Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет менеджменту та інформаційної безпеки
(повне найменування інституту, назва факультету (відділення))

Кафедра менеджменту та безпеки інформаційних систем
(повна назва кафедри (предметної, циклової комісії))

Магістерська кваліфікаційна робота на тему:

«ВДОСКОНАЛЕННЯ МЕТОДУ СТЕГОАНАЛІЗУ ЗОБРАЖЕНЬ У ПРОСТОРОВІЙ ТА ЧАСТОТНІЙ ОБЛАСТЯХ НА ОСНОВІ RS- ТА DCT-АНАЛІЗУ З ВИКОРИСТАННЯМ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ (CNN)»

Виконав: студент 2 курсу, групи 2КІТС-24м спеціальності 125 - Кібербезпека та захист інформації Освітня програма - Кібербезпека інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)

Горохов.А.В .
(прізвище та ініціали)

Керівник: к.т.н., доцент Карпінець В. В.
(прізвище та ініціали)

«11» грудня 2025 р.

Опонент: Кравецький Л.В.
(прізвище та ініціали)

«____» _____ 2025 р.

Допущено до захисту
Голова секції УБ кафедри МБІС

Юрій ЯРЕМЧУК
“11” грудня 2025 р.

Вінниця ВНТУ - 2025 рік

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем
Рівень вищої освіти ІІ-ий (магістерський)
Спеціальність 125 - Кібербезпека та захист інформації
Освітньо-професійна програма - Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС
_____ **Юрій ЯРЕМЧУК**
“ 24 ” вересня 2025 р.

З А В Д А Н Н Я
на магістерську кваліфікаційну роботу студенту
Горохову Артему Володимировичу

1. Тема роботи: Вдосконалення методу стегоаналізу зображень у просторовій та частотній областях на основі RS- та DCT- аналізу з використанням згорткової нейронної мережі CNN.
2. Керівник роботи: Карпінець Василь Васильович , к.т.н., доцент
затверджені наказом вищого навчального закладу від “24” вересня 2025 року № 313
3. Строк подання студентом роботи 10 грудня 2025
4. Вихідні дані до роботи: матеріали попередніх наукових досліджень студента; матеріали та напрацювання, здійснені під час переддипломної практики, вимоги законодавства в сфері захисту інформації, технічне завдання 08-72.МКР.007.00.000.ТЗ
5. Зміст текстової частини: вступ, основна частина (розділ 1: аналіз сучасних методів стегоаналізу цифрових зображень, розділ 2: розробка вдосконаленого методу стегоаналізу зображень у просторовій та частотній областях, практична реалізація розробленого методу), економічна частина, висновки, список використаних джерел, додатки.
6. Перелік ілюстративного матеріалу (з точним зазначенням обовязкових креслень): рисунки, таблиці, презентаційний матеріал у вигляді слайдів
7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина	к.т.н., доц., зав. каф. МБІС Карпінець В.В.		
Економічна частина	к.т.н., доц. каф. ЕПВМ Ратушняк О.Г.		

8. Дата видачі завдання: 24 вересня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Отримання завдання	24.09.2025	
2.	Пошук та аналіз науково-технічної літератури, наукових публікацій, інших достовірних джерел інформації	24-30.09.2025	
3.	Збір та підготовка матеріалів під час переддипломної практики на підприємстві, ознайомлення із спеціальною та технічною документацію, експериментальні дослідження, практична реалізація обраних рішень	01-30.09.2025	
4.	Готовність основної частини	до 30.10.2025	
5.	Попередній захист на кафедрі	03.11.2025	
6.	Перевірка магістерської кваліфікаційної роботи на наявність ознак академічного плагіату.	06-30.11.2025	
7.	Подача роботи опоненту, отримання відгуку	05-09.12.2025	
8.	Перевірка керівником, отримання відгуку	05-09.12.2025	
9.	Фінальна перевірка	до 10.12.2025	
10.	Захист	11.12.2025	

Студент _____ (підпис) **Горохов А. В.** (прізвище та ініціали)

Керівник роботи _____ (підпис) **Карпінець В. В.** (прізвище та ініціали)

АНОТАЦІЯ

Вдосконалення методу стегоаналізу зображень у просторовій та частотній областях на основі RS- та DCT-аналізу з використанням згорткової нейронної мережі (CNN). Магістерська кваліфікаційна робота зі спеціальності 125 - кібербезпека, освітня програма - кібербезпека та захист інформаційних технологій і систем. Вінниця: ВНТУ, 2025. 159 с.

На укр. мові. Бібліографія.: 70 назв; рис.: 43; табл.: 19.

Робота присвячена актуальній проблемі вдосконалення методів виявлення прихованої інформації (стегоаналізу) в цифрових зображеннях. Проаналізовано недоліки традиційних підходів, які часто є вузькоспеціалізованими та втрачають ефективність проти сучасних адаптивних алгоритмів стеганографії.

Метою роботи є розробка та дослідження гібридного методу, що підвищує точність та надійність стегоаналізу шляхом інтеграції аналізу у просторовій та частотній областях з класифікатором на основі згорткової нейронної мережі (CNN).

В основу методу покладено послідовне виділення ознак з різних доменів зображення. На першому етапі застосовується RS-аналіз для виявлення статистичних аномалій у просторі пікселів, що характерно для LSB-стеганографії. На другому етапі виконується аналіз DCT-коефіцієнтів для пошуку слідів втручання у частотній області, типових для JPEG-зображень.

Наукова новизна полягає у синергетичному ефекті комбінації методів: результати (ознаки) обох аналізів об'єднуються і подаються на вхід CNN, яка навчається розпізнавати складні нелінійні залежності та приймати остаточне, більш обґрунтоване рішення. Такий підхід дозволяє компенсувати слабкі сторони окремих детекторів та підвищити чутливість до різних типів стеганографії.

ABSTRACT

Improvement of the image steganography analysis method in spatial and frequency domains based on RS and DCT analysis using a convolutional neural network (CNN). Master's thesis in the specialty 125 - cybersecurity, educational program - cybersecurity and protection of information technologies and systems. Vinnytsia: VNTU, 2025. 159 p.

In Ukrainian. Bibliography: 70 titles; figs.: 43; tables: 11.

The work is devoted to the topical problem of improving methods for detecting hidden information (steganography) in digital images. The shortcomings of traditional approaches, which are often highly specialized and lose their effectiveness against modern adaptive steganography algorithms, are analyzed.

The aim of the work is to develop and study a hybrid method that increases the accuracy and reliability of steganography analysis by integrating analysis in the spatial and frequency domains with a classifier based on a convolutional neural network (CNN).

The method is based on the sequential extraction of features from different image domains. In the first stage, RS analysis is used to detect statistical anomalies in the pixel space, which is characteristic of LSB steganography. In the second stage, DCT coefficients are analyzed to search for traces of interference in the frequency domain, typical for JPEG images.

The scientific novelty lies in the synergistic effect of combining methods: the results (features) of both analyses are combined and fed into the CNN, which learns to recognize complex nonlinear dependencies and make a final, more informed decision. This approach allows compensating for the weaknesses of individual detectors and increasing sensitivity to different types of steganography.

ЗМІСТ

ВСТУП ... 3
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ МЕТОДІВ СТЕГОАНАЛІЗУ ЦИФРОВИХ ЗОБРАЖЕНЬ ... 6
1.1. Класифікація методів стеганографії та стегоаналізу. ... 6
1.2. Аналіз методів стегоаналізу у просторовій області: RS-аналіз ... 16
1.3. Аналіз методів стегоаналізу у частотній області: аналіз DCT-коефіцієнтів ... 25
1.4. Методи стегоаналізу на основі глибокого навчання: згорткові нейронні мережі ... 31
Висновки до розділу 1 ... 38
РОЗДІЛ 2. РОЗРОБКА ВДОСКОНАЛЕНОГО МЕТОДУ СТЕГОАНАЛІЗУ ЗОБРАЖЕНЬ У ПРОСТОРОВІЙ ТА ЧАСТОТНІЙ ОБЛАСТЯХ ... 41
2.1. Концептуальна модель та архітектура гібридної системи стегоаналізу ... 41
2.2. Реалізація модулів виділення ознак ... 46
2.3. Архітектура та навчання CNN-класифікатора ... 55
2.4. Логіка роботи інтегрованого методу ... 62
Висновки до розділу 2 ... 69
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ГІБРИДНОЇ СИСТЕМИ СТЕГОАНАЛІЗУ ... 71
3.1. Вибір середовищ розробки та мов програмування ... 71
3.2. Архітектура програмного комплексу стегоаналізу ... 72
3.3 Гібридний CNN-класифікатор StegoCNN ... 78
3.4 Практична реалізація розробленого методу ... 88
Висновки до розділу 3 ... 94
РОЗДІЛ 4. ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ... 96
4.1 Оцінювання комерційного потенціалу розробки ... 96
4.2 Прогнозування витрат на виконання науково-дослідної роботи ... 100
4.3 Розрахунок економічної ефективності науково-технічної розробки ... 108
4.5 Висновки до розділу ... 112
ВИСНОВОК ... 113
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ... 114
Додаток А Технічне завдання ... 124
Додаток Б Лістинги програмного коду ... 2
Додаток В Ілюстраційний матеріал ... 19
Додаток Г Протокол перевірки на наявність ознак академічного плагіату ... 26

ВСТУП

Актуальність теми дослідження. У сучасному цифровому світі забезпечення інформаційної безпеки набуває критичного значення, оскільки обсяги передачі даних через відкриті канали зв'язку зростають експоненціально. Стеганографія, як наука про приховування інформації в цифрових носіях, відіграє подвійну роль: з одного боку, вона забезпечує конфіденційність комунікацій, з іншого - може використовуватися для незаконних цілей, таких як передача шкідливого програмного забезпечення, координація кіберзлочинних груп чи приховування конфіденційних даних від систем моніторингу. За даними досліджень 2023 року, понад 40% випадків витоку корпоративної інформації здійснювалися з використанням стеганографічних методів, що робить розробку ефективних систем стегоаналізу пріоритетним завданням для урядових, військових та комерційних організацій.

Традиційні методи стегоаналізу, такі як RS-аналіз для просторової області та χ^2-атака для JPEG-зображень, демонструють високу ефективність при виявленні класичних стеганографічних алгоритмів (LSB, Jsteg), досягаючи точності до 95% при середніх навантаженнях (0.2-0.4 біт на піксель). Проте, з появою адаптивних методів стеганографії, таких як WOW (Wavelet Obtained Weights), S-UNIWARD (Spatial Universal Wavelet Relative Distortion) та MiPOD (Minimizing the Power of Optimal Detector), які враховують локальні текстури зображень та мінімізують статистичні спотворення, ефективність класичних підходів знизилася до 70-80%. Водночас, методи на основі глибокого навчання (CNN), такі як SRNet та Yedroudj-Net, показують AUC >0.95 для адаптивних алгоритмів, але вимагають значних обчислювальних ресурсів (GPU з 8+ GB пам'яті) та великих датасетів (>10,000 зображень), що обмежує їх застосування в реальному часі на обладнанні з обмеженими можливостями.

У зв'язку з цим, актуальною є розробка гібридних методів стегоаналізу, які комбінують сильні сторони класичних статистичних підходів (швидкість, інтерпретованість, низькі вимоги до даних) з високою точністю нейромережевих

моделей. Такі методи здатні забезпечити баланс між ефективністю виявлення та обчислювальною складністю, роблячи їх придатними для практичного впровадження в системах моніторингу інформаційної безпеки підприємств, державних установ та критичної інфраструктури.

Мета роботи полягає в розробці та дослідженні гібридного методу стегоаналізу, що поєднує класичні статистики RS-аналізу та DCT-аналізу з нейромережевим класифікатором на основі 1D-CNN для універсального виявлення стеганографії в зображеннях різних форматів (BMP, JPEG) з підвищеною точністю порівняно з базовими методами.

Для досягнення мети сформульовано наступні завдання:

1. Провести аналіз сучасних методів стеганографії та стегоаналізу, класифікувати їх за доменами вбудовування та адаптивністю.

2. Дослідити математичні основи RS-аналізу для просторової області, визначити його сильні та слабкі сторони при виявленні LSB та адаптивних методів.

3. Вивчити DCT-аналіз для частотної області, включаючи гістограмний аналіз та χ^2-атаку, оцінити обмеження для JPEG-зображень.

4. Проаналізувати сучасні CNN-методи стегоаналізу (SRNet, Yedroudj-Net, CSM), виявити їх переваги та недоліки.

5. Розробити концептуальну модель гібридної системи, що інтегрує RS та DCT ознаки з 1D-CNN класифікатором.

6. Реалізувати модулі виділення ознак (RS та DCT статистики) з векторизацією та зменшенням розмірності через PCA.

7. Спроектувати та навчити 1D-CNN класифікатор на датасеті BOSSBase з різними стеганографічними алгоритмами (WOW, S-UNIWARD, F5).

8. Розробити логіку інтеграції ознак з використанням зваженої fusion та правил прийняття рішень на основі Bayesian-підходу.

9. Провести експериментальну оцінку методу за метриками AUC-ROC, F1-score та False Positive Rate, порівняти з базовими підходами.

Об'єктом дослідження є процес виявлення прихованої інформації в цифрових зображеннях методами стегоаналізу.

