу, завжди присутній шум квантування; далі, при відтворенні цих об’єктів з'являється додатковий аналоговий шум та неточності викликані апаратурою, все це сприяє більшій непомітності прихованої інформації.

Одним з проявів розвитку стеганографії та криптографії є цифрові водяні знаки. Вони з’явилися у спробах подолати обмеження шифрування та стеганографії для забезпечення та захисту прав інтелектуальної власності. У порівнянні з ідеєю шифрування, інформація про водяний знак вставлена в початкову форму не заважає користувачам слухати, переглядати, дивитися або маніпулювати даними. І на відміну від стеганографії, технології цифрових водяних знаків (ЦВЗ) використовуються для встановлення ідентичності інформації, щоб уникнути несанкціонованого використання. Як правило, додаткова інформація вбудовується безпосередньо в вихідний мультимедійний файл або хост-сигнал.

Зберігання зображень у цифровому форматі спрощує їх зберігання та розповсюдження, але також збільшує ризик порушення авторських прав, несанкціонованої зміни та розповсюдження. З метою захисту інтелектуальної власності та визначення змін, розробляються та застосовуються ЦВЗ. До таких стеганографічних методик пред’являються особливі вимоги:

- якість вхідної текстури не повинна бути серйозно порушена, помітність прихованих даних повинна бути мінімальною;

- приховані дані повинні зберігатися у різних форматах, тобто утримуватися не лише у заголовку, а у всьому тілі цифрового об’єкта;

- приховані дані повинні бути стійкими до навмисних спроб видалення;

- необхідна наявність надлишкового коду для корекції помилок, оскільки деградація даних під час передачі/модифікації неминуча.

Гарним контейнером для впровадження в них прихованої інформації можуть бути цифрові зображення. Це пов’язано з фактом певної надмірності візуальної інформації. Популярність використання цифрових зображень як стегоконтейнерів обумовлена такими причинами:

- існуванням практично значущим завданням захисту фотографій, відео від незаконного тиражування та розповсюдження;

- великим обсягом пропускної спроможності зображень, що дозволяє впроваджувати ЦВЗ великого обсягу чи підвищувати надійність використання;

- наявністю у більшості реальних зображень текстурних областей, що мають шумову структуру та підходять для вбудовування інформації;

- слабкою чутливістю людського ока до незначних змін кольорів зображення, його яскравості, контрастності, змісту шуму, спотворень поблизу контурів.

1.2 Види систем вбудови цифрових водяних знаків

Розрізняють декілька різновидів систем ЦВЗ в залежності від інформації, яка потрібна системі щоб виявити ЦВЗ – оригінал текстури, ЦВЗ, секретний ключ чи додаткова інформація [3]. Розглянемо визначення кожного виду.

Системи сліпих ЦВЗ. У системах такого типу для зчитування ЦВЗ потрібна лише текстура з вбудованим знаком та секретний ключ. Такі системи не вимагають оригінальної текстури. Основними сферами застосуваннями сліпих водяних знаків є охорона здоров’я, захист авторських прав, електронні системи голосування.

Системи несліпих ЦВЗ. У таких системах для зчитування водяного знаку необхідне і оригінальна текстура, і знання секретного ключа. Потенційним застосуваннями цього типу систем водяних знаків є приховане спілкування та захист авторських прав.

Системи напівсліпих ЦВЗ. Такі стеганосистеми виявляють ЦВЗ за допомогою секретного ключа, який залежить від оригінальної текстури. Деякі з важливих застосувань таких систем - аутентифікація текстур, моделі CAD тощо.

1.3 Застосування цифрових водяних знаків

Дослідники використовують різні схеми водяних знаків для численних застосувань, що розвиваються. Застосування водяних знаків включають захист авторських прав, цифрову криміналістику, військову, цифрову судову експертизу, охорону здоров’я, медичні програми тощо. Нижче приведені найбільш популярні способи застосування.

Захист авторських прав. Основною метою таких програм по вбудовуванню водяних знаків є захист авторських прав на цифрову інформацію шляхом приховування в ній секретної інформації.

Моніторинг трансляцій. Такі програми дозволяють власникам медіа автоматично перевіряти, коли, де і скільки часу їх трансляції передавалися через кабельне чи супутникове телебачення.

Фінгерпринтинг. Це процес, за допомогою якого вміст водяного знаку містить ідентифікаційну інформацію передбачуваного одержувача, щоб потім було можливо відстежити джерело розповсюдження.

Медичне застосування. Методи водяних знаків забезпечують аутентифікацію та конфіденційність медичних даних. Дослідження показало, що використання водяних знаків забезпечують додаткову цінність інструментів безпеки для додатків охорони здоров’я.

Системи електронного голосування. Інтернет уже охопив майже весь

світ завдяки швидкому розвитку комп'ютерної мережі. Електронна система голосування – це процес супроводу виборів шляхом збереження безпеки під час виборів. Завдяки використанню водяних знаків, досягається повна безпека при обробці голосів електорату наданих за допомогою глобальної мережі.

Захист даних при хмарному збереженні. Зі збільшенням кількості даних, які зберігаються на хмарних сховищах, стала актуальною проблема перевірки прав доступу до певних конфіденційних даних. Всі конфіденційні документи, медіа-файли чи будь-що інше необхідно засвідчити перед передачею їх за запитом. Хмарний сервер може вставляти унікальний водяний знак до, наприклад, зашифрованих зображень, і потім перед передачею зображення користувачеві відбувається додаткова перевірка, що даний користувач дійсно має право отримати доступ до бажаних даних.

1.4 Характеристики ЦВЗ

Важливо визначити критерії системи водяних знаків для порівняння результатів різних методів роботи з водяними знаками. Очевидно, що кожна система повинна мати особливі властивості щодо даного застосування; тому не існує унікального набору властивостей, яким повинні задовольняти всі системи водяних знаків. Загалом, є п'ять важливих характеристик, які зазвичай розглядаються [1]. Вони висвітлені в наступних підрозділах.

1.4.1 Непомітність

Непомітність є важливою умовою цифрових водяних знаків. Візуальна подібність між версією з водяним знаком та оригінальним елементом медіа та якість сприйняття оригінального сигналу мають бути непомітно трансформовані шляхом вставки водяного знаку. Існують дві основні причини, чому важливо зберегти непомітність після кодування з даними

водяного знаку. По-перше, наявність чи відсутність водяного знаку неможливо відрізнити виходячи з основного призначення медіа-файлу, якщо носій із водяним знаком настільки сильно спотворений, що його цінність втрачена. Крім того, підозрілі помітні артефакти можуть позначати наявність водяного знаку, і його точне розташування може бути виявлене. Ця інформація може надавати можливість для зловмисного спотворення, заміни або видалення даних ЦВЗ. Тому вбудована в нього інформація може стати недоступною.

1.4.2 Надійність

Однією з найбільш часто вимірюваних властивостей є те, наскільки водяні знаки стійкі до різних атак і поширених операцій обробки сигналів у системах цифрових водяних знаків. Після того, як певний сигнал ЦВЗ було вбудовано в оригінальний носій, цей носій може піддаватися зовнішньому впливу під час кодування, декодування та розповсюджується через Інтернет. Ці зміни можуть виконуватися для очікуваних перетворень до сигналів з водяними знаками або стиснення їх перед передачею. Такі перетворення можуть значно порушити сигнали з водяними знаками. Неможливо, щоб система водяних знаків була надійною проти всіх операцій обробки сигналів, тому вимоги надійності залежать від застосування. Наприклад, для цифрових водяних знаків зображень хороший метод нанесення водяних знаків має протистояти додаванню шуму, обробці фільтрації, геометричним перетворенням, як масштабування і повороту, а також стиснення JPEG.

1.4.3 Ємність

Ємність визначає найбільшу кількість інформації, яку можуть приховати вставлені водяні знаки, а вбудовані водяні знаки можна надійно використати для цілей аутентифікації та захисту авторських прав. За умови

непомітності, а також вимог надійності, ємність залежить від розміру вихідних даних. Чим більший розмір оригінальних даних, тим більше бітів можна вставити. Проте вставка великого обсягу інформації у вигляді водяного знаку є складним завданням.

Ємність визначається в залежності від практичне застосування носія водяного знаку. Для аудіо ємність буде пов’язана з кількістю вставлених бітів у кожну секунду. Для зображень ємність буде залежати від кількості вбудованих бітів у пікселі. Для відео ємність означає кількість бітів у кожній секунді або кількість бітів на кадр. Одним словом, чим меншим за кількістю бітів є водяний знак, тим більше можливостей є для його вставки з максимальною обчислювальною складності взлому; менша кількість помилкових спрацьовувань.

Умови непомітності, стійкості та надійності суперечать один одному та обмежуються. Можна підвищити надійність водяного знаку, але це призводить до більшої помітності ЦВЗ. З іншого боку, за умови непомітності водяний знак повинен бути створений з максимально можливим розділенням по оригінальним даних, щоб уникнути ситуації, коли невелике пошкодження зображення водяного знаку призведе до помилкового виявлення водяного знаку. Аналогічно, можна збільшити корисне навантаження даних, зменшуючи кількість вибірок, виділених кожному прихованому біту, але це врівноважується втратою надійності. Іншими словами, для будь-якої схеми водяних знаків неможливо виконати ці три вимоги одночасно. Як наслідок, між цими вимогами має бути досягнутий баланс.

1.4.4 Безпечність

Усі існуючі алгоритми водяних знаків, які не є безпечними, не можна використовувати для захисту авторських прав, аутентифікації даних або відстеження незаконного розповсюдження цифрового вмісту. Таким чином, алгоритм водяних знаків є безпечним і надійним, якщо зловмисник,

використовуючи процедури та знання водяних знаків, не зможе визначити використаний ключ. Таким чином, прихована інформація про водяний знак не може бути знищена або пошкоджена. Крім того, складність процесу створення ЦВЗ може бути пов’язана з безпекою, оскільки складність вставки, зазвичай, підвищує і складність знаходження водяного знаку.

1.5 Атаки на ЦВЗ

У більшості програм для нанесення водяних знаків, позначені дані, ймовірно, будуть певним чином оброблені, перш ніж вони досягнуть адресата. Обробкою може бути стиснення з втратами, посилення сигналу або цифро-аналогове (ЦАП) та аналого-цифрове (АЦП) перетворення. Така обробка може ненавмисно пошкодити вбудований водяний знак. Інші типи обробки можуть бути застосовані з явною метою перешкодити зчитуванню водяних знаків. У термінології водяних знаків, атака – це будь-яка обробка, яка може зашкодити зчитуванню ЦВЗ або передачі інформації, що передається у вигляді водяного знаку. Тоді оброблені дані з водяними знаками називаються атакованими [5]. Основні види атак наведені нижче.

Активні атаки. Під час цієї атаки хакер намагається навмисно видалити водяний знак або зробити так, щоб його було неможливо виявити. Вони спрямовані на те, щоб спотворити вбудований водяний знак до невпізнання. Це велика проблема в захисті авторських прав, фінгерпринтингу або контролю копіювання.

Пасивні атаки. У цьому випадку зловмисник не намагається видалити водяний знак, а просто намагається визначити, чи присутні водяний знак у певному стегоконтейнері. Захист від пасивних атак має надзвичайне значення у прихованих комунікаціях, де просте знання про наявність ЦВЗ часто більше, ніж те, що хочеться надати.

Атаки підробки. Це, ймовірно, головна проблема аутентифікації даних. Під час атак націлених на підробку, хакер прагне вбудувати новий дійсний

водяний знак, а не видалити його. Роблячи це, він дозволяє змінювати захищені дані, а потім повторно імплантувати новий ключ, щоб замінити вбудований раніше, таким чином пошкоджене зображення виглядає справжнім.

Атаки змови. У цих атаках мета хакера така ж, як і в активних атаках, але метод дещо інший. Щоб видалити водяний знак, хакер використовує кілька копій одних і тих самих даних, які містять різні водяні знаки, кожна з яких підписана певним ключем, щоб створити нову копію без ЦВЗ. Це проблема в програмах фінгерпринтингу (наприклад, у кіноіндустрії), але вона не є широко поширеною, оскільки зловмисник повинен мати доступ до кількох копій одних і тих самих даних, і ця кількість є досить важливою.