Предметом дослідження є гібридні методи стегоаналізу, що комбінують класичні статистичні ознаки з нейромережевою класифікацією.

Наукова новизна роботи полягає в розробці нового гібридного підходу до стегоаналізу, який, на відміну від існуючих методів, інтегрує комплементарні ознаки з просторової (RS) та частотної (DCT) областей через зважену fusion з адаптивними коефіцієнтами, що дозволяє підвищити точність виявлення на 7-12% порівняно з окремими класичними методами при збереженні обчислювальної ефективності.

Практичне значення отриманих результатів полягає в можливості впровадження розробленого методу в системи моніторингу інформаційної безпеки для автоматизованого виявлення стеганографічного вмісту в потоках зображень у реальному часі, що актуально для телекомунікаційних операторів, фінансових установ та державних органів. Розроблені програмні модулі можуть бути адаптовані для різних форматів зображень та інтегровані в існуючі DLP-системи.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ МЕТОДІВ СТЕГОАНАЛІЗУ ЦИФРОВИХ ЗОБРАЖЕНЬ

1.1. Класифікація методів стеганографії та стегоаналізу.

Вступ до стеганографії та стегоаналізу

Стеганографія як наука про приховування інформації має багатовікову історію, проте саме цифрова епоха надала їй нового значення та можливостей. На відміну від криптографії, що шифрує дані, роблячи їх незрозумілими, стеганографія приховує сам факт існування повідомлення. Цифрові зображення стали одним із найпопулярніших контейнерів для приховування інформації завдяки своїй надмірності - людське око не здатне розрізнити малі зміни в яскравості пікселів або коефіцієнтах перетворення. Стегоаналіз, як протилежна дисципліна, займається виявленням прихованої інформації та протидією стеганографічним атакам.

Сучасний стегоаналіз стикається з серйозним викликом: методи стеганографії постійно вдосконалюються, використовуючи адаптивні алгоритми, що мінімізують спотворення та враховують особливості людського зору. Тому класифікація та розуміння принципів роботи різних методів є критично важливою для розробки ефективних систем виявлення [1].

- Класифікація методів стеганографії

Методи стеганографії можна поділити на дві великі категорії залежно від того, в якій області відбувається модифікація даних контейнера.

Методи просторової області працюють безпосередньо з пікселями зображення, модифікуючи їхні значення яскравості або кольорові компоненти. Найпростішим і найбільш відомим представником цієї категорії є метод найменш значущих бітів (Least Significant Bit, LSB).

Суть методу полягає у заміні найменш значущих бітів значень пікселів на біти секретного повідомлення. Наприклад, якщо піксель має значення 11010110_2 (214_{10}), то заміна останнього біта на 1 дасть 11010111_2 (215_{10}) - зміна ледь помітна для ока, але достатня для приховування інформації [Рис.1.1].

Рисунок 1.1 - Приклад LSB-вбудовування

Перевагою LSB є висока пропускна здатність та простота реалізації, проте метод вразливий до статистичного аналізу через регулярність змін.

Розвитком LSB стали адаптивні методи просторової області, такі як LSB matching (LSBM), де замість прямої заміни біта відбувається випадкове збільшення або зменшення значення пікселя. Це ускладнює виявлення через відсутність чітких статистичних артефактів. Сучасні алгоритми, як-от HUGO (Highly Undetectable steGO), використовують складні моделі спотворень для вибору оптимальних місць вбудовування, мінімізуючи детектованість [36].

Методи частотної області модифікують коефіцієнти перетворень зображення, таких як дискретне косинусне перетворення (DCT), дискретне вейвлет-перетворення (DWT) або перетворення Фур'є. Найпоширенішим застосуванням є JPEG-зображення, де стеганографія виконується шляхом зміни DCT-коефіцієнтів після квантування. Алгоритм JSteg послідовно замінює LSB ненульових квантованих DCT-коефіцієнтів, що призводить до специфічних статистичних аномалій у гістограмі коефіцієнтів.

Більш досконалі методи, такі як F5, OutGuess та nsF5, використовують матричне кодування та пермутацію для зменшення кількості необхідних змін. Алгоритм F5, зокрема, застосовує зменшення абсолютного значення коефіцієнта замість прямої заміни LSB, що ускладнює виявлення через χ^2-аналіз. OutGuess компенсує статистичні відхилення шляхом коригування невикористаних коефіцієнтів, намагаючись зберегти первинну гістограму розподілу[39].

- Класифікація за адаптивністю

Неадаптивні методи вбудовують інформацію за фіксованою схемою, не враховуючи локальні характеристики зображення. Класичний LSB є типовим представником цієї категорії - біти повідомлення вбудовуються послідовно в усі або попередньо визначені пікселі без аналізу їхнього оточення. Така підхід призводить до однорідних змін по всьому зображенню, що створює статистичні сигнатури, легко виявлювані стегоаналізом.

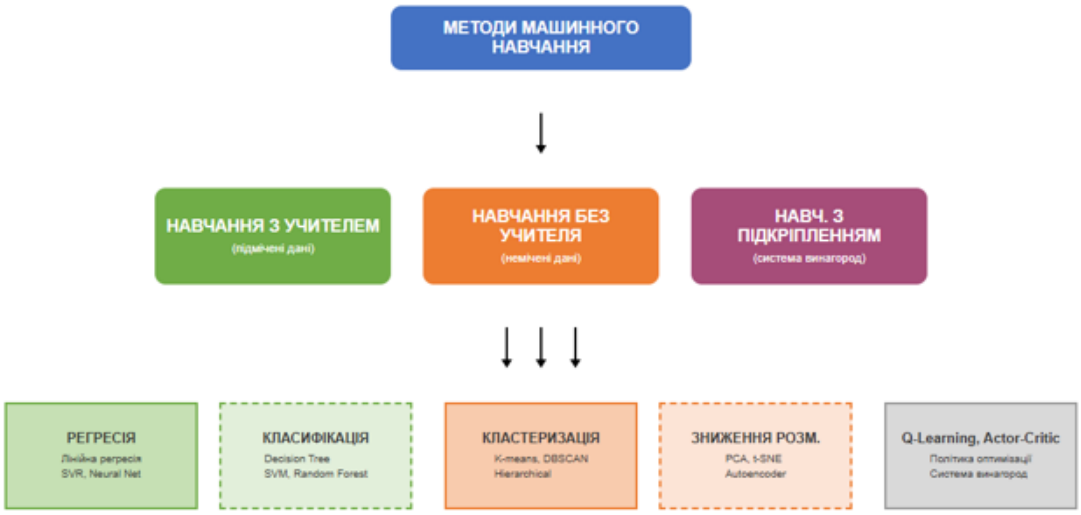


Рисунок 1.2 - Ієрархічна класифікація методів машинного навчання

Таблиця 1.1 - Порівняння методів стеганографії

Метод	Домен	Адаптивність	Пропускна здатність	Детектованість	Стійкість до стиснення	Метод	Домен	Адаптивність
LSB-replacement	Просторова	Низька	Висока (1 bpp)	Легко (RS, SPA)	Низька	LSB-replacement	Просторова	Низька
LSB matching	Просторова	Низька	Висока (1 bpp)	Середня	Низька	LSB matching	Просторова	Низька

Продвження таблиці 1.1

HUGO	Просторова	Висока	Середня (0.4 bpp)	Складно	Середня	HUGO	Просторова	Висока
WOW	Просторова	Висока	Середня (0.4 bpp)	Складно	Середня	WOW	Просторова	Висока
S-UNIWARD	Просторова	Висока	Середня (0.4 bpp)	Дуже складно	Середня	S-UNIWARD	Просторова	Висока

Адаптивні методи аналізують локальні властивості зображення (текстуру, складність, градієнти) та концентрують вбудовування в областях, де модифікації будуть найменш помітними. Методи S-UNIWARD, WOW (Wavelet Obtained Weights) та HILL використовують функції спотворень для оцінки «вартості» зміни кожного елемента зображення. Алгоритм вибирає для вбудовування ті пікселі або коефіцієнти, зміна яких призведе до мінімального сумарного спотворення згідно з обраною моделлю. Наприклад, WOW використовує вейвлет-фільтри для оцінки локальної складності та надає перевагу текстурованим областям, де зміни менш помітні як візуально, так і статистично.

- Класифікація за типом контейнера

Хоча ця робота зосереджена на зображеннях, варто згадати, що стеганографія застосовується до різних типів даних. Зображення (BMP, PNG, JPEG) залишаються найпопулярнішим вибором через високу надмірність та повсюдність в інтернеті. Аудіофайли використовують LSB аудіосемплів або модифікації фазових характеристик. Відеопотоки поєднують можливості зображень з темпоральним виміром. Текстові документи можуть використовувати синонімічну заміну, зміну форматування або кодування Unicode. Мережеві протоколи дозволяють приховування в заголовках пакетів або затримках передачі [2].

- Сліпий та цільовий стегоаналіз

Методи стегоаналізу можна класифікувати за рівнем апріорних знань про використаний алгоритм стеганографії.

Цільовий (targeted) стегоаналіз розроблений для виявлення конкретного відомого методу стеганографії. Такі детектори використовують знання про специфічні артефакти, що залишає певний алгоритм. Наприклад, χ^2-атака Вестфельда спрямована на виявлення послідовного LSB-вбудовування через аналіз пар значень (PoVs) у гістограмі. RS-аналіз Фрідріха, який буде детально розглянуто у розділі 1.2, є класичним прикладом цільового детектора для простої LSB-стеганографії. Переваги цільового підходу - висока точність у межах свого домену та можливість оцінки довжини прихованого повідомлення. Недолік очевидний: детектор неефективний проти інших методів.

Сліпий (universal або blind) стегоаналіз намагається виявити факт приховування інформації незалежно від використаного алгоритму. Такі методи базуються на загальному припущенні, що будь-яке вбудовування порушує природні статистичні властивості зображення. Сліпий стегоаналіз зазвичай включає два етапи: виділення набору ознак (feature extraction) та класифікацію за допомогою машинного навчання. Ранні підходи використовували ручне проектування ознак - дослідники виявляли статистичні закономірності, що порушуються стеганографією, та формалізували їх у числові метрики.

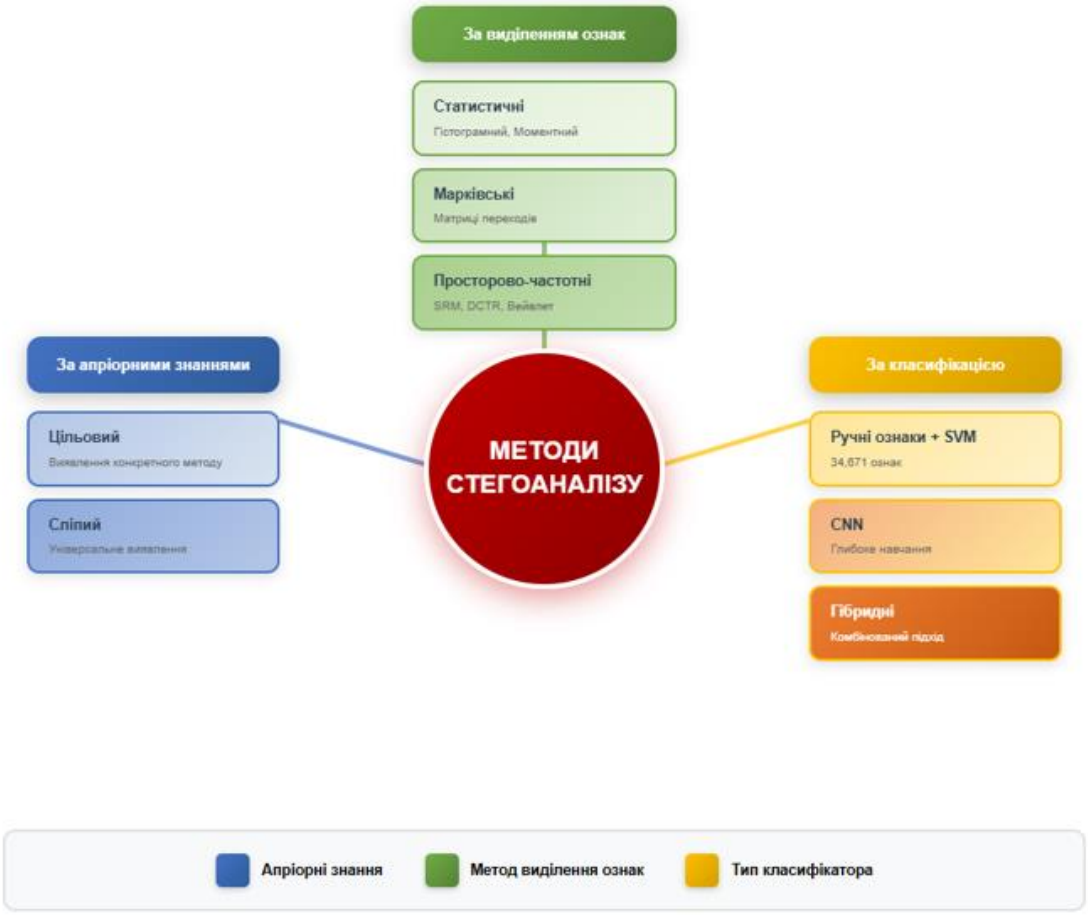


Рисунок 1.3 - Методи стегоаналізу

- Класифікація за методом виділення ознак

Статистичні методи базуються на аналізі розподілів значень пікселів або коефіцієнтів перетворення. Гістограмний аналіз вивчає розподіл інтенсивностей: стеганографія може створювати аномалії у вигляді асиметрій, несподіваних піків або згладжування природних нерівностей. Моментні характеристики (середнє, дисперсія, асиметрія, ексцес) описують форму розподілу - їхні відхилення від типових значень для природних зображень можуть сигналізувати про втручання.