Прості атаки. Інша назва цієї атаки – атака за формою сигналу та атака за шумом. Це називається простими атаками, оскільки вони націлені на пошкодження вбудованого ЦВЗ, змінюючи весь водяний знак без спроби його розпізнати. Деякі приклади цих атак: фільтрація, додавання шуму, стиснення на основі форми сигналу (JPEG, MPEG), гамма-корекція.

Криптографічні атаки. Головною метою таких атак є порушення методу безпеки в техніках водяних знаків і пошук режиму видалення вставленої інформації про водяний знак. Через високу обчислювальну складність застосування цих атак обмежено.

Атака на видалення. Ці атаки націлені на повне видалення даних водяного знаку без порушення безпеки техніки водяних знаків. Ця техніка використовує шуми та квантування.

Геометрична атака. На відміну від атак видалення, геометричні атаки фактично не видаляють сам вбудований водяний знак, але мають на меті зламати синхронізацію детектора ЦВЗ з вбудованою інформацією. Детектор може відновити вбудовану інформацію ЦВЗ лише при повній синхронізації. Однак складність необхідного процесу синхронізації може бути занадто великою, щоб ця атака була дійсно практичною.

1.6 JPEG стиснення

JPEG – популярний графічний формат, який характеризується високим ступенем стиснення, що призводить до зниження якості текстури. У ньому використовується технологія кодування плавної передачі кольору, що забезпечує можливість зменшення кількості даних під час збереження текстур. Через невеликий розмір вихідних файлів він зазвичай використається, коли треба зберігати великі обсяги графічних матеріалів. Він також часто використовується в картах пам'яті цифрових відеокамер. Алгоритм JPEG оптимально підходить для стиснення фотографій, текстур і зображень, в яких присутні реалістичні сцени з невеликим контрастом кольорів. Цей формат не рекомендується використовувати для стиснення малюнків і різного роду графіки, оскільки сильний контраст між рядом близько розташованих пікселів провокує появу видимих артефактів.

Процедура стиснення текстур у форматі JPEG здійснюється в кілька етапів. Спочатку фотографія перетворюється в колірний простір YCbCr, потім розбивається на квадрати для визначення верхнього діапазону колірного спектру. Нарешті, кольори та яскравість кодуються. JPEG використовує систему стиснення з втратами та технологію дискретного косинусного перетворення. Коефіцієнт стиснення файлів може задаватися перед початком роботи алгоритму. У цьому випадку зниження якості текстури може варіюватися від незначного до істотного.

2 АНАЛІЗ МЕТОДІВ ТА АЛГОРИТМІВ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ

Окремий компонент системи відповідає за вбудову ЦВЗ в зображення. В даному випадку вхідними даними є оригінальне зображення, з якого буде створено стегоконтейнер, а також повідомлення, яке має бути вбудованим у вигляді водяного знаку [2].

Рисунок 2.1 – Схематичне зображення компонента, що вбудовує ЦВЗ

З іншого боку є компонент, який буде використовуватися для зчитування змісту ЦВЗ зі стегоконтейнеру. Вхідною інформацією для декодеру є різний набір даних, який залежить від обраного типу системи вбудови ЦВЗ. Наприклад для несліпих систем ЦВЗ, вхідними даними є текстура, що містить водяний знак, оригінальна текстура та дані, які мали бути вбудовані в стегоконтейнер. На виході цей модуль повідомляє чи є в заданому контейнері ЦВЗ та, у випадку його наявності, його вміст.

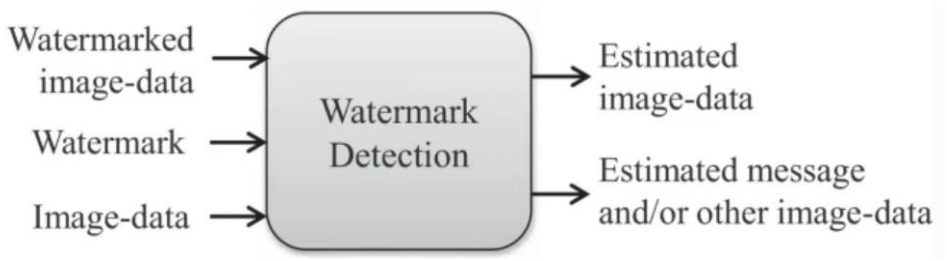

Рисунок 2.2 – Схематичне зображення компонента, що зчитує ЦВЗ

У сфері ЦВЗ, вбудова водяних знаків у цифрові зображення привернула велику увагу дослідників з двох причин: перша – це її доступність, а інша – це передача достатньої кількості додаткової інформації, яку можна було б використовувати для вбудовування ЦВЗ. Техніка вбудови ЦВЗ у зображення завжди працює в двох областях: просторовій або області перетворень. Технології просторової області працюють безпосередньо з пікселями. Найбільш часто використовуваним методом просторової області є LSB. Методи області перетворення вставляють водяний знак шляхом зміни коефіцієнтів просторової області. Найбільш часто використовуваними методами з області перетворення є ДКТ, ДВТ і ДФТ. Для досягнення надійності та непомітності методи області перетворення є більш ефективними, ніж методи просторової області.

2.1 Методи просторової області

Просторова область представляє зображення у вигляді пікселів [4]. Вбудова ЦВЗ відбувається шляхом зміни інтенсивності та значення кольору деяких вибраних пікселів. Сильні сторони систем ЦВЗ просторової області:

- простота;
- низька обчислювальна складність;
- менші часові витрати.

Вбудова ЦВЗ методами просторової області простіше, при чому їх обчислювальна швидкість висока порівняно з методами з частотної області, але вони менш стійкі до атак. Методи просторової області можна легко застосувати до будь-якого зображення. Найважливішим методом просторової області є LSB (Least Significant Bit).

Метод найменший значущого біту (LSB) – це найпростіша техніка вбудови ЦВЗ в просторовій області, яка націлена на найменш значущі біти деяких вибраних пікселів зображення. Основна перевага цього методу полягає в тому, що він легко виконується на зображеннях. І це забезпечує

високу прозорість сприйняття. При вбудові ЦВЗ за допомогою LSB якість зображення не погіршується. Основним недоліком техніки LSB є її погана стійкість до звичайних операцій обробки сигналів, оскільки за допомогою цієї техніки водяний знак можна легко знищити будь-якими атаками. Цей метод не вразливий до атак і шуму, але дуже непомітний.

Аддитивні ЦВЗ. Найпростіший метод для вбудовування ЦВЗ в просторовій області – це додати псевдовипадковий шаблон шуму до інтенсивності пікселів зображення. Сигнал шуму зазвичай являє собою цілі числа (-1, 0, 1) або іноді числа з плаваючою комою. Щоб гарантувати, що водяний знак може бути виявлений, шум створюється ключем, так що кореляція між номерами різних ключів буде дуже низькою.

2.2 Методи частотної області

Порівняно з методами просторової області, методи частотної області застосовуються ширше. Метою є вбудовування ЦВЗ у спектральні коефіцієнти зображення. Найбільш часто використовуваними перетвореннями є дискретне косинусне перетворення (ДКП), дискретне перетворення Фур’є (ДПФ), дискретне вейвлетне перетворення (ДВП). Причина вбудови ЦВЗ у частотній області полягає в тому, що характеристики зорової системи людини (ЗСЛ) краще фіксуються спектральними коефіцієнтами.

2.2.1 Дискретне косинусне перетворення

ДКП, як і перетворення Фур’є, представляє дані в рамках частотного, а не амплітудного простору. Це корисно, оскільки це більше відповідає тому, як людина сприймає світло, так що іншу частину можна визначити та викинути. Методи нанесення цифрових водяних знаків на основі ДКП є надійними порівняно з методами просторової області. Такі алгоритми стійкі

до простих операцій обробки зображень, таких як фільтрація низьких частот, регулювання яскравості та контрастності, розмиття тощо. Однак їх важко реалізувати, а також вони є дорожчими з точки зору обчислень. У той же час, вони слабкі проти геометричних атак, таких як обертання, масштабування, обрізання тощо. Вбудову ЦВЗ в області ДКП можна класифікувати на глобальні ДКП і ті, що побудовані на основі блоків. Вбудовування в перцептивно значущу частину зображення має свої переваги, оскільки більшість схем стиснення видаляє незначну для сприйняття частину зображення. Кроки алгоритму ЦВЗ на основі блокового ДКП описані нижче.

Крок 1. Зображення розбивається на блоки, що не перекриваються.

Крок 2. Застосовується прямий ДКП до кожного з цих блоків.

Крок 3. Вибираються блоки за певним алгоритмом (наприклад, ЗСЛ).

Крок 4. Застосовуються критерії вибору коефіцієнта (наприклад, найвищий або найменший).

Крок 5. Відбувається вбудова ЦВЗ, змінивши вибрані коефіцієнти.

Крок 6. Застосовується зворотне перетворення ДКП до кожного блоку.

2.2.2 Дискретне вейвлетне перетворення

Вейвлет-перетворення – це сучасна техніка, яка часто використовується в цифровій обробці зображень, стиснення, цифрових водяних знаків тощо. Перетворення засновані на малих хвилях, які називаються вейвлетами, різної частоти та обмеженої тривалості. Вейвлет-перетворення розкладає зображення на три просторові напрямки: горизонтальний, вертикальний і діагональний. Отже, вейвлети більш точно відображають анізотропні властивості ЗСЛ. Величина коефіцієнтів ДВП більша в найнижчих діапазонах (LL) на кожному рівні розкладання і менша для інших діапазонів (HH, LH і HL). Піддіапазон LL1 представляє коефіцієнти ДВП більшого масштабу, тоді як піддіапазони LH1, HL1 і HH1 представляють дрібні коефіцієнт. Для отримання наступної більш грубої шкали вейвлет-

коефіцієнтів, піддіапазон LL1 додатково обробляється, поки не буде досягнуто остаточного масштабу N. Коли N буде досягнуто, буде отримано 3N+1 піддіапазонів, що складаються з піддіапазонів з великою роздільною здатністю LLN і LHx, HLx і HHx, де x коливається від 1 до N. Дискретне вейвлетне перетворення (ДВП) в даний час використовується в широкому спектрі програм обробки сигналів, таких як стиснення аудіо та відео, видалення шуму в аудіо та моделювання розподілу бездротової антени. Енергія вейвлетів зосереджена в часі і добре підходять для аналізу перехідних сигналів, що змінюються в часі. Оскільки більшість реальних сигналів, які зустрічаються в реальному житті, мають різний характер у часі, вейвлетне перетворення дуже добре підходить для багатьох застосувань. Однією з головних викликів проблеми ЦВЗ є досягнення кращого компромісу між нідійністю та непомітністю. Надійність може бути досягнута шляхом збільшення міцності вбудованого цифрового водяного знаку, але помітність перетворення також буде збільшено. Однак ДВП є більш доцільним, оскільки воно забезпечує як одночасну просторову локалізацію, так і частотне розповсюдження ЦВЗ всередині основного зображення. Основна ідея дискретного вейвлет-перетворення в процесі зображення полягає в розкладанні зображення на підзображення з різними просторовими областями та незалежними частотами.

Переваги ДВП перед ДКП. Вейвлет-перетворення краще працює з ЗСЛ, ніж ДКП. Закодоване зображення за допомогою вейвлетного перетворення є багатошаровим описом зображення. Таким чином, зображення можна відображати з різними рівнями роздільної здатності та послідовно обробляти від низької до високої роздільної здатності.

Недоліки ДВП перед ДКП. Обчислювальна складність ДВП більша в порівнянні з ДКП. Для обчислення ДКП для блоку 8x8 потрібно лише 54 множення, на відміну від вейвлет-розрахунку залежно від довжини використовуваного фільтра, який становить принаймні 1 множення на коефіцієнт.

2.2.3 Дискретне перетворення Фур’є

ДПФ перетворює безперервну функцію в її частотні компоненти. Воно стійке до геометричних атак, таких як обертання, масштабування, обрізання, перенос тощо. ДПФ показує незмінність переносу. Просторові зсуви в зображенні впливають на фазове представлення зображення, але не на відображення величини, або кругові зсуви в просторовій області не впливають на величину перетворення Фур’є.

2.2.4 Сингулярне розкладання значень

Сингулярне розкладання значень (СРЗ) є одним із найпотужніших інструментів чисельного аналізу, який використовується для аналізу матриць. Цей тип алгоритмів виявився надійним у системах водяних знаків [6]. У СРЗ-перетворенні матрицю можна розкласти на три матриці, які мають той самий розмір, що й вихідна матриця. СРЗ-перетворення зберігає як односторонні, так і несиметричні властивості, які зазвичай неможливо отримати в перетвореннях ДКТ і ДФТ. Використання СРЗ в цифровій обробці зображень має переваги, такі як розмір матриць від СРЗ-перетворення не є фіксованим і може бути квадратом або прямокутником; сингулярні значення в цифровому зображенні менш впливають, якщо виконується загальна обробка зображення, а одиничні значення містять внутрішні алгебраїчні властивості зображення. Поодинокі значення основного зображення змінюються для вбудовування ЦВЗ за допомогою кількох одиничних функцій. Водяний знак вбудовується та зчитується шляхом коригування значення між вибраними коефіцієнтами та фактичним результатом, отриманим регресією опорного вектора. Факторизація СРЗ виконується на різних блоках, що не перекриваються, шляхом вейвлет-перетворення. Водяні знаки генеруються одиничним значенням різного блоку.

2.3 Опис використаних технологій

Для програмної реалізації обраних методів було обрано мову програмування Python, оскільки вона є легкою у використанні та за допомогою неї можна реалізувати бажане максимально швидко. Це досягається в першу чергу завдяки готовим бібліотекам.

2.3.1 PyWavelets

PyWavelets – це відкрите програмне забезпечення для виконанння вейвлет-перетворень створене на Python. Він поєднує в собі простий інтерфейс високого рівня з продуктивністю низькорівневих програм написаних на мові програмування C. Ця бібліотека дуже проста у використанні, вона не потребує додаткових налаштувань – достатньо просто встановити пакет і одразу можна використати всі можливості цього модулю [7].

Основні можливості PyWavelets:

- N-вимірне пряме та зворотне дискретне вейвлет-перетворення (ДВП і зворотнє ДВП);
- N-вимірні багаторівневі ДВП і зворотнє ДВП;
- N-вимірне стаціонарне вейвлет-перетворення;
- розкладання та реконструкція 1D та 2D вейвлет-пакетів;
- одномірне неперервне вейвлет-перетворення;
- обчислення апроксимацій вейвлетів і масштабуючих функцій;
- понад 100 вбудованих вейвлет-фільтрів і підтримка користувацьких вейвлетів;
- розрахунки з використанням чисел одинарної та подвійної точності;
- реальні та комплексні розрахунки;
- результати сумісні з Matlab Wavelet Toolbox (TM).

2.3.2 Keras

Keras – це бібліотека, що пропонує ефективний високорівневий інтерфейс програмного програмування (API), написаний на Python, який націлений на комфорт й простоту роботи програміста. Keras дотримується найкращих практик щодо зниження когнітивного навантаження: він пропонує послідовні та прості API, мінімізує кількість дій користувача, необхідних для поширених випадків використання, і надає чіткі й ефективні повідомлення про помилки [8]. Для Keras створена змістовна й вичерпна документація, а також багато посібників для розробників.

Ця бібліотека для роботи з нейронними мережами має відкритий програмний код і є розробленою для швидкого експериментування з глибокими нейронними мережами. Вона може працювати поверх різних бекенідів CNTK, TensorFlow і Theano. Keras зосереджується на модульності, зручності і розширюваністі. Всі низькорівневі обчислення не обробляютья нею, замість цього вони передаються до іншої бібліотеки під назвою бекенд (Backend).

Keras була прийнята і інтегрована в TensorFlow в середині 2017 року. Вона є центральною частиною тісно пов’язаної екосистеми TensorFlow 2, яка охоплює кожен крок робочого процесу машинного навчання, від керування даними до навчання гіперпараметрів і рішень для розгортання. Користувачі можуть отримати доступ до інтерфейсу Keras через модуль tf.keras. Проте ця бібліотека все ще може працювати окремо й незалежно радше як інтерфейс, аніж як самостійна система машинного навчання.

Keras містить численні втілення широко вживаних будівельних блоків для нейронних мереж, таких як шари, цільові та передавальні функції, оптимізувальники та безліч інструментів для спрощення роботи із зображеннями та текстом для спрощення написання коду глибинних нейромереж [9].

На додаток до стандартних нейронних мереж, Keras підтримує

згорткові та рекурентні нейронні мережі, а також загальні утиліти, такі як вилучення, нормалізація пакетів і об'єднання.

Keras дозволяє користувачам створювати глибокі моделі на смартфонах, в Інтернеті або на віртуальній машині Java. Це також дозволяє використовувати розподілене навчання моделей глибокого навчання на кластерах графічних процесорів (GPU) і тензорних процесорів (TPU).

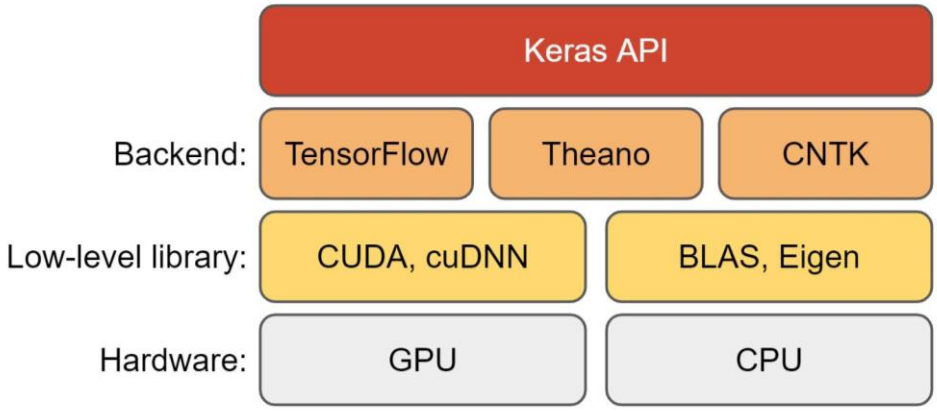

Рисунок 2.3 – Відношення Keras та інших бібліотек для машинного навчання

2.3.3 OpenCV

OpenCV (Open Source Computer Vision Library) – це програмний інтерфейс (API), створений компанією Intel, що може бути використаний у багатьох програмах обробки зображень і комп’ютерного зору. OpenCV було офіційно запущено в 1999 році, і спочатку цей проект був ініціативою Intel Research для просування додатків із інтенсивним процесором, частиною серії проектів, включаючи трасування променів у реальному часі та 3D-дисплеї.

Бібліотека OpenCV – це набір алгоритмів і функцій, а також класів, які реалізують деякі алгоритми обробки зображень і комп’ютерного зору. OpenCV був розроблений для обчислювальної ефективності та з сильним акцентом на додатках реального часу. OpenCV написаний на C з великою кількістю оптимізацій і може використовувати переваги багатоядерних

процесорів.

OpenCV містить понад 500 функцій, які охоплюють багато областей, включаючи заводську перевірку продукції, медичне зображення, безпеку, інтерфейс користувача, калібрування камери, стереобачення та робототехніку. Принципи створення бібліотеки полягають у тому, щоб сприяти комерційному використанню комп'ютерного зору в інтерфейсі «людина-комп'ютер», робототехніці, моніторингу, біометрії та безпеці, забезпечуючи безкоштовну та відкриту інфраструктуру. Підтримка OpenCV включає звичайну підтримку для введення, відображення та зберігання фільмів та окремих зображень. OpenCV використовує DirectX, який являє собою набір API, розроблений Microsoft для створення мультимедійних програм та ігор.

Бібліотека OpenCV є кроссплатформною і працює як в операційних системах Windows, так і в Linux. OpenCV швидко набирає популярності для розробки програм комп'ютерного зору в режимі реального часу. Функціонал бібліотеки включає засоби розпізнавання облич, розпізнавання об'єктів і трекери руху, і це лише деякі з них. Бібліотека особливо набула популярності в дослідницькому середовищі комп'ютерного зору. Це дозволяє дослідникам швидко запускати демонстраційні або дослідницькі проекти, а також користуватися великою колекцією уже реалізованих алгоритмів. Базові компоненти бібліотеки достатньо повні, щоб створити повне рішення практично будь-якої проблеми комп'ютерного зору.

Обробка зображень належить до низькорівневих процесів, які включають примітивні операції, такі як попередня обробка зображення для зменшення шуму. Дані обробки зображень низького рівня складаються з оригінальних зображень, представлених матрицями, що складаються із значень яскравості. Методи низького рівня зазвичай використовують дуже мало знань про зміст зображень. Низький рівень характеризується тим, що на його входах і виходах є зображення. Послідовність обробки зображень низького рівня передбачала попередню обробку та сегментацію зображення.

З іншого боку, комп’ютерний зір – це високорівнева обробка. Обробка високого рівня включає в себе розуміння або «осмислення» групи розпізнаних об'єктів, аналіз зображень і, нарешті, виконання когнітивних функцій, пов'язаних з людським баченням. Комп’ютерний зір високого рівня намагається імітувати людське сприйняття і його здатність приймати рішення відповідно до інформації, що міститься на зображенні. Наприклад, розмір, форма та взаємозв’язки між об’єктами на зображенні. Дані високого рівня зазвичай виражаються в символічній формі.

OpenCV підтримує наступні основні різновиди функцій:

- загальні алгоритми комп'ютерного зору та обробки зображень (API середнього та низького рівня);
- модулі комп’ютерного зору високого рівня;
- штучний інтелект та методи машинного навчання;
- методи створення та аналізу бінарних зображень;
- математичні процедури для обробки зображень, комп’ютерного зору та інтерпретації зображень.

2.4 Згорткові нейронні мережі

Згорткова нейронна мережа (ConvNet/CNN) – це алгоритм глибокого навчання, який може приймати вхідне зображення, призначати важливість (визначені ваги) різним аспектам/об’єктам зображення та мати можливість відрізнити їх [10]. Попередня обробка, необхідна для CNN, набагато нижча в порівнянні з іншими алгоритмами класифікації. Хоч в примітивних методах фільтри розробляються вручну, при достатньому підготовці CNN мають можливість вивчати ці фільтри/характеристики.

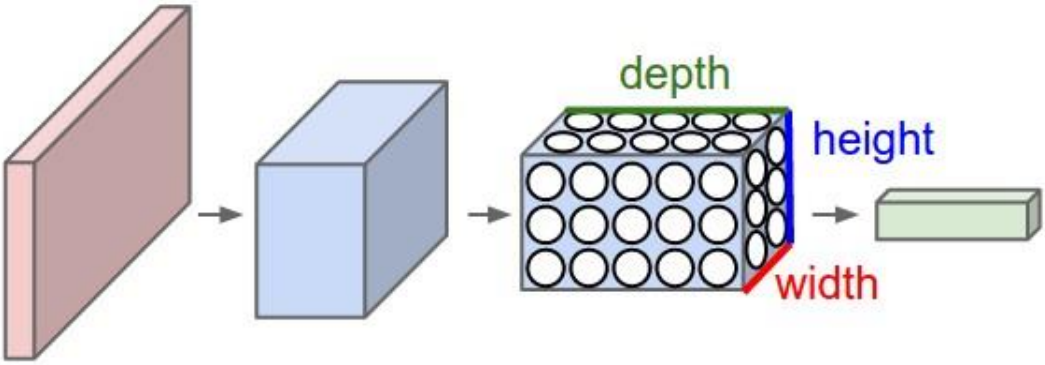

Рисунок 2.4 – Згоркова нейронна мережа

Архітектура CNN аналогічна структурі зв’язку нейронів у людському мозку і була натхненна організацією зорової кори. Окремі нейрони реагують на подразники лише в обмеженій області поля зору, відомому як рецептивне поле. Набір таких полів перекривається, щоб охопити всю візуальну область.

Згорткові нейронні мережі відрізняються від інших нейронних мереж їх продуктивністю й ефективністю при роботі з зображенням, мовленням або звуковим сигналом. Роль CNN полягає в тому, щоб зменшити зображення до форми, яку легше обробляти, без втрати особливостей, які є критичними для отримання гарного прогнозу.