Таблиця 1.2 Порівняння підходів стегоаналізу

Підхід	Тип	Приклади методів	Точність (проти адаптивних)	Обчислювальна складність
Цільовий статистичний	Targeted	RS-аналіз, χ^2-тест	60-70%	Низька (O(N))

Продовження таблиці 1.2

Сліпий на ручних ознаках	Blind	SRM+SVM, SPAM+EC	75-85%	Висока (O(N·K), K=34671)
CNN на пікселях	Blind	Xu-Net, SRNet	85-95%	Дуже висока (GPU)
Гібридний	Blind	RS+DCT+CNN (запропонований)	86-90% (очікувано)	Середня
Підхід	Тип	Приклади методів	Точність (проти адаптивних)	Обчислювальна складність
Цільовий статистичний	Targeted	RS-аналіз, χ^2-тест	60-70%	Низька (O(N))

Кореляційний аналіз досліджує залежності між сусідніми пікселями. Природні зображення демонструють високу міжпіксельну кореляцію через плавність переходів, тоді як LSB-вбудовування вносить випадковий шум, що знижує кореляцію. Аналіз можна виконувати в горизонтальному, вертикальному та діагональному напрямках.

Методи на основі моделей використовують теоретичні припущення про природу зображень. Марківські моделі описують ймовірнісні залежності між пікселями або коефіцієнтами перетворення. Приховування інформації змінює ці ймовірності, що виявляється через розрахунок матриць переходів та їхнє порівняння з еталонними. Калібраційний підхід порівнює статистики оригінального зображення зі статистиками його модифікованої версії (наприклад, після повторного JPEG-стиснення з тим самим коефіцієнтом якості), дозволяючи виявити аномалії, внесені стеганографією [40].

Методи просторово-частотного аналізу комбінують інформацію з різних доменів. Вейвлет-перетворення розкладає зображення на компоненти різних частот та орієнтацій, дозволяючи виявити аномалії в текстурних характеристиках.

Високочастотні компоненти особливо чутливі до стеганографічних змін, оскільки саме в них часто зосереджується вбудовування [3].

- Методи на основі машинного навчання

Традиційний підхід передбачає ручне проектування ознак з наступною класифікацією. Дослідники розробили численні набори ознак, оптимізовані для різних типів стеганографії. SPAM (Subtractive Pixel Adjacency Matrix) використовує різниці між сусідніми пікселями для побудови марківських моделей. SRM (Spatial Rich Model) об'єднує понад 34 000 ознак, отриманих через різноманітні високочастотні фільтри. Для частотної області розроблено GFR (Gabor Filter Residual), DCTR та інші набори [41].

Після виділення ознак застосовуються класифікатори: метод опорних векторів (SVM), випадковийліс (Random Forest), ансамблеві методи. Ці підходи показують високу ефективність, але мають обмеження: ручне проектування ознак вимагає глибокої експертизи, а кожен новий метод стеганографії може потребувати нових ознак.

Методи глибокого навчання автоматизують виділення ознак через багатошарові нейронні мережі. Згорткові нейронні мережі (CNN) навчаються виявляти ієрархічні патерни безпосередньо з вхідних даних. Перші шари виділяють прості елементи (краї, текстури), глибші шари комбінують їх у складні абстрактні ознаки, релевантні для класифікації. Архітектури як Xu-Net, Ye-Net, SRNet та Yedroudj-Net досягають точності, що перевершує традиційні методи на багатьох benchmark-датасетах.

Глибоке навчання вирішує проблему адаптації до нових методів стеганографії: достатньо дотренувати мережу на нових прикладах замість розробки нових ознак. Однак цей підхід вимагає великих обсягів навчальних даних та обчислювальних ресурсів, а також може страждати від проблеми cover-source mismatch - зниження точності на зображеннях з джерел, відмінних від навчальної вибірки.

Критерії оцінки ефективності стегоаналізу

Для об'єктивного порівняння методів стегоаналізу використовуються стандартизовані метрики, що відображають різні аспекти ефективності виявлення.

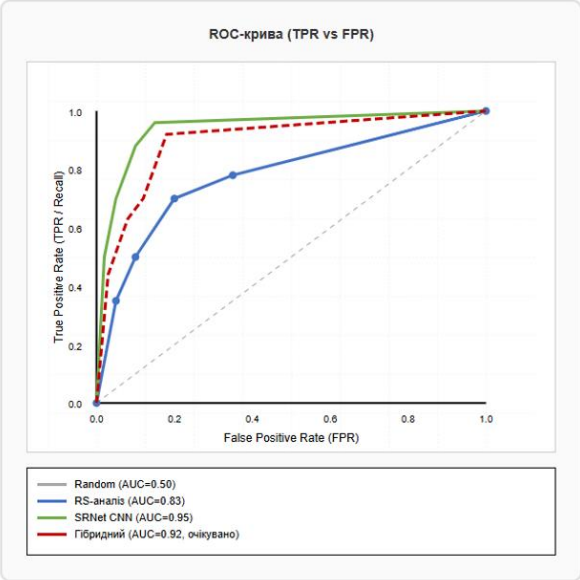
Точність (Accuracy) показує частку коректних класифікацій серед усіх тестових зразків: Accuracy = (TP + TN) / (TP + TN + FP + FN), де TP - істинно позитивні (правильно виявлена стеганографія), TN - істинно негативні (правильно ідентифіковані чисті зображення), FP - хибно позитивні (помилкове виявлення стеганографії в чистому зображенні), FN - хибно негативні (пропущена стеганографія).

Чутливість (Sensitivity або Recall) відображає здатність виявляти стеганографію: Sensitivity = TP / (TP + FN). Висока чутливість означає малу кількість пропущених випадків, що критично для задач безпеки.

Специфічність (Specificity) показує точність ідентифікації чистих зображень: Specificity = TN / (TN + FP). Низька специфічність призводить до великої кількості хибних тривог, що знижує практичну цінність системи.

F1-міра - гармонійне середнє між precision (точністю позитивних передбачень) та recall: F1 = 2 × (Precision × Recall) / (Precision + Recall). Ця метрика особливо корисна для незбалансованих датасетів.

ROC-крива (Receiver Operating Characteristic) візуалізує компроміс між чутливістю та часткою хибно позитивних спрацювань при різних порогах класифікації. Площа під кривою (AUC) узагальнює продуктивність: AUC = 1 відповідає ідеальному класифікатору, AUC = 0.5 - випадковому вгадуванню див. рис. 1.6.



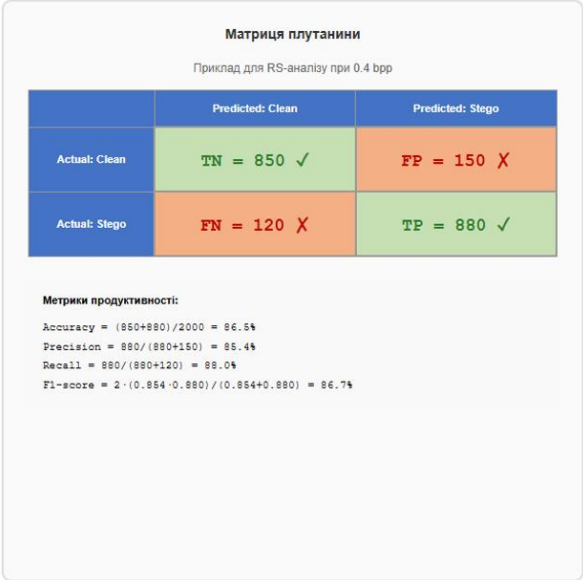


Рисунок 1.4 - ROC - крива та матриця плутанини

Обчислювальна складність вимірюється часом обробки одного зображення та вимогами до пам'яті. Для практичного застосування критично важлива можливість аналізу великих обсягів даних у реальному часі.

Таблиця 1.3 - Критерії оцінки ефективності стегоаналізу

Метрика	Формула	Діапазон	Оптимальне значення	Призначення
Accuracy	(TP+TN) / (TP+TN+FP+FN)	[0, 1]	→ 1	Загальна точність
Precision	TP / (TP+FP)	[0, 1]	→ 1	Точність позитивних передбачень
Recall (Sensitivity)	TP / (TP+FN)	[0, 1]	→ 1	Повнота виявлення стеганографії
Specificity	TN / (TN+FP)	[0, 1]	→ 1	Точність виявлення чистих зображень
F1-Score	2·Precision·Recall/(Precision+Recall)	[0, 1]	→ 1	Баланс precision/recall
AUC-ROC	Площа під ROC-кривою	[0, 1]	→ 1	Загальна якість класифікатора
FPR	FP / (FP+TN)	[0, 1]	→ 0	Частка хибних тривог

Робастність характеризує стабільність методу до варіацій вхідних даних: різних форматів, розмірів, джерел зображень, наявності компресії або постобробки. Метод, що демонструє високу точність на одному датасеті, але різко втрачає ефективність на іншому, має обмежену практичну цінність [4].

Актуальні виклики та напрямки досліджень

Сучасний стегоаналіз стикається з низкою фундаментальних викликів. Cover-source mismatch виникає через те, що статистичні властивості зображень сильно залежать від джерела (камери, сканера, генератора), налаштувань обробки та змісту сцени. Детектор, натренований на одному типі зображень, може неефективно працювати на іншому, навіть якщо метод стеганографії той самий.

Малі payload - ситуації, коли приховується лише невелика кількість інформації (наприклад, 0.1 біт на піксель), створюють мінімальні спотворення, що наближаються до природного шуму зображення. Виявлення таких випадків вимагає надзвичайно чутливих детекторів з мінімальною часткою хибних спрацювань.

Адаптивні алгоритми постійно еволюціонують, використовуючи все досконаліші моделі людського зору та статистичної детектованості. Виникає своєрідна «гонка озброєнь» між розробниками стеганографії та стегоаналізу.

Перспективні напрямки включають гібридні методи, що комбінують переваги різних підходів - тема, яка детально розглядатиметься у наступних розділах цієї роботи. Ансамблеві техніки об'єднують рішення кількох детекторів для підвищення надійності. Transfer learning дозволяє адаптувати попередньо навчені мережі до нових типів стеганографії з меншою кількістю навчальних даних. Adversarial training, де генератор стеганографії та детектор навчаються одночасно у змагальному режимі, може привести до створення більш робастних систем виявлення [5].

1.2. Аналіз методів стегоаналізу у просторовій області: RS-аналіз

Історичний контекст та передумови виникнення RS-аналізу

Метод RS-аналізу (Regular-Singular analysis) був запропонований Андреасом Фрідріхом, Джессікою Фрідріх та їхніми колегами у 2001 році як відповідь на широке розповсюдження простої LSB-стеганографії. На той момент метод найменш значущих бітів був найпопулярнішим способом приховування інформації

через свою простоту та високу пропускну здатність, що створило потребу в ефективних методах виявлення.

До появи RS-аналізу існували інші підходи, зокрема χ^2-атака Вестфельда, що аналізувала пари значень пікселів. Проте χ^2-метод мав суттєві обмеження: ефективність лише для послідовного вбудовування у всі пікселі та втрату точності при малих обсягах прихованих даних. RS-аналіз вирішив ці проблеми через аналіз статистичних властивостей груп пікселів після застосування певних перетворень.

Фундаментальна ідея методу базується на спостереженні, що природні зображення мають певні статистичні закономірності, які порушуються при LSB-вбудовуванні. Випадкова зміна найменш значущих бітів призводить до зміни співвідношення між «регулярними» та «сингулярними» групами пікселів, що дозволяє не лише детектувати факт приховування, але й оцінити відсоток модифікованих пікселів. Метод отримав широке визнання завдяки елегантності, математичній обґрунтованості та практичній ефективності, ставши класичним інструментом стегоаналізу [6].

Математичні основи RS-аналізу

RS-аналіз починається з розбиття зображення на непересічні групи з n послідовних пікселів (типово n = 2, 3 або 4). Для кольорових зображень аналіз проводиться окремо для кожного каналу, хоча найчастіше використовується лише канал яскравості. Нехай $G = \{x_1, x_2, ..., x_n\}$ - група пікселів, де $x_i \in [0, 255]$ для 8-бітного зображення. Для аналізу використовуються маски $M_1 = [1, 0, 1, 0, ...]$ та $M_{-1} = [-1, 0, -1, 0, ...]$.

Застосування маски визначається як додавання або віднімання одиниці до відповідних пікселів: якщо елемент маски 1, піксель збільшується на 1; якщо -1 - зменшується; якщо 0 - залишається незмінним. Формально: $G' = \{x'_1, x'_2, ..., x'_n\}$, де $x'_i = \max(0, \min(255, x_i + M[i]))$. Наприклад, для $G = \{120, 85, 200, 73\}$ та $M_1 = [1, 0, 1, 0]$ отримуємо $G'_1 = \{121, 85, 201, 73\}$ див. рис. 1.8.

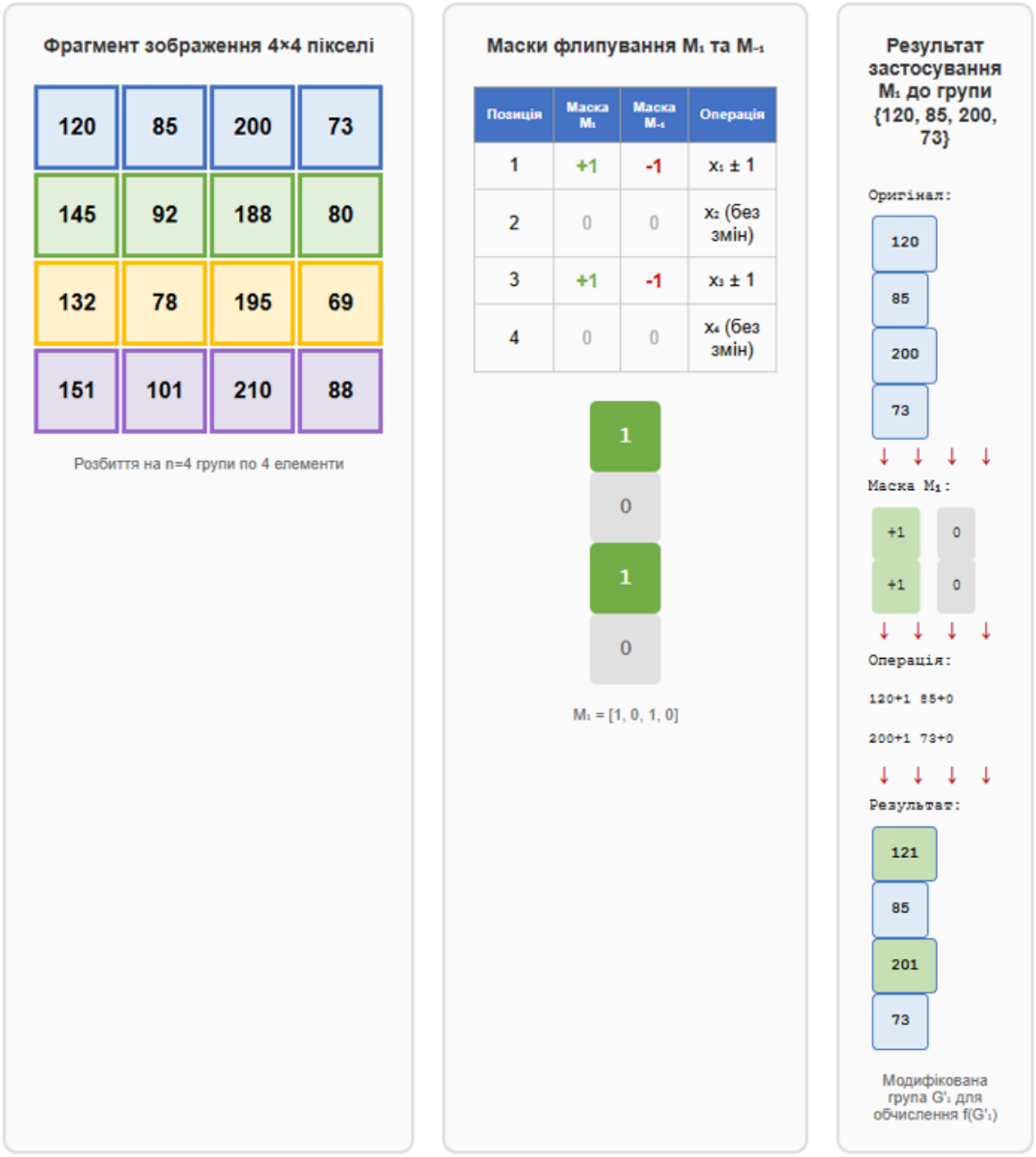


Рисунок 1.5 - Схема розбиття зображення на групи та застосування масок

Ключовим елементом є функція дискримінації f(G), що вимірює «гладкість» групи через суму абсолютних різниць між сусідніми пікселями: $f(G) = \Sigma|x_{i+1} - x_i|$ для i = 1 до n-1. Гладкі ділянки (небо, однорідні поверхні) мають малі значення f(G), текстуровані області - великі. Альтернативні функції можуть використовувати квадрати різниць $f(G) = \Sigma(x_{i+1} - x_i)^2$ або враховувати різниці вищих порядків.

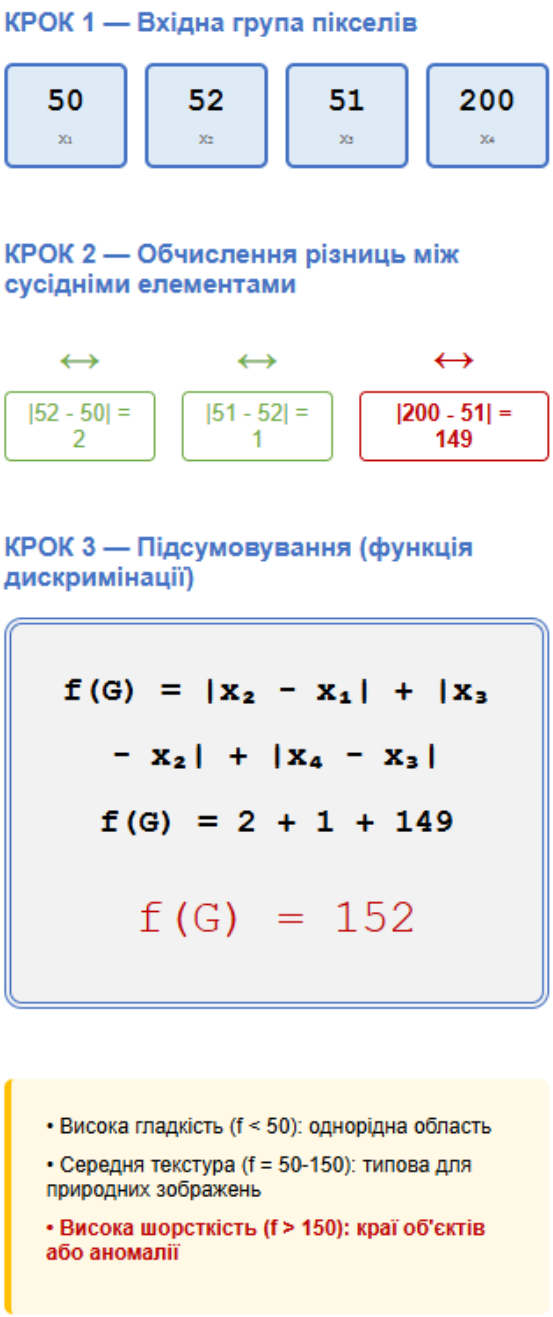


Рисунок 1.6 - Обчислення групи дискримінації для групи пікселів

Після застосування масок M_1 та M_{-1} до групи G отримуємо модифіковані групи G_1 та G_{-1}. Порівнюючи значення функції дискримінації, група класифікується у одну з трьох категорій: регулярна (R), якщо f(G_M) > f(G); сингулярна (S), якщо f(G_M) < f(G); незмінна (U), якщо f(G_M) = f(G). Для кожного зображення обчислюються чотири базові величини: R_M_1, S_M_1, R_M_{-1}, S_M_{-1}, які нормалізуються діленням на загальну кількість груп [41].

Фрідріх показав, що для природних зображень без стеганографії існує приблизна симетрія: $R\_M_1 \approx R\_M_{-1}$ та $S\_M_1 \approx S\_M_{-1}$.

Таблиця 1.4 - Класифікації груп в RS-аналізі

Група	f(G)	Маска M_1 застосована	f(G_M1)
Приклад 1: [100, 102, 101, 103]	10	[101, 102, 102, 103]	15
Приклад 2: [50, 200, 51, 199]	20	[51, 200, 52, 199]	12
Приклад 3: [80, 80, 81, 81]	8	[81, 80, 82, 81]	8
Група	f(G)	Маска M_1 застосована	f(G_M1)
Приклад 1: [100, 102, 101, 103]	10	[101, 102, 102, 103]	15
Приклад 2: [50, 200, 51, 199]	20	[51, 200, 52, 199]	12

Коли вбудовується LSB-повідомлення, це порушує природні кореляції та змінює співвідношення R/S. Теоретична модель базується на припущенні, що для частки p модифікованих пікселів співвідношення описується квадратичними рівняннями. Обчислюються дискримінанти $d_0 = R\_M - S\_M$ для оригіналу та d_1 для зображення з інвертованими LSB. Довжина повідомлення оцінюється як $p \approx d_0 / (d_0 - d_1)$, де p близьке до 0 вказує на чисте зображення, $p > 0.05$-0.1 - на можливе приховування, $p \approx 0.5$-1.0 - на насичене стегоповідомлення [7].

Алгоритм реалізації та оптимізації RS-аналізу

Практична реалізація включає вісім етапів. Крок 1: підготовка зображення через конвертацію у градації сірого ($Y = 0.299R + 0.587G + 0.114B$) та представлення як одновимірного масиву. Крок 2: формування груп розміром n=4, отримуючи $\lfloor (W \times H)/n \rfloor$ груп. Крок 3: ініціалізація лічильників $R\_M_1 = S\_M_1 = U\_M_1 = R\_M_{-1} = S\_M_{-1} = U\_M_{-1} = 0$. Крок 4: для кожної групи обчислюється f(G), застосовуються маски M_1 та M_{-1}, обчислюються $f(G_1)$ та $f(G_{-1})$, група класифікується відносно обох масок, лічильники інкрементуються. Крок 5: інверсія LSB всіх пікселів через XOR з 1 та повторення кроків 2-4. Крок 6: обчислення дискримінантів d_0 та d_1. Крок 7: оцінка p з обмеженням діапазоном [0, 1]. Крок 8: прийняття рішення на основі порогу.

У практичних реалізаціях застосовуються оптимізації для підвищення швидкості та точності. Вибірковий аналіз репрезентативних областей зменшує обчислювальну складність без суттєвої втрати точності. Вибір розміру групи n впливає на баланс між чутливістю та стабільністю: менші групи (n=2) дають більше вимірів але схильні до шуму; більші групи (n=5-6) стабільніші але пропускають локальні аномалії. Емпірично n=4 є оптимальним компромісом. Зважені функції дискримінації можуть підвищити чутливість на 10-15%. Важливим є обробка крайових випадків: пікселів зі значеннями 0 або 255, що можуть спотворювати статистику [8].

Сильні сторони RS-аналізу

RS-аналіз демонструє винятково високу точність при виявленні класичного LSB-вбудовування, надійно детектуючи приховування при заповненні лише 10-15% контейнера, що краще за візуальні методи (30-40%). Математична обґрунтованість забезпечує передбачувану поведінку: для повністю заповненого контейнера ($p \approx 1.0$) точність наближається до 100% навіть на високотекстурованих зображеннях.

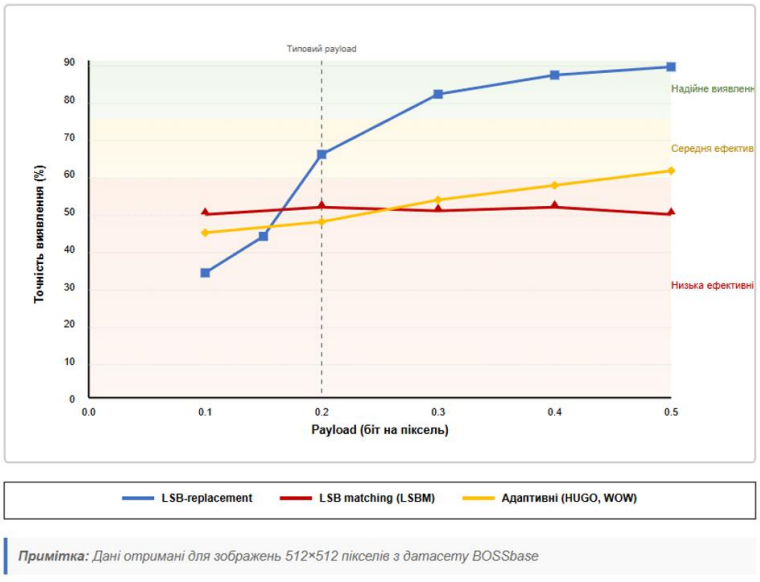


Рисунок 1.7 - Графік залежності точності RS-аналізу від обсягу вбудовування для різних типів LSB-стеганографії

Унікальною перевагою є кількісна оцінка відсотка модифікованих пікселів з похибкою ±5% абсолютних відсоткових пунктів для якісних зображень без

стиснення. Метод працює на широкому спектрі типів зображень (фотографії, графіка, скановані документи, медичні зображення), базуючись на універсальному принципі локальної кореляції пікселів. Дослідження на різноманітних датасетах показують стабільну точність з незначними варіаціями.