Згортковий шар є першим шаром згорткової мережі. З кожним шаром CNN підвищується складність, ідентифікуючи більші частини зображення. Ранні шари зосереджені на простих елементах, таких як кольори та краї. Коли дані зображення просуваються через шари CNN, вона починає розпізнавати більші елементи або форми об’єкта, поки нарешті не ідентифікує передбачуваний об’єкт.

Згортковий шар є основним будівельним блоком CNN, і саме в ньому відбувається більшість обчислень. Цей шай потребує кілька компонентів: вхідні дані, фільтри та карта ознак. Припустимо, що вхідним буде кольорове зображення, яке складається з матриці пікселів у 3D. Це означає, що вхідні дані будуть мати три виміри – висоту, ширину та глибину – які відповідають RGB зображенню. Також є детектор ознак, відомий як ядро або фільтр, який

буде переміщатися по сприйнятливих полях зображення, перевіряючи чи є певна особливість чи деталь. Цей процес відомий як згортка.

Детектор ознак – це двовимірний масив вагових коефіцієнтів, який представляє частину зображення. Хоча вони можуть відрізнятися за розміром, розмір фільтра зазвичай є матрицею 3x3, вона також визначає розмір рецептивного поля. Потім фільтр застосовується до області зображення, і між вхідними пікселями та фільтром обчислюється скалярний добуток. Цей добуток потім подається у вихідний масив. Після цього фільтр зміщується на один крок, повторюючи процес, поки ядро не охопить усе зображення. При чому ваги в детекторі функцій залишаються фіксованими, коли він рухається по зображенню. Результатом проходу фільтром по всьому зображенню є карта ознак, карта активації або згорнута ознака [11].

Деякі параметри ядра, такі як значення вагових коефіцієнтів, коригуються під час навчання за допомогою процесу зворотного поширення та градієнтного спуску. Проте є три гіперпараметри, які впливають на розмір виводу, який необхідно встановити перед початком навчання нейронної мережі. До них належать:

- кількість фільтрів, що впливає на глибину виводу. Наприклад, три різні фільтри дадуть три різні карти об’єктів, створюючи глибину рівній трьом;
- крок – це відстань або кількість пікселів, на які ядро переміщається по вхідній матриці. Хоча значення кроку два або більше зустрічається рідко, більший крок дає менший результат;
- заповнення, що зазвичай використовується, коли фільтри не мають змінювати розміри вхідного зображення. Ця дія обнуляє всі елементи, що виходять за межі фільтра, створюючи більший або однаковий за розміром вихідний тензор.

Після кожної операції згортки CNN застосовує перетворення Rectified Linear Unit (ReLU) до карти ознак, вносячи нелінійність у модель. Шари згортки можуть використатися кілька разів поспіль. Коли так відбувається,

структура CNN може стати ієрархічною, оскільки пізні рівні можуть обробляти пікселі в сприйнятливих полях попередніх шарів. Як приклад, припустимо, що ми намагаємося визначити, чи містить зображення велосипед. Люди сприймають велосипед як суму частин, він складається з рами, керма, коліс, педалей тощо. Кожна окрема частина велосипеда складає шаблон нижнього рівня в нейронній мережі, а комбінація його частин представляє шаблон вищого рівня, створюючи ієрархію функцій у CNN.

Згорткові нейронні мережі тепер забезпечують більш масштабований підхід до завдань класифікації зображень і розпізнавання об’єктів, використовуючи принципи лінійної алгебри, зокрема множення матриць, для виявлення шаблонів у зображенні. Тим не менш, вони можуть бути вимогливими до обчислень, вимагаючи графічних процесорів (GPU) для навчання моделей.

3 СИСТЕМИ ДЛЯ РОБОТИ З ЦИФРОВИМИ ВОДЯНИМИ ЗНАКАМИ

Нехай O позначає оригінальну текстуру, що має три канали кольору (R, G, B) і відповідно 24 біта на піксель, яка використовується для вбудовування водяного знака. Оригінальна текстура зазвичай представляється у вигляді двовимірного масиву і тому її можна представити у вигляді матриці [12]:

$$O = \begin{bmatrix} o(0,0) & o(0,1) & \dots & o(0,O_w-1) \\ o(1,0) & o(1,1) & \dots & o(1,O_w-1) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & o(i,j) & \dots \\ \dots & \dots & \dots & \dots \\ o(O_h-1,0) & o(O_h-1,1) & \dots & o(O_h-1,O_w-1) \end{bmatrix} \quad (3.1)$$

де $o(i,j)$ – ціле число, $0 \le o(i,j) \le 255$, $0 \le i \le O_h$, $0 \le j \le O_w$. O_h і O_w – висота і ширина вхідної текстури відповідно.

У розроблених системах вбудови ЦВЗ використано цифровий водяний знак W, який являє собою зображення. Це зображення має бути значущим доказом того, хто володіє авторськими правами на оригінальну текстуру [13]. Загалом, «торговельна марка» підприємства може бути обрана як водяний знак. Водяний знак визначається наступним чином:

$$W = \begin{bmatrix} w(0,0) & w(0,1) & \dots & w(0,W_w-1) \\ w(1,0) & w(1,1) & \dots & w(1,W_w-1) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & w(i,j) & \dots \\ \dots & \dots & \dots & \dots \\ w(W_h-1,0) & w(W_h-1,1) & \dots & w(W_h-1,W_w-1) \end{bmatrix} \quad (3.2)$$

де $w(i, j)$ – ціле число, $0 \le w(i, j) \le 255$, $0 \le i \le W_h$, $0 \le j \le W_w$. W_h і W_w – це висота і ширина водяного знака відповідно.

Тому мету створених систем вбудовування ЦВЗ можна визначити так: створити з текстури O стегоконтейнер для даних B таким чином з мінімально помітними змінами в текстурі і можливістю точного відновлення B зі стегоконтейнеру. Результати сворених систем будуть порівняні для знаходження оптимального підходу до вбудови ЦВЗ у текстури тривимірних об’єктів.

До реалізації й подальшого порівняння було обрано два способи вбудови водяного знаку у текстуру тривимірного об’єкту: детермінований алгоритм, що базується на використанні комбінації ДВП та ДКП, та нейронні мережі. Кожен з цих способів має свої переваги та обмеження, які будуть порівняні.

3.1 Опис детермінованого алгоритму

Алгоритм на основі ДВП та ДКП був обраний оскільки саме ці алгоритми було обрано оскільки вони мають певні властивості, які будуть корисними при вбудові ЦВЗ.

Завдяки чудовим властивостям просторово-частотної локалізації ДВП гарно підходить для визначення ділянок на основній текстурі, куди можна ефективно вставити водяний знак. Зокрема, ця властивість дозволяє використовувати маскувальний ефект зорової системи людини таким чином, що якщо коефіцієнт ДВП буде змінено, буде змінена лише область, що відповідає цьому коефіцієнту. Для створюваного алгоритму було вирішено використати коеффіцієнти LL2.

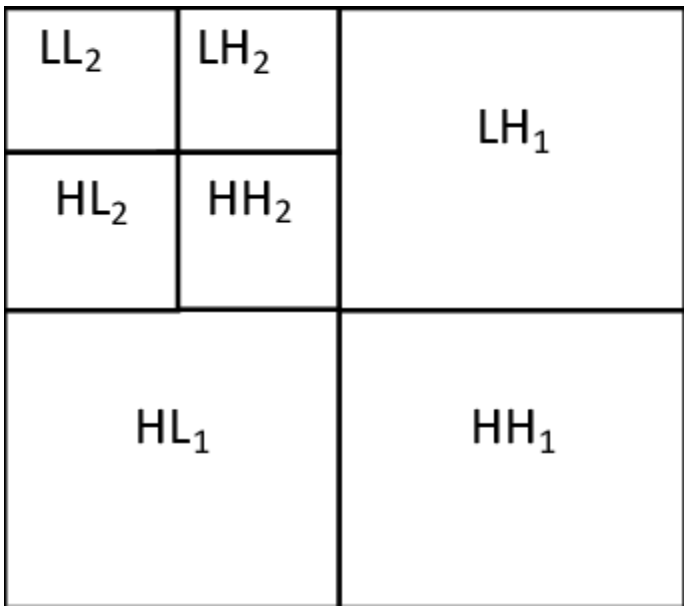

Рисунок 3.1 – Графічне зображення коефіцієнтів вейвлетного розкладу

ДКП також дозволяє розкласти зображення на декілька піддіапазонів, і коли водяний знак вбудовується шляхом модифікації коефіцієнтів середньочастотного піддіапазону, він матиме мінімальний вплив на вигляд текстури й допоможе уникнути видалення ЦВЗ шляхом стиснення. Це пов’язано з тим, що значна частина енергії сигналу лежить на низькочастотному піддіапазоні, який містить найважливіші візуальні частини зображення. Високочастотні компоненти текстури ж зазвичай видаляються під час стиснення та шумових атак [14].

3.1.1 Алгоритм вбудови водяного знаку

Розроблений алгоритм можна розділити на 9 кроків.

Крок 1. Перетворити водяний знак у вектор бітів, які в сукупності представляють собою дані ЦВЗ.

Крок 2. Розподілити текстуру на окремі канали кольору (R, G, B)

Крок 3. До кожного каналу кольору застосовувати ДВП для розподілу його на 4 піддіапазони, що не перекриваються: LL1, HL1, LH1 і HH1

Крок 4. До отриманого діапазону LL1 ДВТ застосовувати повторно для отримання чотирьох менших піддіапазонів (LL2, HL2, LH2 і HH2), з яких повторно вбирається піддіапазон LL2.

Крок 5. Розкласти вибранний піддіапазон LL2 на блоки 4 x 4.

Крок 6. До кожного блоку у вибраному піддіапазоні застосовувати ДКП.

Крок 7. В другий елемент третього рядку вбудовується значення біту з векторного представлення водяного знаку. Нове значення елемента буде визначено за наступною формулою 3.3.

Крок 8. До кожного блоку після того, як його коефіцієнти середнього діапазону були змінені, щоб вставити біти водяного знака, застосовується зворотне ДКП (ЗДКП).

Крок 9. До отриманої текстури застосовується зворотний ДВП (ЗДВП), включаючи змінений піддіапазон, щоб створити текстуру-контейнер з водяним знаком.

Вбудова даних ЦВЗ у окремі блоки має надавати додаткової стійкості цьому алгоритму, адже при втраті чи зміні частини зображення всі дані ЦВЗ не будуть втрачені. Тому хоча б частина прихованих даних буде зчитана.

Лістниг 3.1 – вбудова даних ЦВЗ у блоки текстури

```
def encode_frame(self, frame, wm, scale):
    (rows, cols) = frame.shape
    block_rows = rows//self._block
    block_cols = cols//self._block

    wm_bits = DwtDct.__to_bits(wm.data)

    for i in range(block_rows):
        for j in range(block_cols):
            block_i = i*self._block
            block_j = j*self._block
            block = frame[block_i : block_i + self._block,
                          block_j : block_j + self._block]
            wm_bit = wm_bits[(self._wm_idx % wm.length)]
            self._wm_idx += 1
```

Продовження Лістнигу 3.1

```
            diffused_block = DwtDct.__diffuse_dct_matrix(block,
self._block, wm_bit, scale)
            frame[block_i : block_i + self._block,
                  block_j : block_j + self._block] =
diffused_block
```

3.1.2 Алгоритм зчитування водяного знаку

Процесс зчитування даних ЦВЗ дуже схожий на алгоритм вбудови. Текстура піддається необхідним перетворенням, але тепер дані читаються з обраних комірок. Нижче приведені кроки для зчитування.

Крок 1. Текстура-контейнер розподіляється на канали кольору.

Крок 2. До кожного каналу кольору застосовується ДВТ для розподілу його на 4 піддіапазони LL1, HL1, LH1 і HH1

Крок 3. До отриманого діапазону LL1 ДВТ застосовується повторно для отримання чотирьох менших піддіапазонів (LL2, HL2, LH2 і HH2), з яких повторно вбирається піддіапазон LL2.

Крок 4. Вибранний піддіапазон LL2 розбивається на блоки 4 x 4.

Крок 5. До кожного блоку у вибраному піддіапазоні застосовується ДКТ.

Крок 6. У кожному блоці після перетворення вибирається другий елемент третього ряду. З нього за допомогою формули 3.4 вираховується значення біта закодованого водяного знаку й додається до результуючого вектору.

Крок 7. Після обробки всіх блоків в кожному каналі текстури отриманий вектор перетворюється у зображення, яке і є водяним знаком.

Значення бітів записуються відповідно формулам, оскільки алгоритму треба мати якийсь спосіб розуміти яке значення біта треба додати до результуючого вектора. Тому використаються наступні формули:

$$o(2,1)' = o(2,1) \setminus 20 + (0.25 + 0.5 \cdot w(i)) \cdot 20 \tag{3.3}$$

$$w(i) = \begin{cases} 1, \; if \;\; o(2,1) \setminus 20 \geq 20 / 2 \\ 0, \; if \;\; o(2,1) \setminus 20 < 20 / 2 \end{cases} \tag{3.4}$$

де $w(i)$ – значення біту водяного знаку, 20 – ваговий коефіцієнт обраний емпірично, $o(2,1)$ – значення пікселя текстури.

3.2 Опис розробленої нейронної мережі

Для порівняння було створено нейронну мережу, яка також виконує вбудову та зчитування ЦВЗ з текстур. Таку нейронну мережу можна віднести до класу автоенкодерів, де текстура й водяний знак є вхідними даними, а закодована текстура-стегоконтейнер – це вектор прихованого стану. Розроблену нейронну мережу можна розділити на декілька окремих логічних блоків: кодер, декодер, обчислювач помилки.

Архітектура отриманої нейронної мережі була отримана в результаті експериментальних перевірок різних структур, де найбільш ефективно себе показала архітектра нейронної мережі з великою кількостю згорткових слоїв. Вони дуже широко використовуються в обробці зображень, а тому відповідно можуть бути використані для обробки текстур. Архітектуру отриманої нейронної мережі наведено у ДОДАТКУ Б.

Кодер відповідає за вбудовування водяного знаку у текстуру. Входами кодера є сама текстура, на основі якої має бути створений стегоконтейнер, а також водяний знак представлений також у вигляді зображення.

Водяний знак інтегрується в зображення після пропускання через кілька згорткових слоїв. Цей модуль мережа окремо оброблює всі канали кольору текстури і в кожний з каналів записуються дані водяного знаку. При чому для більшої непомітності вбудованого ЦВЗ, дані інтегруються в оригінальне трьома паралельними слоями, які мають різну кількість згорткових слоїв і потім конкатенуються між собою. Цей модуль інтеграції зображений на рисунку 3.2.

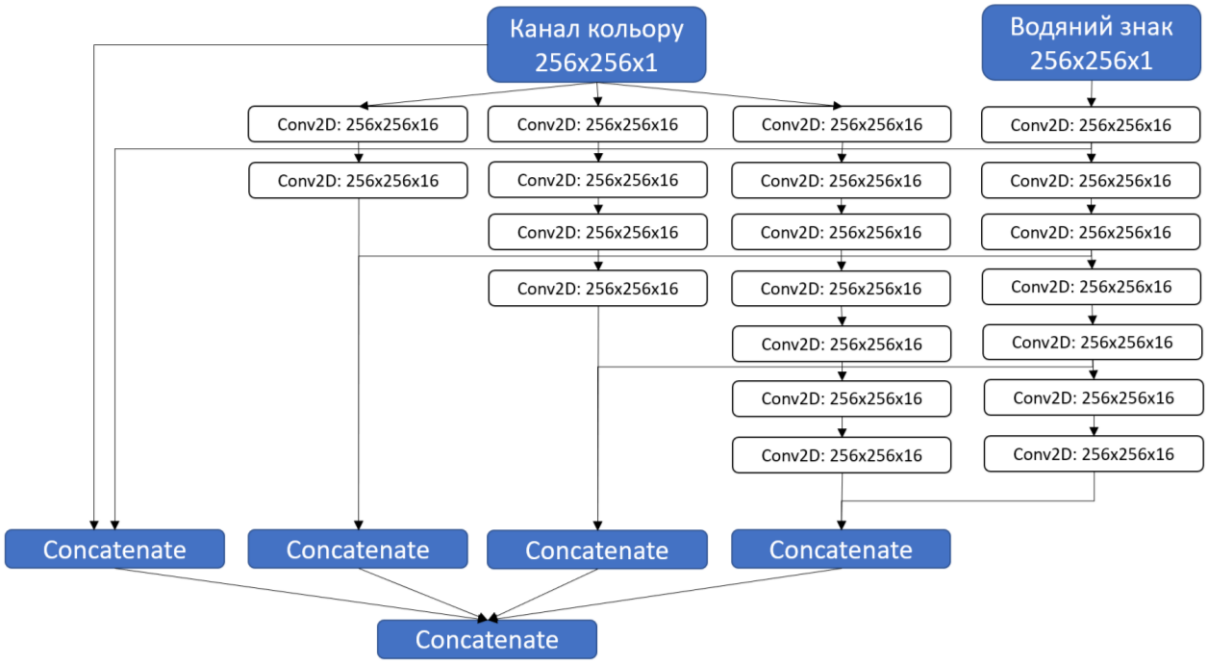

Рисунок 3.2 – Модуль інтеграції водяного знаку в канал кольору текстури

При чому, що згорка реалізована не одним шаром і являє собою окремий будівельний блок для нейронної мержі. Вона вимагає додаткової нормалізації й використання функції активації, що показано на рисунку 3.3. У якості функції активації було обрано ReLU.

Лістинг 3.2 – Реалізація модуля згортки на Keras

```
def conv2D(X, num_filters, filter_size,
           strides=1, padding="same", activation=True,
           kernel_initializer=None, bias_initializer=None):

    X = Conv2D(num_filters, filter_size,
               strides=strides, padding=padding,
               kernel_initializer=kernel_initializer,
               bias_initializer=bias_initializer)(X)
    X = BatchNormalization(axis=3, scale=False)(X)
    if activation:
        X = Activation("relu")(X)
```

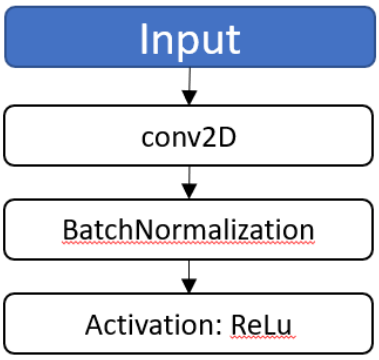

Рисунок 3.3 – Шари використані для згортки

Після блоку інтеграції тензори даний передаються блокам «занурення» або ж інсепшн блоками (Inception), які проводять додаткову обробку наобхідну для більш непомітної вбудови водяного знаку в текстуру. При чому ці блоки використаються з різними кількостями фільтрів у згорткових слоїв. Схему даного блоку приведено на рисунку 3.4.

Інсепшн модулі використовуються в згорткових нейронних мережах для забезпеченн більш ефективних обчислення та глибших мереж за рахунок зменшення розмірності, які досягаються за допомогою згорток. Ці модулі були розроблені для вирішення проблеми обчислювальних витрат, а також перенавчання. Рішення полягає в тому, щоб взяти кілька розмірів фільтрів ядра в CNN і замість того, щоб з'єднувати їх послідовно, розмістити їх паралельно щоб вони працювали на одному рівні [15]. Мета полягає в тому, щоб нейронна мережа засвоїла найкращі ваги під час навчання та автоматично вибрала корисніші ознаки в текстурі.

Інсепшн модулі вбудовуються в CNN як спосіб скорочення обчислювальних витрат. Оскільки нейронна мережа має справу з величезним набором текстур із широкими варіаціями помітних частин, їх потрібно спроектувати належним чином. Найбільш спрощена версія початкового модуля працює, виконуючи згортку на вході з не одним, а трьома різними розмірами фільтрів (1x1, 3x3, 5x5). Також виконується макс-пулінг (MaxPooling). Потім отримані вихідні дані об’єднуються і надсилаються на

наступний шар. Структуруючи CNN, щоб виконувати згортки на одному рівні, мережа ставатиме ширшою, а не глибшою [16].

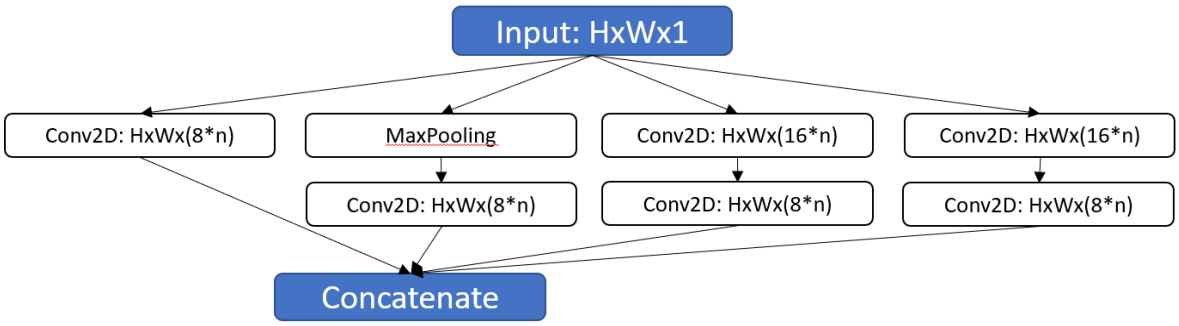

Рисунок 3.4 – Структура інсепшн модуля

Експериментально було з’ясовано, що для ефективного функціонування створеної системи потрібно шість таких блоків. І в результаті було отримано структуру кодера, яку зображено на рисунку 3.5.

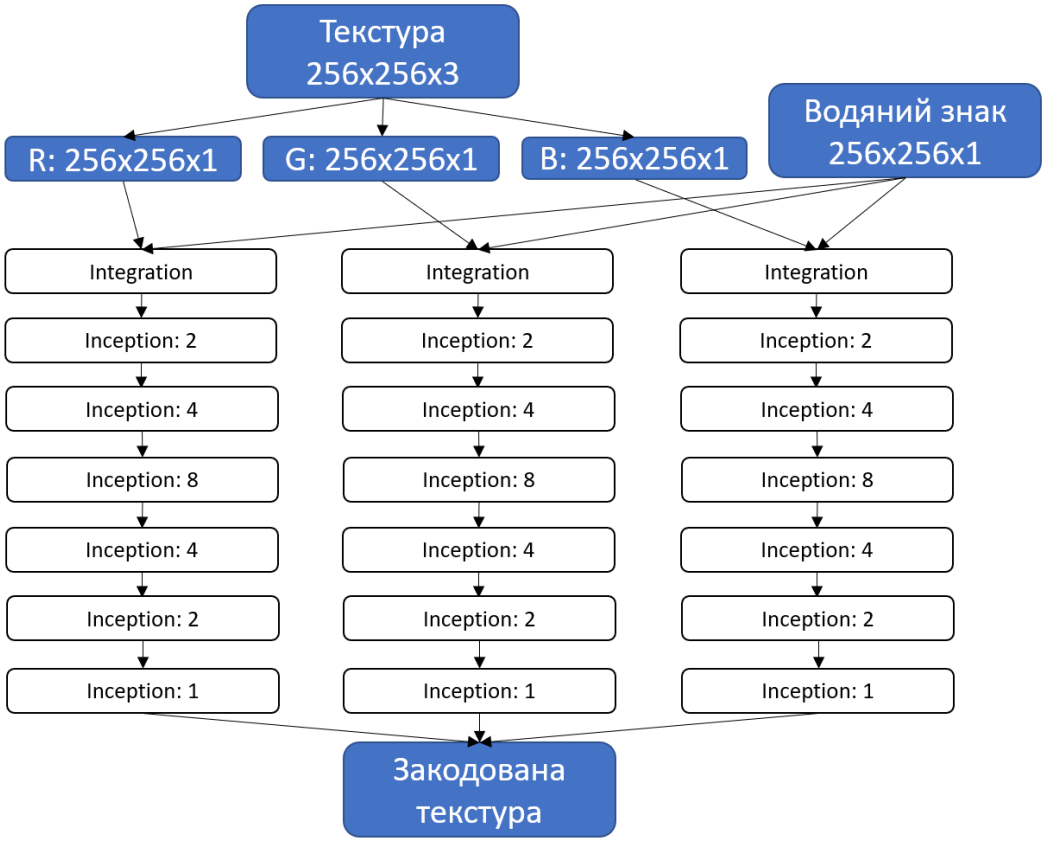

Рисунок 3.5 – Архітектура кодера

Декодер. Мережа декодера відповідає за зчитування водяного знаку з текстури-контейнера. Вона приймає на вхід текстуру визначеного розміру та зчитує вбудовані в неї дані. За своєю структурою декодер значно простіший і складається лише з п’яти згорткових слоїв для кожного каналу кольору, які тут також обробляються окремо. В даному випадку використаються ті ж самі згортвокі слої що і в кодері. Архітектура декодера приведена на рисунку 3.6.