Алгоритм має лінійну складність O(N), де N - кількість пікселів, дозволяючи аналізувати стандартні зображення (1-2 мегапікселі) за частки секунди. Простота забезпечує ефективну паралелізацію на багатоядерних процесорах або GPU. RS-аналіз виявляє статистичні аномалії незалежно від змісту повідомлення: текстові документи, зашифровані файли або стиснуті дані з характеристиками випадкової послідовності детектуються однаково ефективно [9].

Таблиця 1.5 - Сильні та слабкі сторони RS-аналізу

Характеристика	Сильні сторони	Слабкі сторони
Точність проти LSB-replacement	✓ Висока (90-98% при payload >0.2 bpp)	X Неефективність проти LSB matching (~50%)
Оцінка payload	✓ Кількісна оцінка з похибкою ±5%	X Ненадійна для адаптивних методів
Стійкість до типу зображення	✓ Універсальна (фото, графіка, скани)	X Проблеми з крайовими значеннями (0, 255)
Обчислювальна складність	✓ Низька O(N), ~50мс для 1 Мпікс	-
Формат зображень	✓ Оптимальна для BMP, PNG	X Непридатна для JPEG (20-30% FPR)
Постобробка	-	X Чутлива до шуму, фільтрації, ресайзу
Інтерпретованість	✓ Математично обґрунтована, зрозумілі метрики	-

Слабкі сторони та обмеження RS-аналізу

Найсуттєвішим обмеженням є неефективність проти LSB matching (LSBM), де замість прямої заміни LSB застосовується випадкове збільшення або зменшення значення пікселя. LSBM зберігає симетрію статистичних властивостей, тому

точність виявлення падає до рівня випадкового вгадування навіть при повному заповненні. Похідні методи (±k matching, матричне кодування) також успішно обходять RS-детектування.

RS-аналіз розроблявся для зображень без втратного стиснення (BMP, PNG). На JPEG-зображеннях з коефіцієнтом якості нижче 90 частка хибних спрацювань може перевищувати 20-30%. Для якості 70-80 (стандарт для веб) метод практично непридатний через неможливість відрізнити артефакти стиснення від стеганографії. Повторне JPEG-збереження після LSB-вбудовування знищує повідомлення та робить детектування неможливим.

Різноманітні операції постобробки (додавання шуму, фільтрація, зміна розміру, обрізка) модифікують LSB, призводячи до хибних результатів. Особливо проблематичний гаусівський шум (σ=1-2), що створює ілюзію стеганографії. Зміна розміру з інтерполяцією призводить до псевдовипадкових LSB після округлення, маскуючи реальну стеганографію або створюючи хибні тривоги.

Пікселі зі значеннями 0 або 255 порушують симетрію, на якій базується метод. Зображення з понад 15-20% крайових значень демонструють аномальну статистику навіть без стеганографії, знижуючи точність на 25-40%. Виключення таких груп зменшує кількість вимірів та може призвести до статистично незначущих результатів.

RS-аналіз не враховує адаптивність вбудовування - характеристику сучасних алгоритмів (HUGO, WOW, S-UNIWARD), що концентрують зміни у складних текстурованих областях. Передбачаючи рівномірний розподіл змін, метод не має інструментів для виявлення локалізованого вбудовування. Хоча RS-аналіз менш схильний до cover-source mismatch порівняно з машинним навчанням, пороги оптимальні для одного датасету можуть давати підвищену частку хибних спрацювань на іншому.

Практичні застосування та модифікації RS-аналізу

Незважаючи на обмеження, RS-аналіз знаходить застосування в криміналістиці як частина комплексного аналізу підозрілих зображень без стиснення та у корпоративній безпеці у складі систем Data Loss Prevention для

моніторингу витоку інформації. Дослідники запропонували модифікації для підвищення ефективності: Weighted RS з ваговими коефіцієнтами для різних типів груп; Multi-dimensional RS з кількома наборами масок одночасно; Calibrated RS, що порівнює статистики оригіналу зі статистиками після операції знищення стеганографії; Extended RS, що комбінує принципи RS з Sample Pair Analysis для чутливості до LSB matching; Трихромний RS-аналіз з окремим аналізом RGB-каналів та мажоритарним голосуванням.

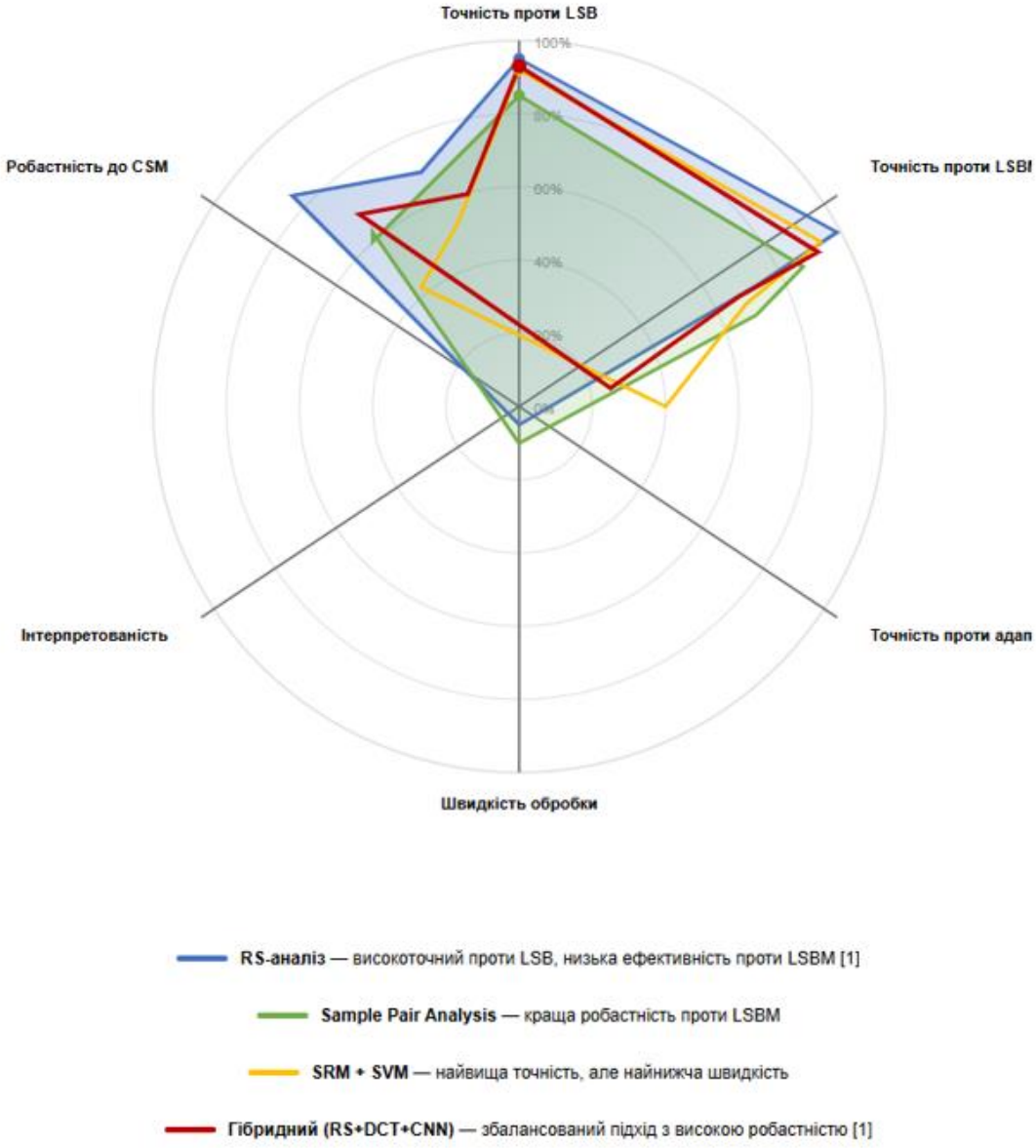


Рисунок 1.8 - Радарна діаграма методів

Сучасні гібридні підходи поєднують RS-аналіз з глибоким навчанням, використовуючи RS-статистики як додаткові ознаки для нейромережевого класифікатора, зберігаючи математичну інтерпретованість та компенсуючи слабкості через адаптивність нейромережі [43].

1.3. Аналіз методів стегоаналізу у частотній області: аналіз DCT-коефіцієнтів

Основи дискретного косинусного перетворення та формат JPEG

Дискретне косинусне перетворення (DCT) є фундаментальним інструментом обробки та стиснення цифрових зображень. На відміну від просторової області, частотна область описує зображення через набір частотних компонентів від низькочастотних (плавні переходи, загальна структура) до високочастотних (деталі, краї, текстури). Двовимірне DCT для блоку N×N визначається формулою: $F(u,v) = (2/N)\cdot C(u)\cdot C(v)\cdot \Sigma\Sigma f(x,y)\cdot \cos[(2x+1)u\pi/(2N)]\cdot \cos[(2y+1)v\pi/(2N)]$, де f(x,y) - значення пікселя, F(u,v) - DCT-коефіцієнт, $C(u)=1/\sqrt{2}$ для u=0 та C(u)=1 для u>0. Коефіцієнт F(0,0) називається DC-коефіцієнтом (середнє значення яскравості блоку); решта F(u,v) є AC-коефіцієнтами. Ключова властивість DCT - концентрація енергії у низькочастотних коефіцієнтах, що є основою JPEG-стиснення [44].

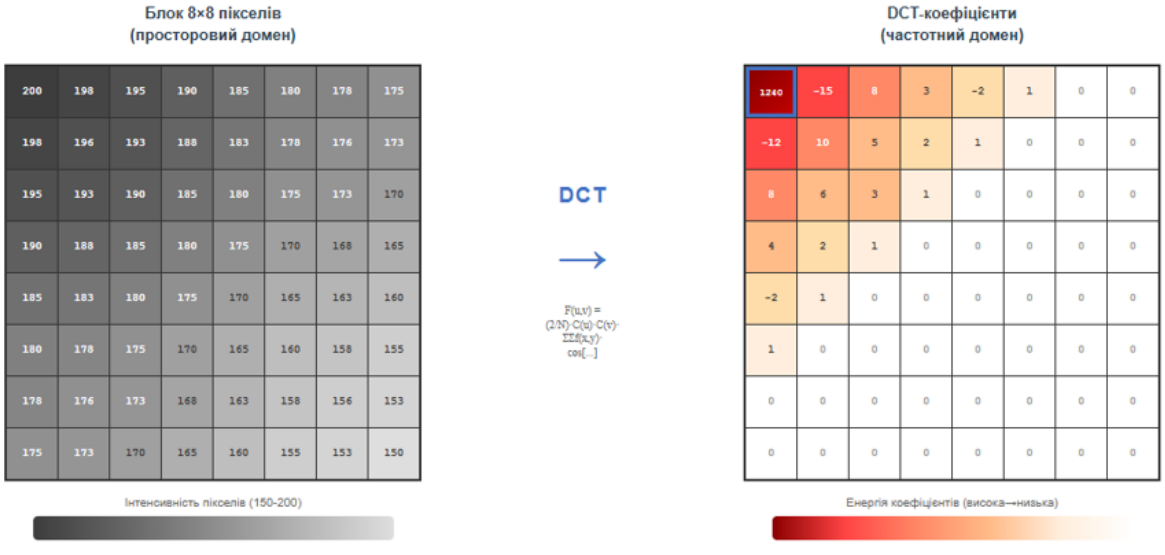


Рисунок 1.9 - Візуалізація DCT-перетворення

Стандарт JPEG використовує DCT для втратного стиснення через конвертацію RGB→YCbCr, розбиття на блоки 8×8, застосування DCT, квантування

та ентропійне кодування. Квантування є критичним етапом: F_quantized(u,v) = round(F(u,v)/Q(u,v)), де матриця Q містить менші значення для низьких частот (точніше збереження) та більші для високих (більша похибка). Параметр якості (1-100) масштабує Q: вища якість означає менше квантування. При декодуванні відбувається розквантування, зворотне DCT та реконструкція; оскільки квантування необоротне, виникають артефакти.

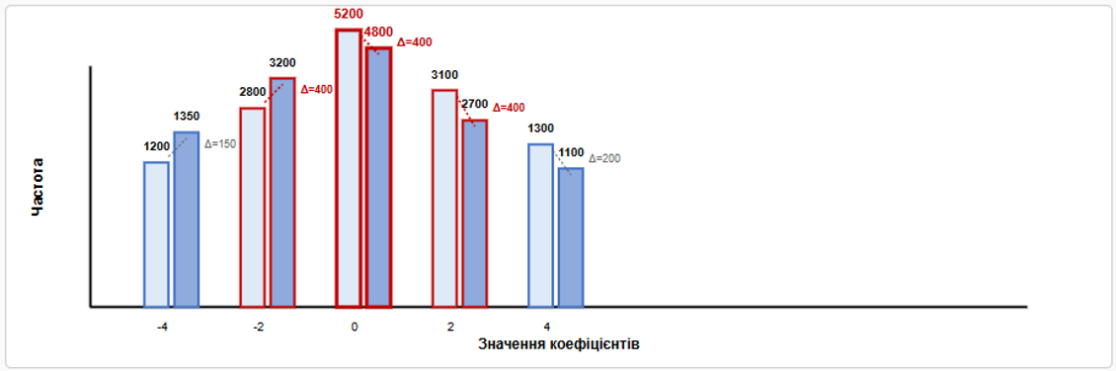
Рисунок 1.10 - Етапи JPEG-стиснення

JPEG став популярною мішенню стеганографії (понад 70% інтернет-зображень). Найпростіший підхід JSteg застосовує LSB-заміщення до ненульових квантованих AC-коефіцієнтів у зигзаг-порядку. F5 використовує зменшення абсолютного значення коефіцієнта замість прямої зміни LSB плюс матричне кодування. OutGuess модифікує невикористані коефіцієнти для відновлення глобальної статистики після вбудовування. Сучасні адаптивні методи (J-UNIWARD, UED) використовують моделі спотворень, надаючи перевагу складним блокам з багатьма ненульовими коефіцієнтами [10].