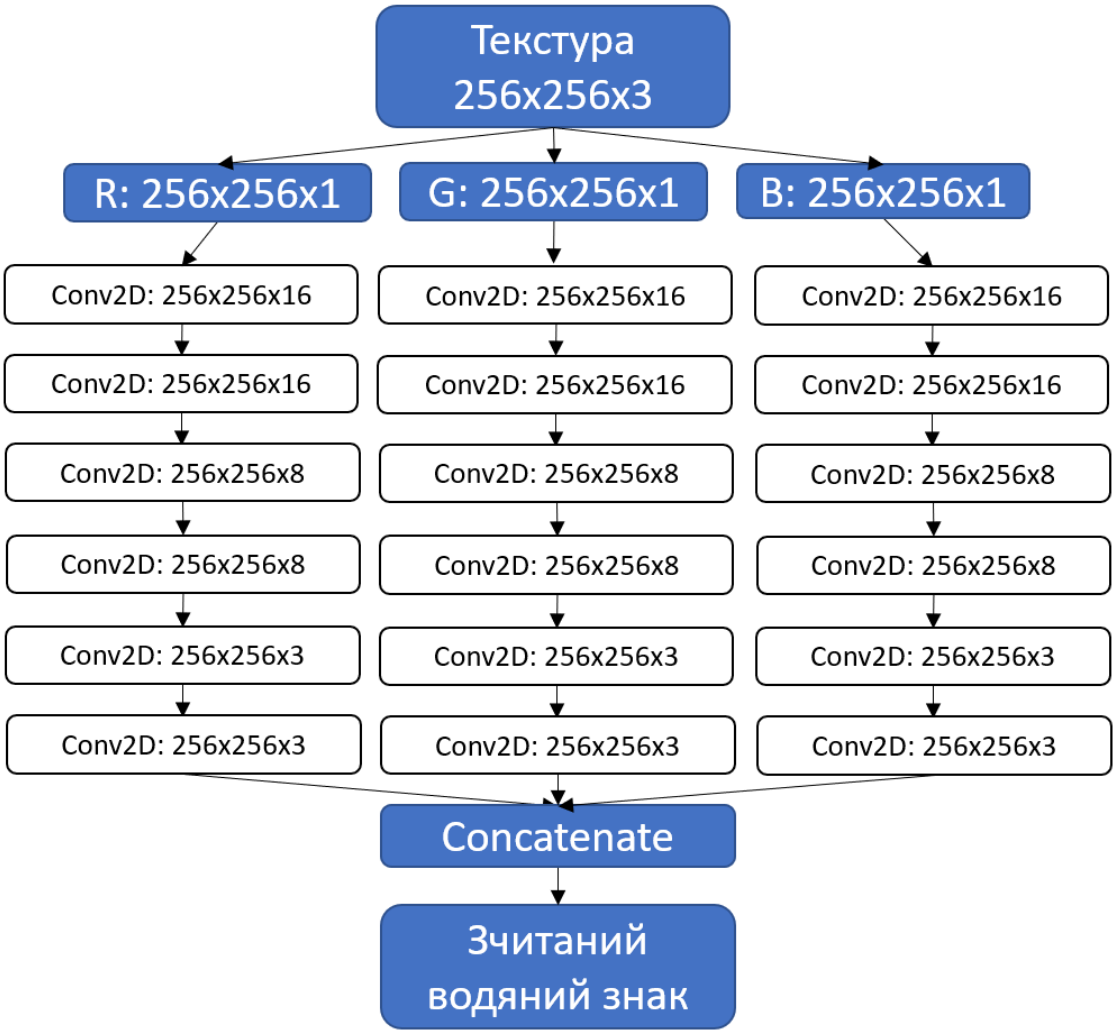

Рисунок 3.6 – Архітектура декодера

Навчання нейронної мережі відбувалося для досягнення мінімальної квадратичної помилки між каналами кольору оригінальної текстури та результуючої текстури-контейнера, а також між оригінальним водяним знаком і даними, зчитаними під час декодування контейнеру. Таким чином

нейронна мережа навчалась вбудовувати водний знак з мінімальними змінами до текстури і при цьому дані водяного знаку досить точно зчитувались.

У ході навчання використався оптимізатор Adam. Adam – це метод ефективної стохастичної оптимізації, який вимагає лише градієнтів першого порядку з невеликою потребою в пам’яті. Метод обчислює індивідуальні швидкості адаптивного навчання для різних параметрів на основі оцінок першого та другого моментів градієнтів. Цей метод розроблено для об’єднання переваг двох інших популярних методів: AdaGrad, який добре працює з розрідженими градієнтами, і RMSProp [17]. Емпіричні результати демонструють, що Adam добре працює на практиці і вигідно порівнюється з іншими методами стохастичної оптимізації. На даний момент цей алгоритм є рекомендованим за замовчуванням, бо зазвичай він є ефективнішим за інші алгоритми [18].

3.3 Вимоги до текстур та водяних знаків

Створені засоби для роботи з ЦВЗ мають певні обмеження. На данному етапі нейронною мережую підтримуються лише зображення розміром 256x256. Для роботи з текстурами інших розмірностей мережу потрібно додатково навчати і створювати нові датасети із зображеннями відподвідного розміру. У якості ЦВЗ можуть використатися лише зображення 256x256 пікселів в градаціях сірого, тобто лише з одним каналом кольору. А отже ця система може бути вдосконалена для підтримки текстур та ЦВЗ різномантіного розміру.

Обраний детермінований алгоритм підтримує текстури розміром від 256x256. Таке обмеження пов'язане з тим, що лише так буде достатньо даних для непомінтної вбудови ЦВЗ. Максимальний розмір водяного знаку для цього алгоритму становить 18x18 пікселів по 8 бітів на піксель, що сумарно становить 324 байти. Такі обмеження пов'язані з особливостями обраного

алгоритму: він не так ефективно знаходить ділянки текстури, які будуть менш чутливими до змін. Тому вбудова великої кількості даних призведе до понівечення зображення.

З точки зору предметної області таке обмеження на розміри вхідних текстур є допустимим, адже зазвичай вони мають розміри кратні степеням двійки. Так сталося, бо більшість алгоритмів компресії текстур мають такі обмеження, а алгоритми згдажування дають кращий результат з текстурами, чиї розміри кратні двійці. Компресія текстур необхідна для оптимізації операцій завантаження/вигрузки текстур з пам'яті графічного процессора, саме тому використовуються такі алгоритми як ETC [19], PVRTC [20].

Взагалі у якості водяного знаку можуть використатися будь-які дані, що можуть бути представлені у вигляді одномірного масиву (для детермінованого алгоритму) та двомірного – для нейронної мережі. В ході дослідження ЦВЗ представлені зображеннями, оскільки так можна візуально оцінювати зміни водяного знаку після зчитування.

4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Розглянемо приклади роботи розроблених систем для роботи з ЦВЗ на досить простій текстурі цегли. На рисунку 4.1 приведено оригінальний вигляд текстури і після вбудови ЦВЗ за допомогою детермінованого алгоритму.

Рисунок 4.1 – Вхідна текстури та отримана в результаті вбудови водяного знаку детермінованим алгоритмом

Текстура зберегла свій вигляд. Зміни є візуально непомітними. Вставка водяного знаку ніяким чином не погіршила якість текстури. При чому водяний знак гарантовано буде відновлено з цього стегоконтейнеру. На рисунку 4.2 дуже гарно видно, що оригінальний водяний і той, що був прочитаний з текстури, неможливо відрізнити.

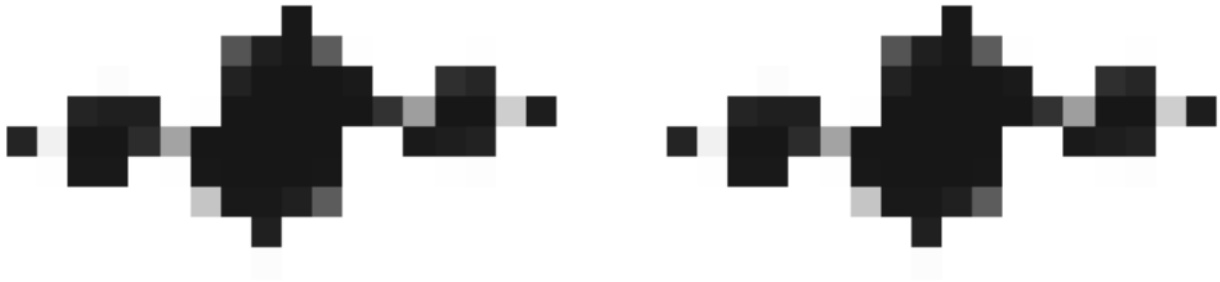

Рисунок 4.2 – Оригінальний водяний знак та зчитаний детермінованим алгоритмом

Відповідно розглянемо аналогічні дії з системою реалізованою за допомогою нейронної мереж, зображених на рисунку 4.3. При використанні цього методу текстура також змінюється мінімально: трохи знижується контрастність та підвищується яскравість.

Рисунку 4.3 – Вхідна текстури та отримана в результаті вбудови водяного знаку нейронною мережею

Водяний знак, зчитаний даною системою, не повністю співпадає з тим, який був вбудований. Є розбіжності в тонах: білий колір став яскраво-сірим, а чорний колір – темно-сірим.

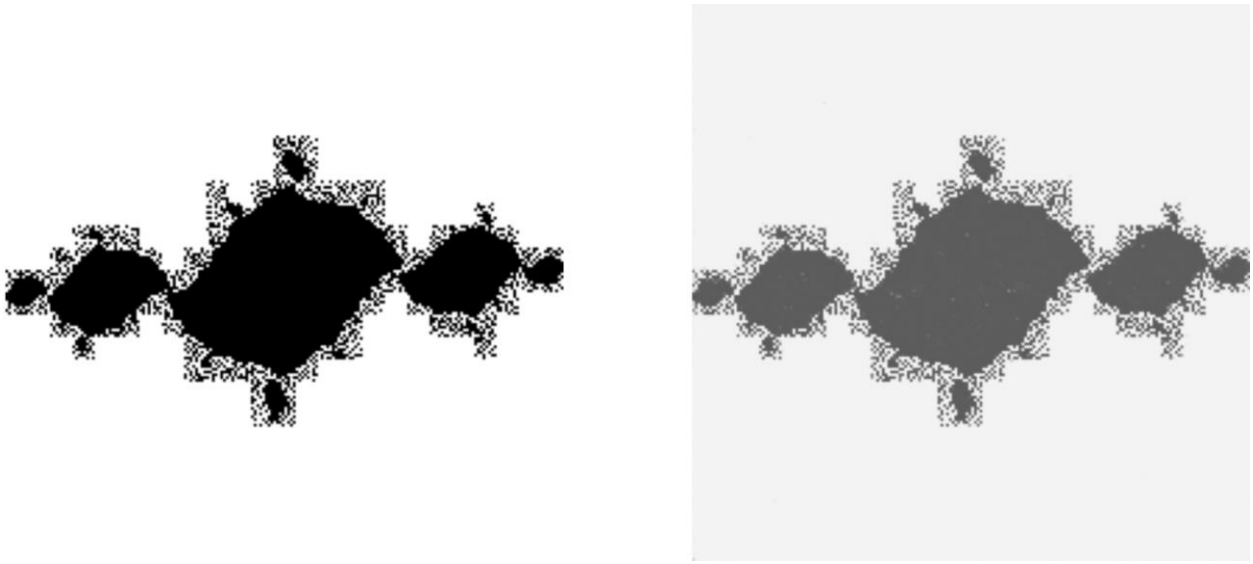

Рисунок 4.4 – Оригінальний водяний знак та зчитаний нейронною мережею

З цього можна зробити висновок, що розроблені системи для роботи з ЦВЗ успішно виконують свої функції. Вони успішно вбудовують дані у текстури, а потім зчитують з них ці приховані дані. Без стороннього впливу вставка ЦВЗ за допомогою детермінованого алгоритму проявляє себе краще, бо зчитаний ЦВЗ повністю співпадає з вхідним. В аналогічних умовах нейронна мережа поводить себе гірше, бо є візульно помітні зміни й середньковадратична помилка становить 0.02.

Але виконання базових функцій звісно недостатньо. Особливу увагу треба приділити надійності вбудованих водяних знаків. Стегоконтейнери зазвичай піддаються зовнішній обробці, яка може понівечити дані ЦВЗ чи повністю їх знищи.

В ході експерименту досліджувався вплив наступних типів атак:

- розмивання заданим ядром;
- додавання нормально розподіленого шуму із заданим середнім

значенням і дисперсією (Гауссівський шум);

- додавання шуму типу «сіль та перець» із заданою щільністю;
- зміна яскравості;
- стиснення JPEG із заданим параметром якості.

До текстури-контейнера застосовувалися всі ці атаки і після чого проводилося зчитування водяного знаку і прораховувалася середньо-квадратична помилка між оригінальним водяним знаком і зчитаним.

У ході дослідження впливу розмивання використалися фільтри з різними розмірностями ядр. Отримані значення помилки представлені у таблиці 4.1.

Таблиця 4.1 – Отримані помилки після атаки розмиванням

Ядро	2x2	3x3	4x4	5x5	6x6	7x7
НМ	0.171	0.171	0.172	0.170	0.170	0.170
ДВП-ДКП	0.145	0.218	0.302	0.317	0.343	0.357

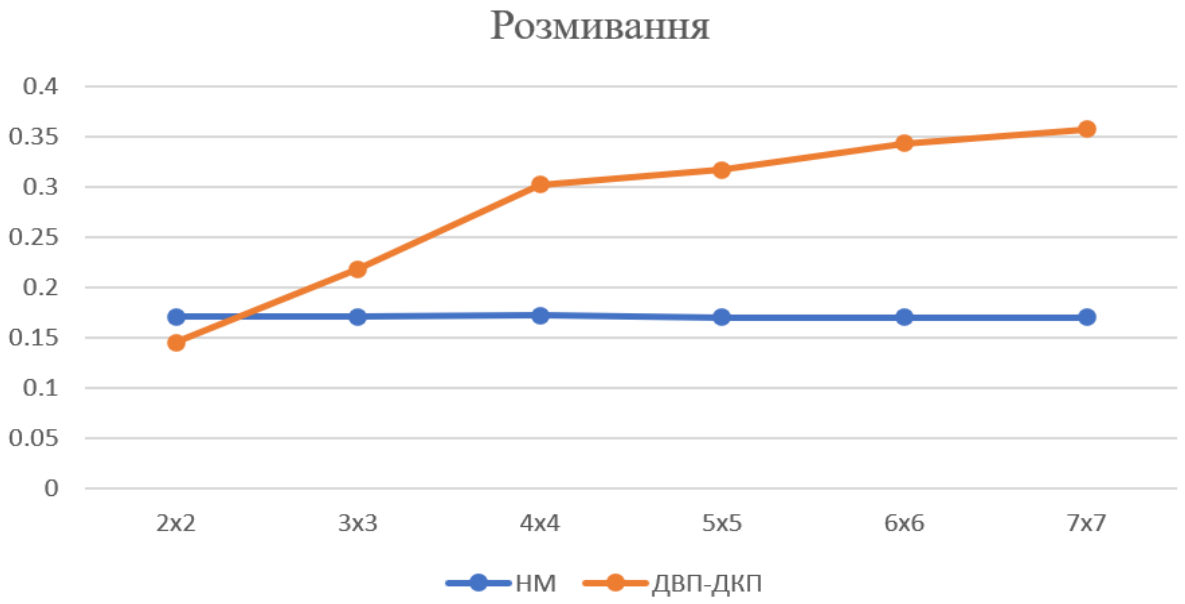

Рисунок 4.5 – Графік среденьоквадратичної помилки при зчитуванні водяного знаку після атаки розмивання

Ні один метод не є стійким до такого виду атак. В обох випадках зображення, використане в якості водяного знаку неможливо відновити з текстури-контейнеру. Приклад впливу атаки розмивання на текстуру є мінімальним (рисунок 4.6, а), текстура не здається сильно розмитою. Та не дивлячись на це, зчитана нейронною мережею ЦВЗ є повністю знищеною і не містить ніяких корисних даних (рисунок 4.6, б). У детермінованого алгоритму результат такий же поганий, це можна зрозуміти виходячи з даних на графіку 4.5.

а) б)

Рисунок 4.6 – Вплив атаки розмивання: а) на текстуру-контейнер; б) ЦВЗ зчитану нейронною мережею

Тепер розгялнемо вплив додавання шуму на зчитування ЦВЗ. Шуми генерувались з двома різними типами розподілу: алгоритмом «сіль та перець» (Salt-and-pepper) та з нормальним розподілом.

Результати роботи кожного з цих алгоритмів залежать від певних параметрів. Нормально розподілений шум генерувався з наступними параметрами: середнє значення μ=1: 0.5: 5 та дисперсією σ^2=1: 0.5: 5. При використанні другий алгоритм змінювалася лише щільність ρ=0: 0.1: 0.5.

Таблиця 4.2 – Отримані помилки після атаки Гауссівським шумом

	μ=1 σ=1	μ=1.5 σ=1.5	μ=2 σ=2	μ=2.5 σ=2.5	μ=3 σ=3	μ=3.5 σ=3.5	μ=4 σ=4	μ=4.5 σ=4.5	μ=5 σ=5
НМ	0.029	0.035	0.046	0.061	0.079	0.099	0.122	0.143	0.161
ДВП-ДКП	0.000	0.000	0.001	0.021	0.056	0.095	0.144	0.158	0.199

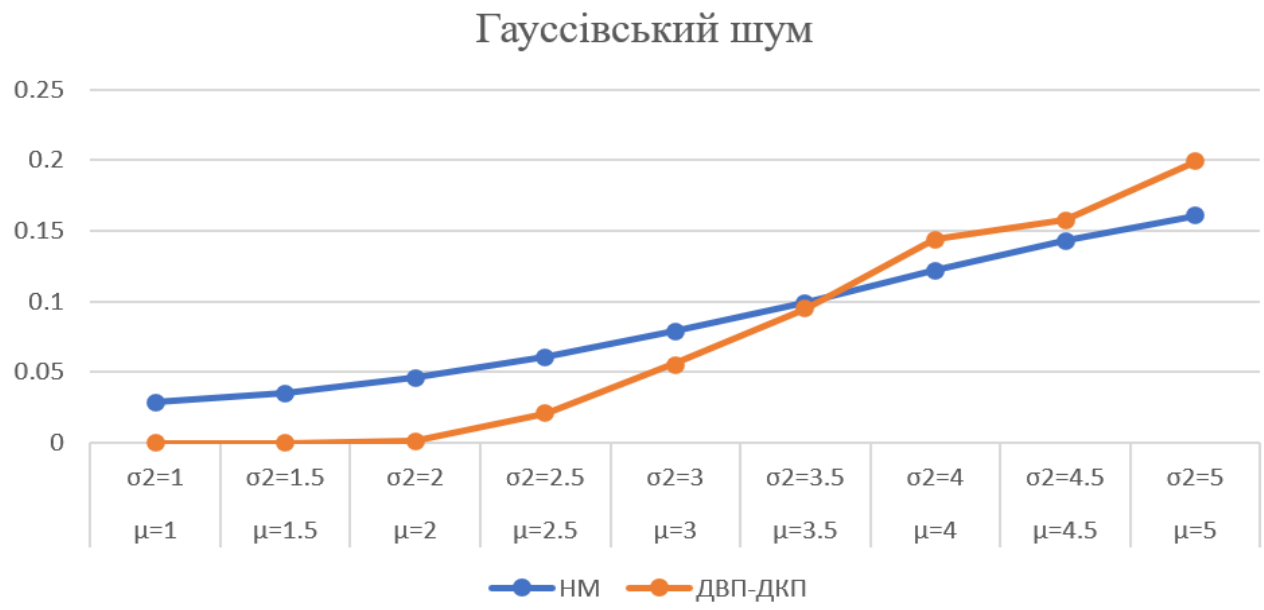

Рисунок 4.7 – Графік помилки при атаці Гауссівським шумом

Шум з нормальним розподілом має майже однаковий вплив на розроблені стегосистеми. При мінімальному впливі шуму детермінований алгоритм показує себе трішки краще, зчитуючи ЦВЗ без помилок, але при збільшенні середнього значення та дисперсії значення помилки навіть перевищує цей показник у системі основаній на НМ.

Таблиця 4.3 – Отримані помилки після атаки шумом «сіль та перець»

	ρ=0.01	ρ=0.02	ρ=0.03	ρ=0.04	ρ=0.05	ρ=0.06	ρ=0.07	ρ=0.07
НМ	0.065	0.091	0.112	0.125	0.137	0.144	0.150	0.155
ДВП-ДКП	0.010	0.063	0.094	0.167	0.192	0.213	0.205	0.221

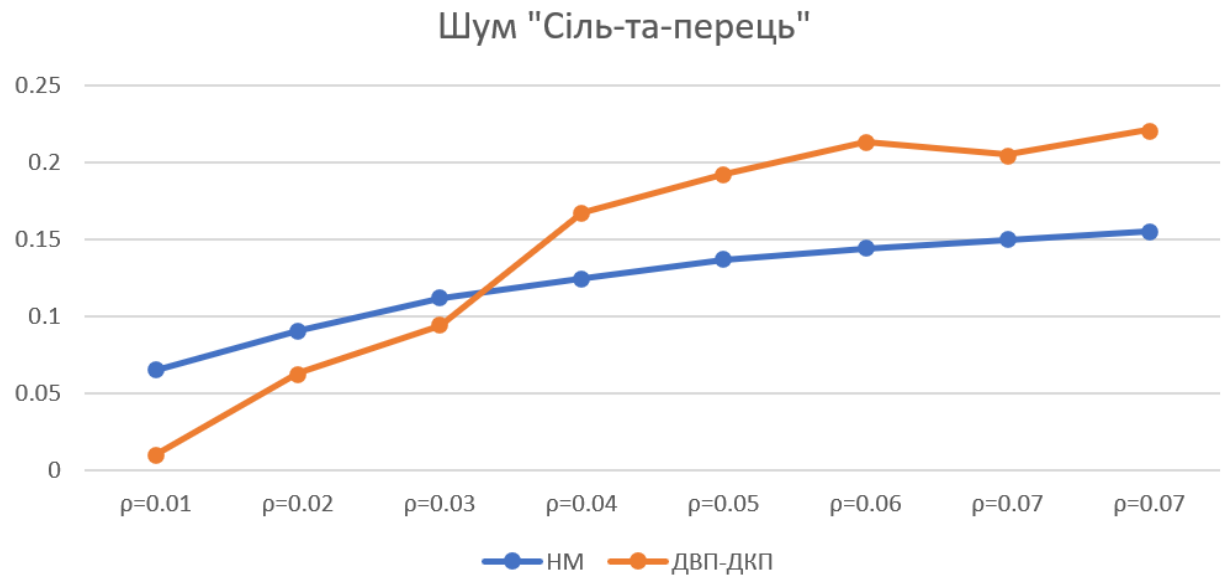

Рисунок 4.8 – Помилка між оригінальним ЦВЗ та зчитаним після атаки шумом типу «Сіль-та-перець»

Результати атак шумом типу «Сіль-та-перець» є схожим на отримані дані для Гауссівського шуму, але переваги системи на основі НМ є значно більш вираженими, помилок значно менше при збільшенні щільності шуму.