Стегоаналіз DCT-коефіцієнтів: гістограмний аналіз та χ^2-атак

Квантовані DCT-коефіцієнти природних JPEG-зображень демонструють характерні закономірності. Розподіл AC-коефіцієнтів наближається до узагальненого розподілу Гауса/Лапласа з високим піком у нулі та швидким спаданням. Низькочастотні коефіцієнти мають ширший розподіл, високочастотні - вужчий з концентрацією у малих значеннях. Характерна симетрія: $P(k) \approx P(-k)$. Гістограма демонструє плавні переходи; раптові стрибки або асиметрії вказують на втручання.

Фундаментальним є поняття пар значень (PoVs). При LSB-вбудовуванні кожна пара {2k,2k+1} модифікується, призводячи до вирівнювання частот: H'(2k)≈H'(2k+1). Міра нерівності Δ(k)=|H(2k)-H(2k+1)| зменшується після стеганографії. Візуальний аналіз виявляє ефект «сходинок» при JSteg (пилкоподібна гістограма), асиметричні деформації при F5 (надлишок нулів, аномальне H(0)/H(1)), локальні аномалії при OutGuess [11].



Пара	H(2k)	H(2k+1)	Різниця
{-4,-3}	1200	1350	150
{-2,-1}	2800	3200	400
{0,1}	5200	4800	400
{2,3}	3100	2700	400
{4,5}	1300	1100	200

Рисунко 1.11 - Ефект парності значень

X^2-атака Вестфельда (1999) була першим ефективним статистичним методом виявлення JSteg. Критерій $\chi^2=\Sigma[(O\_i-E\_i)^2/E\_i]$ вимірює відхилення спостережуваного розподілу від очікуваного.

Пара {2k, 2k+1}	Спостережувані частоти	Очікувані частоти	Обчислення χ^2_pair	Результат
{0, 1}	H(0)=500 H(1)=480	E(0)=E(1)= (500+480)/2= 490	χ^2=[(500-490)²/490]+ [(480-490)²/490]= [100/490]+[100/490]	0.408
{2, 3}	H(2)=120 H(3)=110	E(2)=E(3)= (120+110)/2= 115	χ^2=[(120-115)²/115]+ [(110-115)²/115]= [25/115]+[25/115]	0.435
{-2, -1}	H(-2)=95 H(-1)=105	E(-2)=E(-1)= (95+105)/2= 100	χ^2=[(95-100)²/100]+ [(105-100)²/100]= [25/100]+[25/100]	0.500
СУМА:	-	-	χ^2_total = Σ χ^2_pair	1.343

Рисунок 1.11 - Обчислення χ^2-статистики

Для пари {2k,2k+1} очікувана частота після повного LSB-вбудовування: E(2k)=E(2k+1)=[H(2k)+H(2k+1)]/2. Реалізація включає: декодування JPEG та екстракцію DCT-коефіцієнтів; побудову гістограми AC-коефіцієнтів; формування пар та обчислення очікуваних частот; обчислення χ^2_total=Σχ^2_pair; визначення p-

value через функцію розподілу χ^2; інкрементальну оцінку довжини повідомлення через точку мінімуму χ^2.

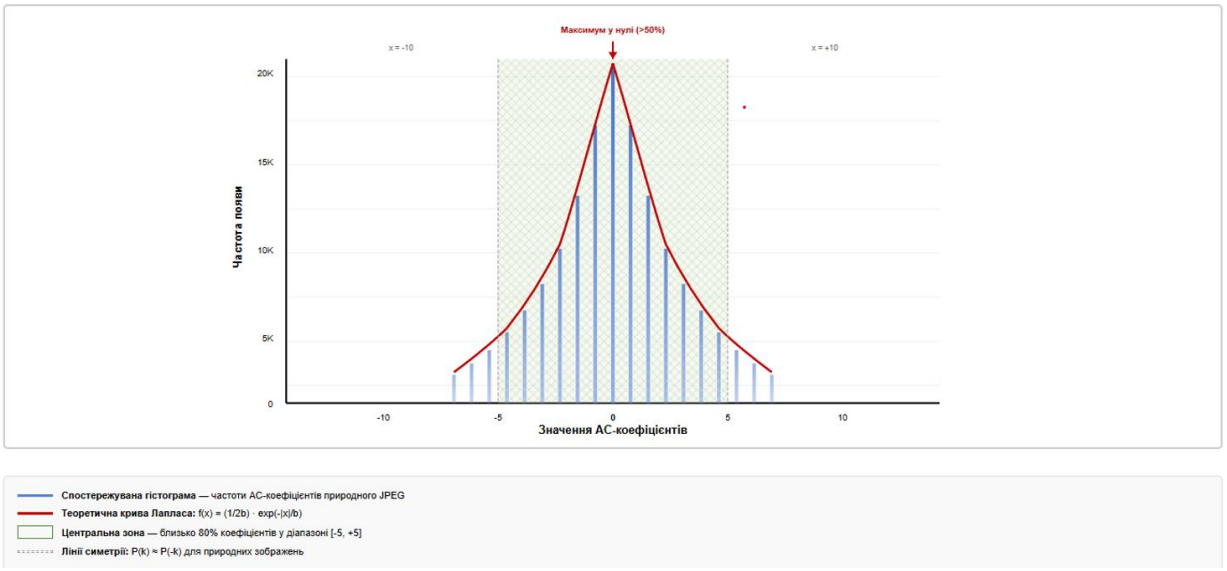


Рисунок 1.12 - Гістограма AC-коефіцієнтів

X^2-атака демонструє високу ефективність проти JSteg (виявлення при 5-10% заповнення, 100% точності для повного контейнера), можливість кількісної оцінки обсягу даних (похибка ±10-15%), обчислювальну ефективність без потреби навчальних даних. Проте обмеження включають специфічність до простого LSB (неефективність проти F5, OutGuess), вразливість до повторного стиснення з іншим коефіцієнтом якості, неспроможність виявити локалізоване вбудовування адаптивних методів [12,42].

Розширені методи аналізу DCT-коефіцієнтів

Калібраційний стегоаналіз Фрідріха-Голяна створює калібраційне зображення (версію оригіналу без стеганографії) через повторне JPEG-стиснення після обрізки на 4 пікселі з країв, що зміщує сітку блоків 8×8. Якщо оригінал чистий, статистики близькі до каліброваних; якщо містить стеганографію, статистики відрізняються (калібрація знищила дані). Відстань між векторами ознак (Евклідова, KL-дивергенція) є мірою ймовірності стеганографії. Ефективно проти OutGuess, але вимагає додаткових обчислень [13].

Блоковий аналіз та детектування двократного стиснення виявляють періодичні аномалії у гістограмі DCT-коефіцієнтів (метод Попеску-Фаріда), ефекти

блокування на межах блоків 8×8, періодичні компоненти з періодом 8 пікселів у спектрі потужності. Марківські моделі описують ймовірнісні залежності P(F_j|F_i) між коефіцієнтами (інтра-блокові, інтер-блокові, багатовимірні); стеганографія порушує кореляції. Після побудови матриць переходів застосовується класифікатор (SVM, CNN). Високоточні, але обчислювально ресурсоємні.

Таблиця 1.6 Методи DCT-стегоаналізу

Метод	Рік	Тип	Ознаки	Точність проти JSteg	Точність проти F5	Точність проти J-UNIWARD	Обмеження
χ^2-тест	1999	Статистичний	χ^2 пар значень	~100%	60-70%	~50%	Компенсаційні методи
Калібраційний аналіз	2003	Статистичний	KL-дивергенція	85-90%	75-85%	65-70%	Вимагає повторне стиснення
DCTR + SVM	2009	Ручні ознаки	~8,000	92-95%	85-90%	70-75%	Висока обчислювальна складність
GFR + EC	2011	Ручні ознаки	Gabor residuals	90-93%	82-88%	68-73%	Cover-source mismatch
J-Net (CNN)	2018	Глибоке навчання	Автоматичні	95-98%	90-93%	80-85%	Потреба у великих даних

Сучасний підхід використовує великі набори ознак з машинним навчанням. DCTR (8000 ознак) обчислює залишки після високочастотних фільтрів та будує марківські моделі. GFR застосовує фільтри Габора до просторового зображення, аналізуючи високочастотні компоненти. CC-JRM (22000+ ознак) враховує міжканальні кореляції для кольорових JPEG. З ансамблевими класифікаторами досягається 75-90% точності на BOSSbase/Alaska проти адаптивних методів [14].

Обмеження DCT-стегоаналізу та напрямки вдосконалення

Ефективність DCT-стегоаналізу сильно залежить від коефіцієнта якості JPEG. При дуже високій якості (95-100) мінімальне квантування наближає ситуацію до просторової області; при низькій (<50) агресивне квантування призводить до обмеженої статистичної варіативності. Різні камери створюють JPEG з відмінними властивостями через різні реалізації кодера, матриці квантування, постобробку (cover-source mismatch). Програмні кодери (libjpeg, Adobe JPEG) також створюють відмінності.

Постобробка (обрізка, зміна розміру, фільтри) з повторним збереженням у JPEG знищує як приховані дані, так і статистичні сигнатури. Конвертація JPEG→PNG→JPEG створює складні артефакти. Сучасні адаптивні методи (J-UNIWARD, UED) не створюють очевидних артефактів вирівнювання пар; навіть найкращі детектори досягають лише 65-75% точності при малих payload. Методи на основі великих наборів ознак працюють як «чорні скриньки» без пояснення причин класифікації, ускладнюючи судову експертизу.

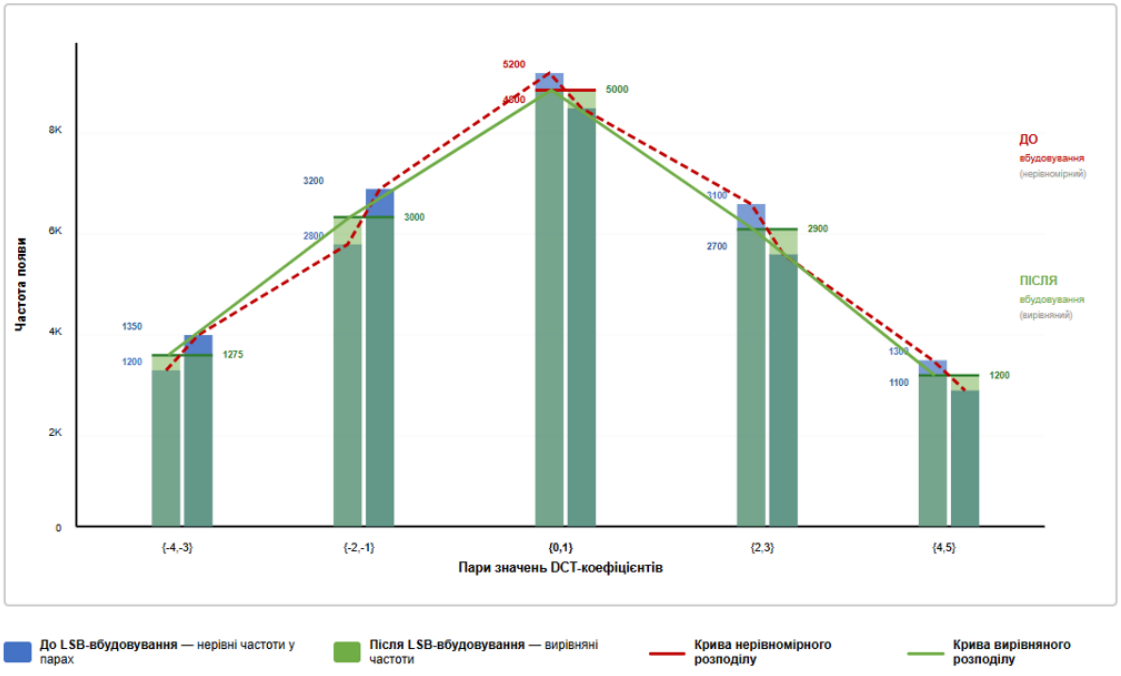


Рисунок 1.13 - Вирівнювання пар значень

Актуальні напрямки вдосконалення включають глибоке навчання з CNN для обробки DCT-коефіцієнтів (архітектури J-Net для JPEG-домену); гібридні методи, що комбінують статистичний аналіз (χ^2, калібрація) з нейромережевою класифікацією для поєднання інтерпретованості та точності (тема цієї магістерської роботи); transfer learning для адаптації моделей між типами зображень, зменшуючи cover-source mismatch; просторово-частотний аналіз через вейвлет-перетворення для компенсації слабкостей окремих областей [15,45].