Ще однією з можливих атак є зміна яскравості. Тобто значення кожного пікселя може бути підвищене або навпаки зменшене на певне значення. Вплив такого типу атаки був проаналізований з наступними зміщеннями якскравості k=-25: 1: 25. Результати приведені в таблиці 4.4.

Таблиця 4.4 – Отримані помилки після атаки стисненням

	k=-25	k=-20	k=-15	k=-10	k=10	k=15	k=20	k=25
НМ	0.025	0.025	0.025	0.026	0.026	0.027	0.027	0.027
ДВП-ДКП	0.198	0	0.201	0	0.001	0.2	0.001	0.201

Рисунок 4.9 – Графік помилки при зчитуванні водяного знаку після зміни яскравості

Виходячи з даних, зображених на рисунку 4.9, можна зробити висновок, що система ЦВЗ на основі нейронної мережі є повністю стійкою до зміни яскравості. Використаний детермінований алгоритм навпаки є повністю нестійким і, завдяки особливості зчитування бітів ЦВЗ, при певних зміщення всіх пікселів водяний знак может бути повністю зруйнованим.

Останнім розглянутим типом атак є стиснення. Оскільки стисення є перетворенням під час якого частина оригінальних даних втрачається – то відповідно й дані ЦВЗ можуть бути втрачені. Під час експерименту текстура-контейнер стисткалася алгоритмом JPEG з різними коеффіцієнтами якості вихідного зображення k=100: 1: 70. Результати приведені в таблиці 4.5.

Таблиця 4.5 – Отримані помилки після атаки стисненням

	k=99	k=95	k=90	k=85	k=80	k=75	k=70
НМ	0.057	0.065	0.091	0.124	0.154	0.164	0.167
ДВП-ДКП	0.020	0.121	0.202	0.196	0.252	0.244	0.260

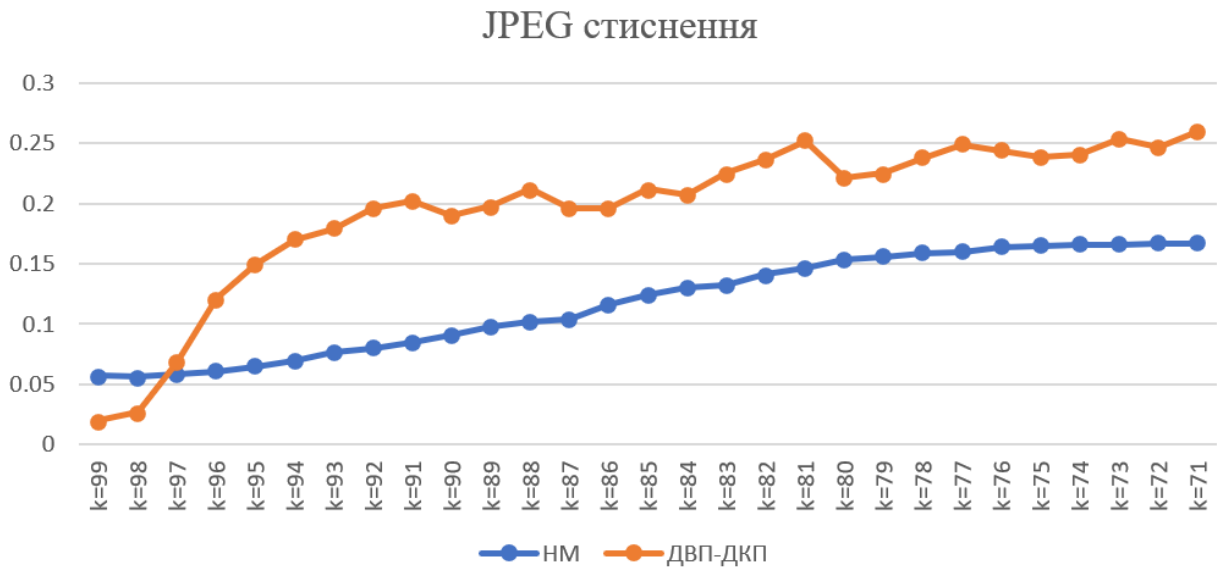

Рисунок 4.10 – Середньоквадратична помилка при зчитування ЦВЗ з контейнеру, який був стискався алгоритмом JPEG

На рисунок 4.10 видно, що при максимальній якості стиснення детермінований алгоритм показує себе трохи краще, але вже починаючи з коефіцієнту якості рівного k=96 він програє НМ. Починаючи з цього коефіцієнту ЦВЗ залишається читаємим лише при використанні НМ. Віходячи зі значень помилки, то розрив між алгоритмами є дуже значним.

Тож виходчи з цих результатів можна зробити висновки, що створена система вбудови ЦВЗ на основі нейронної мережі є більш стійкою до атак. Алгоритм на основі ДВП-ДКП має півну стійкість, але вона значно знижується при збільшенні змін, які завдають атаки до текстури-контейнера.

Безперечною перевагою використання нейронних мереж можна

назвати можливість вбудови дійсно великого об’єму даних, оскільки навчена нейронна мережа може вбудувати в контейнер значно більше інформації, ніж детермінований алгоритм (зображення 256x256 проти 18x18). Нейронна мережа може знаходити частини текстури, зміни яких будуть найменш помітні, а отже при спробі вмістити в контейнер велику кількість прихованої інформації, погіршення якості зображення буде мінімально можливим.

Ще однією вагомою перевагою нейронних мереж є те, що зчитати водяний знак можна лише маючи доступ до самої мережі. Тобто якщо хакер дізнається який детермінований алгоритм був використаний при вбудові ЦВЗ, то йому буде простіше знайти знищити дані ЦВЗ, аніж у випадку використанні нейронної мережі. Навіть якщо хакер дізнається архітектуру нейронної мережі, то він все ще не зможе дізнатися більше деталей щодо модифікованих частин текстури, адже велику роль також відіграють дата-сети, на яких ця нейронна мережа навчалась.

Але у детермінованого алгоритму є перевага у тому, що він може обробляти текстури після відрізання части. Це пов'язано з тим, що алгоритми є блочними і обробляють блоки інформації, в той час як нейронна мережа працює з зображення цілком і має фіксований розмір вхідних даних.

Ще однією перевагою детермінованого алгоритму є менший вплив на текстуру. Вбудова ЦВЗ нейронною мережею має більший вплив на зовнішній вигляд, бо певною мірою підвищує яскравість всього зображення.

ВИСНОВКИ

У ході даної роботи було проаналізовано класифікацію, види та призначення систем цифрових водяних знаків. Було проаналізовано існуючі методи прихованої вбудови даних у текстури, розглянуті їх переваги та недоліки, проведено порівняння. На основі цього було вибрано алгоритми, на основі яких були створені дві системи для роботи з ЦВЗ. Одна з них заснована на комбінації ДВП та ДКТ, тим самим показуючи, що вбудовувати водяні знаки можна використовуючи методи частотної. Протиставлялася цій системі інша – створена на основі згорткової нейронної мережі.

Розроблені системи були протестовані на стійкість різноманітним атакам. До текстур-контейнерів отриманих з обох систем застосовувалися розмивання, стиснення, додавання шуму. Після проведення експерименту було з’ясовано, що ні одна з систем не стійка до розмивання, до шуму й JPEG стиснення більш стійкою виявилася ЦВЗ вбудована нейронною мережею. Враховуючи те, що нейронна мережа може приховувати в текстурі на порядок більше дані, такі результати є гарним доказом того, що система працює дуже добре і розроблена мережа є значно ефективнішою, ніж спосіб вбудови ЦВЗ заснований на дискретних вейвлетному й косинусному перетвореннях.

Подальший розвиток системи вбудови ЦВЗ в текстури передбачає розширення діапазону можливих розмірів текстур, які можуть бути оброблені нейронною мережею. Також варто поліпшити захист від впливу шумів, оскільки вони можуть додаватися під час обробки текстур.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Robust Image Watermarking Theories and Techniques: A Review [Електронний ресурс] / Hai Tao, Li Chongmin, Jasni Mohamad Zain, Ahmed N. Abdalla – Режим доступу до ресурсу: https://www.elsevier.es/es-revista-journal-applied-research-technology-jart-81-articulo-robust-image-watermarking-theories-techniques-S1665642314716128

2. Digital image watermarking: its formal model, fundamental properties and possible attacks [Електронний ресурс] / Hussain Nyeem, Wageeh Boles, Colin Boyd – Режим доступу до ресурсу: https://asp-eurasipjournals.springeropen.com/articles/10.1186/1687-6180-2014-135

3. Digital Watermarking Techniques, Applications and Attacks Applied to Digital Media: A Survey [Електронний ресурс] / Ensaf Hussein, Mohamed A. Belal – Режим доступу до ресурсу: https://www.ijert.org/research/digital-watermarking-techniques-applications-and-attacks-applied-to-digital-media-a-survey-IJERTV1IS7182.pdf

4. Digital Watermarking Applications and Techniques: A Brief Review [Електронний ресурс] / Aaqib Rashid – Режим доступу до ресурсу: https://ijcat.com/archives/volume5/issue3/ijcatr05031006.pdf

5. Attacks and Countermeasures on Digital Watermarks: Classification, Implications, Benchmarks [Електронний ресурс] / Dr. Swati Sherekar, Dr. V.M.Thakare, Dr. Sanjeev Jain – Режим доступу до ресурсу: http://www.researchpublications.org/IJCSA/issue9/2011-IJCSA-134.pdf

6. Literature Review of Wavelet Based Digital Image Watermarking Techniques [Електронний ресурс] / Vaishali S. Jabade, Dr. Sachin R. Gengaje – Режим доступу до ресурсу: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.259.307&rep=rep1&type=pdf

7. Wavelet Transforms in Python [Електронний ресурс] / Режим доступу

до ресурсу: https://pywavelets.readthedocs.io/en/latest/index.html

8. What Is Keras: The Best Introductory Guide To Keras [Електронний ресурс] / Simplilearn – Режим доступу до ресурсу: https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras

9. What is Keras? The deep neural network API explained [Електронний ресурс] / Martin Heller (2021) – Режим доступу до ресурсу: https://www.infoworld.com/article/3336192/what-is-keras-the-deep-neural-network-api-explained.html

10. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way [Електронний ресурс] / Sumit Saha (2018) – Режим доступу до ресурсу: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

11. Convolutional Neural Networks [Електронний ресурс] / IBM Cloud Education (2020) – Режим доступу до ресурсу: https://www.ibm.com/cloud/learn/convolutional-neural-networks

12. Digital watermarking of images using neural networks [Електронний ресурс] / Min-Shiang Hwang, Chin-Chen Chang, Kuo-Feng Hwang – Режим доступу до ресурсу: https://arxiv.org/pdf/1908.11331

13. Методи підвищення надійності водяних знаків у цифрових зображеннях [Текст] / Смирнов В.О., Мартовицький В.О. – Харків: Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління, 2022.

14. Ali Al-Haj (2007). Combined DWT-DCT digital image watermarking. Journal of Computer Science 3(9)(9). DOI: 10.3844/jcssp.2007.740.746

15. Inception Module [Електронний ресурс] / – Режим доступу до ресурсу: https://deepai.org/machine-learning-glossary-and-terms/inception-module

16. A Simple Guide to the Versions of the Inception Network [Електронний ресурс] / Bharath Raj (2018) https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202

17. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning [Електронний ресурс] / Jason Brownlee (2017) – Режим доступу до ресурсу: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

18. Adam: a method for stochastic optimization [Електронний ресурс] / Diederik P. Kingma, Jimmy Lei Ba (2015) – Режим доступу до ресурсу: https://arxiv.org/pdf/1412.6980.pdf

19. Crunch compression of ETC textures [Електронний ресурс] / Alexander Suvorov (2017) – Режим доступу до ресурсу: https://blog.unity.com/technology/crunch-compression-of-etc-textures

20. Understanding PowerVR Series5XT: PVRTC, PVRTC2 and texture compression [Електронний ресурс] / Kristoff Beets (2013) – Режим доступу до ресурсу: https://blog.imaginationtech.com/understanding-powervr-series5xt-pvrtc-pvrtc2-and-texture-compression-part-6/