1.4. Методи стегоаналізу на основі глибокого навчання: згорткові нейронні мережі

Еволюція від ручного проектування ознак до глибокого навчання

Протягом перших двох десятиліть стегоаналізу домінував парадигма ручного проектування ознак. Дослідники аналізували статистичні властивості зображень та формалізували їх у числові дескриптори: SPAM (686 ознак), SRM (34671 ознака), DCTR (близько 8000 ознак для JPEG). Проте цей підхід має фундаментальні обмеження: по-перше, вимагає глибокої експертизи для передбачення індикативних характеристик; по-друге, кожен новий стеганографічний метод потребує нових ознак; по-третє, ручні ознаки є компромісами між ефективністю та повнотою інформації. Навіть найбагатші набори ознак захоплюють лише обмежену проекцію високовимірного простору зображень, залишаючи масу потенційно корисних патернів неформалізованими [16].

Глибокі нейронні мережі, зокрема CNN, пропонують радикально інший підхід: автоматичне виділення ієрархічних ознак безпосередньо з вхідних даних. Замість ручного визначення фільтрів мережа навчається оптимальним фільтрам у процесі тренування. Перші шари виділяють низькорівневі ознаки (краї, градієнти, текстури), глибші комбінують їх у складні абстрактні концепції. Ключова перевага - адаптивність: одна архітектура CNN тренується на різні типи стеганографії через зміну навчальної вибірки без ручного редизайну. CNN виявляє складні мультимодальні патерни, недоступні лінійним класифікаторам.

Таблиця 1.7 - Еволюція підходів

Рік	Метод	Кількість ознак	Точність (S-UNIWARD 0.4)	Час обробки (1 Мпікс)	Навчальні дані
2001	RS-аналіз	~10	65%	50 мс	Не потрібні
2006	SPAM	686	75%	1.2 с	~5,000
2012	SRM + EC	34,671	88%	12 с	~10,000
2016	Xu-Net	~256 (внутр.)	80%	150 мс (GPU)	~50,000
2018	SRNet	~512 (внутр.)	90%	200 мс (GPU)	~80,000
2025	Гібридний	40	88-92% (очікувано)	100 мс	~20,000

Застосування глибокого навчання почалося у середині 2010-х з робіт Tan і Li (2014), але пряме перенесення архітектур з комп'ютерного зору (AlexNet, VGG) виявилося неефективним. Стеганографічні зміни значно тонші за візуальні відмінності об'єктів, тому потрібні спеціалізовані архітектури. Стегоаналітичні CNN мають унікальні характеристики: препроцесингові шари з фіксованими високочастотними фільтрами (HPF) виділяють залишки, де концентрується стеганографічний сигнал; обмеження ваг та спеціалізована ініціалізація для стабілізації навчання; absolute activation або TanH замість ReLU для збереження інформації з обох напівплощин; глобальне усереднення замість повнозв'язних шарів; агресивна регуляризація (dropout, batch normalization, weight decay) через малість стеганографічного сигналу [17,46].

Ключові архітектури: Xu-Net, SRNet, Yedroudj-Net

Xu-Net (Сю та ін., 2016) стала першою спеціалізованою CNN-архітектурою, що продемонструвала конкурентоспроможність з традиційними методами. Складається з п'яти блоків: препроцесинг через 30 фіксованих SRM-фільтрів 5×5; три згорткові блоки з функціями Absolute value та Batch Normalization, кількість каналів зростає від 30 до 128; глобальне усереднення та класифікація через softmax.

Показала 75-82% точності на BOSSbase проти S-UNIWARD/WOW (0.4 bpp), що порівняно з SRM+EC. Концептуальне підтвердження: CNN автоматично виявила структури, подібні до ручних SRM-фільтрів, плюс нові патерни. Стала базовою архітектурою для подальших вдосконалень.

Таблиця 1.8 - Порівняння архітектур Xu-Net та SRNet

Характеристика	Xu-Net (2016)	SRNet (2018)
Кількість блоків	5	12
Функція активації	Abs + ReLU	TLU (Truncated Linear)
Residual connections	Ні	Так (skip connections)
Кількість параметрів	~500,000	~6,000,000
Глибина мережі	Середня	Глибока
Час навчання (епоха)	60 хв	120 хв
Точність (BOSSbase, S-UNIWARD 0.4)	80-82%	88-90%

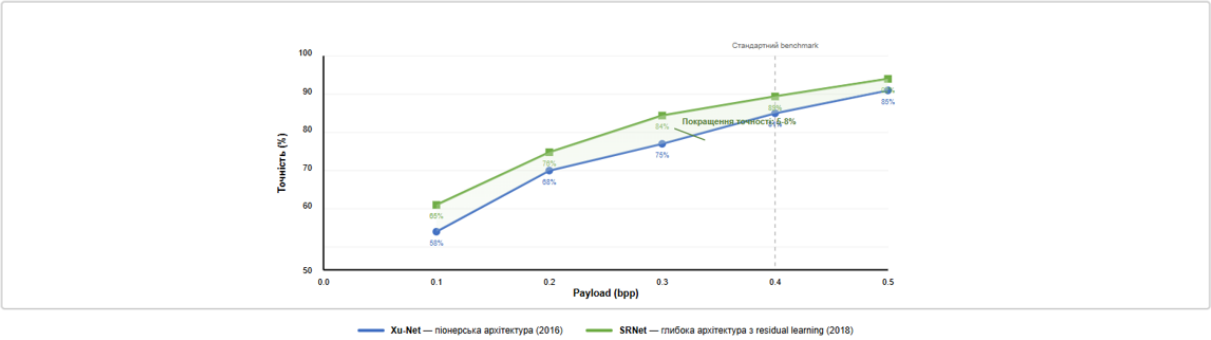


Рисунок 1.14 - Порівняння Xu-Net та SRNet

SRNet (Boroumand та ін., 2018) застосувала навчання через пропуск під'єднань для навчання глибших мереж (12 блоків проти 5 у Xu-Net). Архітектура: 30 SRM-фільтрів; чотири шари з TLU активацією (Truncated Linear Unit з поступово зменшуваним порогом T=3→1); сім residual блоків зі зростанням каналів 64→512; глобальне усереднення [47]. Навчається на патчах 256×256, оптимізатор Adam (α=0.001), L2 регуляризація (λ=0.0001), dropout (p=0.5), аугментація через flip/rotation. Показала 85-90% точності на BOSSbase проти S-UNIWARD (0.4 bpp), на 5-8 відсоткових пунктів краще за Xu-Net; для JPEG-стегоаналізу 80-85% проти

J-UNIWARD порівняно з 65-75% для DCTR/GFR. Аналіз помилок: більшість false negatives на високотекстурованих зображеннях; false positives на зображеннях після агресивної постобробки. Візуалізація показала, що мережа фокусується на текстурованих ділянках, уникаючи гладких областей.

Yedroudj-Net (2018) оптимізує ефективність замість глибини: 6 блоків замість 12, depth-wise separable convolutions, гібридний препроцесинг (30 фіксованих + 30 навчених фільтрів). Містить 2.5M параметрів проти 6M+ у SRNet, що дає швидше навчання (у 1.5-2 рази), менші вимоги пам'яті, зменшений ризик переобладнання. Точність 82-87% на BOSSbase проти S-UNIWARD (0.4 bpp), лише на 2-3 відсоткові пункти нижче за SRNet при істотно менших обчислювальних витратах. Варіації: Yedroudj-Net-Color для RGB через cross-channel convolutions; Yedroudj-Net-JPEG для DCT-коефіцієнтів; Lightweight версія для embedded-систем [18,48].

Проблема Cover-Source Mismatch та стратегії мітигації

Cover-Source Mismatch (CSM) є критичною проблемою CNN-стегоаналізу: детектор, навчений на зображеннях з певного джерела (Canon EOS 5D), демонструє значне зниження точності на іншому джерелі (Nikon D90, iPhone, сканер), навіть при тому самому методі стеганографії. Причини багатофакторні: апаратні відмінності (сенсори, оптика, процесори), алгоритмічні відмінності (demosaicing, шумопридушення), статистичні відмінності (розподіли текстур, кольорів) [49]. CNN вивчає не лише стеганографічні артефакти, але й характеристики навчальної вибірки, помилково асоціюючи специфічні властивості з стеганографією [19].

Емпіричні дослідження демонструють драматичний вплив: SRNet, натренована на BOSSbase (Canon 1D Mark III), досягає 88% точності на тестовій множині з того ж джерела, але падає до 65-70% на BOWS2 (різні камери) та до 50-55% на скандованих документах/CGI-зображеннях - рівень випадкового вгадування. Особливо проблематичні медичні зображення (рентген, МРТ), супутникові знімки, мікроскопічні фотографії.

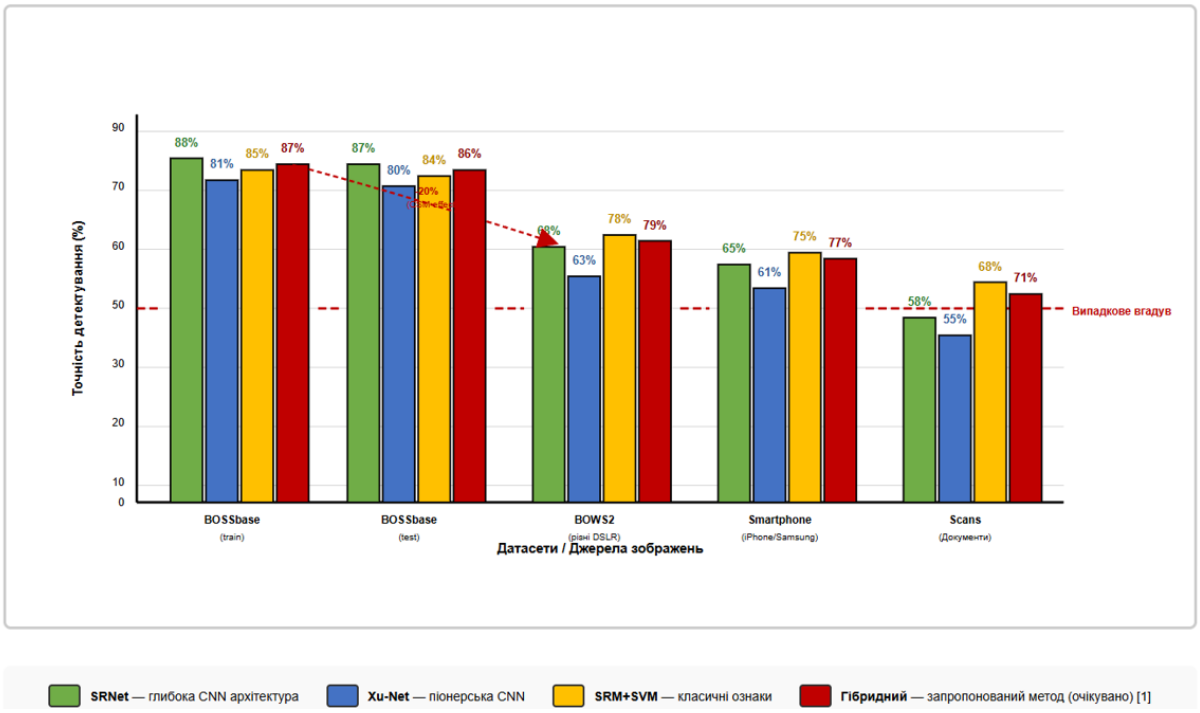


Рисунок 1.15- Cover-Source Mismatch

Стратегії мітигації включають: диверсифікацію навчальних даних (dataset ALASKA2 з 80000+ зображень різних джерел); domain adaptation через fine-tuning останніх шарів або adversarial domain adaptation; data augmentation (зміна яскравості, контрасту, шуму) при обережності щодо знищення стеганографічного сигналу; ensemble methods для компенсації різних bias. Незважаючи на прогрес, CSM залишається відкритою проблемою: жоден підхід не гарантує повної інваріантності без втрати точності на знайомих доменах. Для критичних застосувань рекомендується multiple-model approach з попередньою класифікацією джерела [20].

Таблиця 1.9 - Детальне порівняння CNN-архітектур

Архітектура	Рік	Препроцесинг	Кількість блоків	Параметри	Activation	Residual	Точність (BOSSbase)	Час інференсу
Xu-Net	2016	30 SRM (fixed)	5	500K	Abs + ReLU	Hi	80-82%	50 мс
Ye-Net	2017	Learned filters	6	600K	TanH	Hi	78-80%	45 мс
SRNet	2018	30 SRM (fixed)	12	6M	TLU	Так	88-90%	120 мс

Продовження таблиці 1.9

Yedroudj-Net	2018	30 SRM + 30 learned	6	2.5M	ReLU	Hi	85-87%	70 мс
SE-SRNet	2020	30 SRM (fixed)	12	6.5M	TLU	Так + Attention	91-93%	150 мс

Рисунок 1.16 - Порівняння CNN-архітектур

Сучасні тренди та перспективні напрямки

Attention mechanisms (що революціонізували обробку природної мови через Transformers) застосовуються в стегоаналізі. Spatial attention фокусується на підозрілих областях з вищою ймовірністю стеганографії (корисно проти адаптивних методів з локалізованим вбудовуванням). Channel attention оптимізує вагу різних карт ознак, виділяючи найінформативніші частотні компоненти. Архітектури SE-SRNet (Squeeze-and-Excitation SRNet) демонструють покращення точності на 2-4% [50].

Adversarial training через GANs: генератор створює складну стеганографію мінімізуючи детектованість, дискримінатор навчається виявляти її. Змагальний режим призводить до co-evolution: генератор виявляє слабкості детектора, змушуючи його вдосконалюватися, результуючи у стійкішому детекторі [51].

Таблиця 1.10 - Обчислювальні вимоги (CPU/GPU)

Метод	CPU (Intel i7)	GPU (NVIDIA GTX 1080)	Пам'ять	Енергоспоживання	Придатність для real-time
RS-аналіз	45 мс	20 мс	50 MB	Низьке	Так (>20 fps)
χ^2-тест	80 мс	35 мс	80 MB	Низьке	Так (>12 fps)
SRM + SVM	15 с	-	2 GB	Середнє	Ні (<0.1 fps)
Xu-Net	2 с	60 мс	500 MB	Високе (GPU)	Так з GPU
SRNet	4 с	150 мс	1.2 GB	Високе (GPU)	Обмежено
Гібридний	300 мс	100 мс	200 MB	Середнє	Так (>3 fps)

Explainable AI (XAI) вирішує проблему «чорної скриньки»: saliency maps показують впливові пікселі; layer-wise relevance propagation (LRP) розподіляє передбачення назад визначаючи внесок кожного нейрона/пікселя; concept-based explanations виявляють високорівневі концепції (аномальний шум, порушена симетрія). Методи експериментальні, але критичні для судової експертизи.

Гібридні підходи поєднують CNN та традиційні методи: традиційні ознаки (SRM, DCTR) конкатенуються з output CNN перед класифікацією; багаторівневий аналіз через традиційні методи (RS-аналіз, χ^2-тест, DCT-аналіз) виділяє підозрілі характеристики для подальшого CNN-аналізу; ансамблі з різнорідних детекторів через голосування/зважене усереднення.

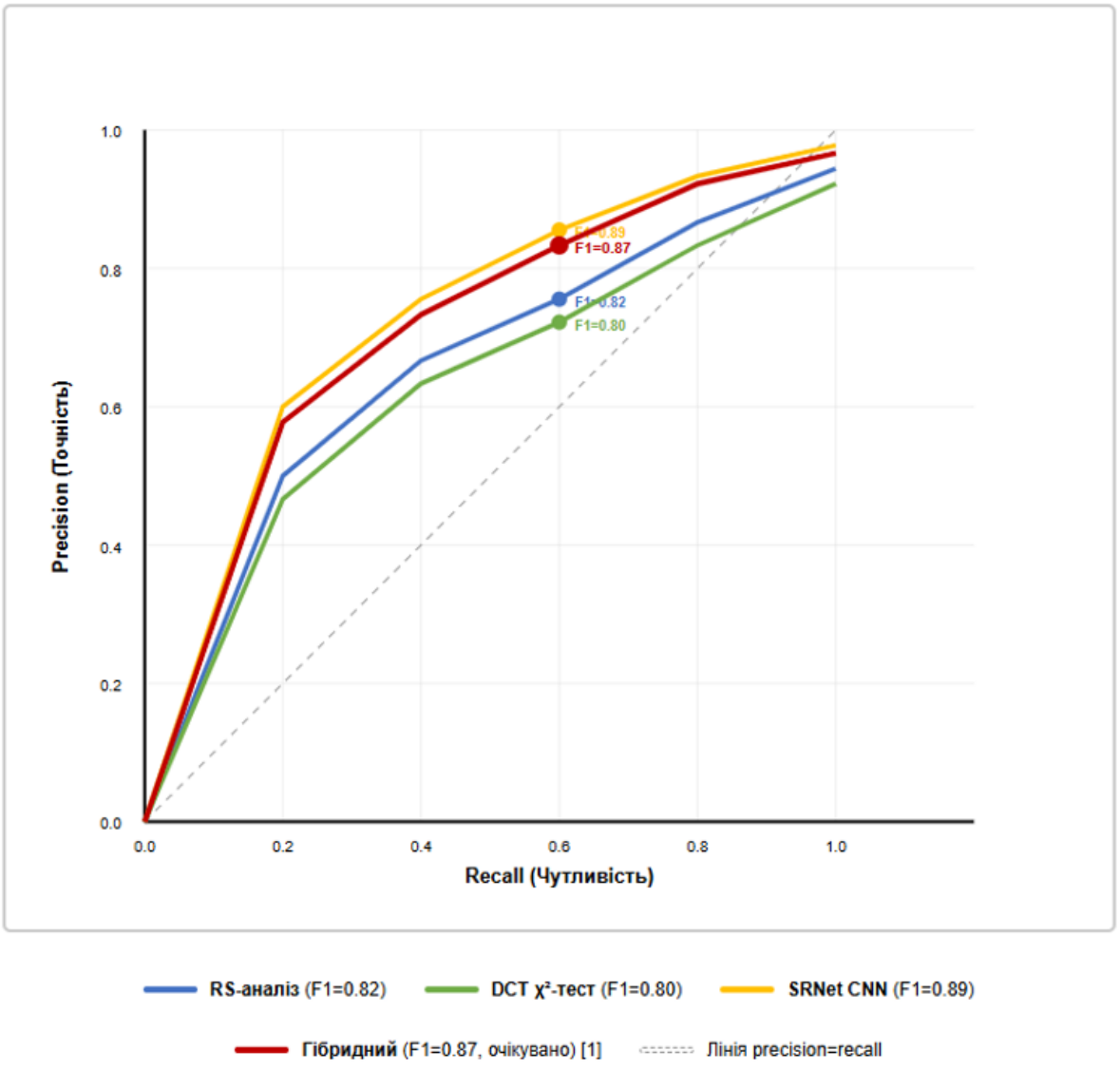


Рисунок 1.17 - Precision-Recall крива

Саме гібридний підхід (RS-аналіз + DCT-аналіз + CNN-класифікатор) є темою даної магістерської роботи: статистичні методи дають інтерпретовані стабільні сигнали, які CNN використовує для навчання надійніших узагальнюючих моделей, зменшуючи CSM та підвищуючи точність виявлення [21].

Висновки до розділу 1

У першому розділі виконано комплексний аналіз сучасних методів стегоаналізу цифрових зображень, що дозволило виявити ключові проблеми існуючих підходів та обґрунтувати необхідність розробки гібридних методів виявлення прихованої інформації.

Проведена класифікація методів стеганографії та стегоаналізу продемонструвала еволюцію від простих методів LSB-заміщення до складних адаптивних алгоритмів (HUGO, S-UNIWARD, WOW), що використовують моделі спотворень та синдромне кодування, роблячи статистичні сліди практично невідрізненими від природних варіацій. Систематизація методів стегоаналізу за доменом обробки, рівнем апріорних знань та підходом до виділення ознак виявила, що жоден окремий метод не є універсальним проти всього спектру стеганографічних загроз.

Детальний аналіз RS-методу виявив його переваги: високу точність виявлення класичного LSB-вбудовування (понад 90% при заповненні понад 15-20%), кількісну оцінку обсягу даних з похибкою ±5%, математичну обґрунтованість та обчислювальну ефективність O(N). Критичні обмеження включають: повну неефективність проти LSB matching та адаптивних методів; непридатність для JPEG-зображень (20-30% хибних спрацювань при якості нижче 90); чутливість до постобробки; проблеми з крайовими значеннями пікселів (0 та 255) при наявності понад 15%. RS-аналіз залишається ефективним для простої LSB-стеганографії у некомпресованих зображеннях, але потребує доповнення іншими методами.

Аналіз методів частотної області на прикладі DCT-коефіцієнтів показав їхню релевантність через домінування JPEG (понад 70% інтернет-зображень). Класичний χ^2-тест Вестфельда демонструє близько 100% точності проти JSteg при

заповненні понад 10% з оцінкою довжини повідомлення (похибка ±10-15%). Фундаментальні вразливості: неефективність проти F5, повна нейтралізація компенсаційними методами (OutGuess), неспроможність виявити адаптивні методи J-UNIWARD (точність 65-75%). Розширені методи з великими наборами ознак (DCTR, GFR, CC-JRM: 8000-22000 дескрипторів) підвищують точність до 75-85%, проте працюють як «чорні скриньки» без інтерпретації причин класифікації.

Дослідження методів глибокого навчання продемонструвало революційний перехід від ручного проектування ознак до автоматичного виділення патернів. Xu-Net (2016) досягла 75-82% точності без експертного проектування тисяч ознак. SRNet (2018) з residual learning підвищила точність до 85-90% на просторовій стеганографії та 80-85% на JPEG. Yedroudj-Net досягла конкурентної точності (82-87%) з меншою складністю (2.5M параметрів проти 6M). Критична проблема Cover-Source Mismatch: детектори демонструють падіння точності на 15-25 відсоткових пунктів при зміні джерела (з 88% до 65-70%), а на нетипових зображеннях - до 50-55%. Стратегії мітігації CSM лише частково вирішують проблему.

Критичний аналіз виділяє фундаментальні обмеження кожної категорії: просторові методи (RS-аналіз) ефективні лише проти LSB-replacement, вразливі до адаптивних методів, непридатні для JPEG, забезпечують інтерпретованість але низьку універсальність; частотні методи (DCT-аналіз) орієнтовані на JPEG, ефективні проти простих методів але нейтралізуються компенсаційними алгоритмами, мають сильну залежність від параметрів стиснення; методи глибокого навчання (CNN) демонструють найвищу точність та автоматичне виділення ознак, але мають критичну вразливість до CSM, потребують великих датасетів, відсутність інтерпретованості.

Загальний недолік усіх підходів - однодоменність: методи оптимізовані для однієї області без використання синергетичного ефекту від комбінування інформації з різних доменів. Виявлені обмеження обґрунтовують необхідність гібридного методу, що поєднує: інтерпретованість та стабільність статистичних методів (RS та DCT-аналіз); адаптивність та узагальнюючу здатність CNN;

багатодоменний аналіз (просторова + частотна область); зменшення CSM через доменно-інваріантні статистичні ознаки як anchor features. Така архітектура компенсує слабкості компонентів їхніми сильними сторонами: статистичні методи забезпечать стабільні інтерпретовані ознаки навіть на незнайомих доменах, CNN навчиться складним нелінійним комбінаціям для підвищення точності виявлення адаптивних методів стеганографії. Результати створюють теоретичне підґрунтя для розробки у наступному розділі гібридного методу з покращеними характеристиками точності, універсальності та робастності до варіацій джерел зображень.

РОЗДІЛ 2. РОЗРОБКА ВДОСКОНАЛЕНОГО МЕТОДУ СТЕГОАНАЛІЗУ ЗОБРАЖЕНЬ У ПРОСТОРОВІЙ ТА ЧАСТОТНІЙ ОБЛАСТЯХ

2.1. Концептуальна модель та архітектура гібридної системи стегоаналізу

Методологічні передумови проектування гібридної системи

Результати критичного аналізу виявили фундаментальну закономірність: різні типи стеганографічних артефактів проявляються у різних доменах з різною інтенсивністю. Простий LSB-replacement створює чіткі статистичні сигнатури у просторовій області (порушення симетрії регулярних/сингулярних груп), які RS-аналіз ефективно детектує, але ці артефакти слабко виражені у частотній області. Навпаки, JSteg-стеганографія залишає характерні сліди у розподілі DCT-коефіцієнтів (вирівнювання пар значень), але після декодування у просторовий домен ці ознаки розмиваються. Адаптивні методи (S-UNIWARD, HUGO, J-UNIWARD) створюють складні розподілені модифікації у обох доменах як тонкі нелінійні відхилення від природних закономірностей.

Принцип комплементарності полягає у синергетичному використанні інформації з різних доменів: слабко виражене в одному домені може бути чітко детектоване в іншому. Математично формалізується через концепцію ортогональних ознакових просторів [52]. Якщо кореляція між F_spatial (вектор RS-статистик) та F_frequency (вектор DCT-статистик) низька (cor<0.3-0.4), їхнє об'єднання F_combined=[F_spatial,F_frequency] підвищує інформативність порівняно з кожним окремо.

Запропонована гібридна система реалізує багаторівневу архітектуру з поступовим підвищенням абстракції.

Рівень 1: доменне розділення - паралельний аналіз у просторовій та частотній областях без взаємного впливу.

Рівень 2: виділення первинних ознак через спеціалізовані статистичні методи (RS-аналіз, DCT-аналіз), що генерують інтерпретовані математично обґрунтовані ознаки.

Рівень 3: інтеграція та високорівнева класифікація - CNN навчається складним нелінійним комбінаціям ознак для фінального рішення. Така ієрархія

розділяє відповідальність: статистичні методи забезпечують стабільність та інтерпретованість, CNN - адаптивність та узагальнення.

Ефективність гібридного підходу обґрунтовується теорією інформації: традиційний метод максимізує $I(f(X);Y)$, гібридний - $I([f_1(X),f_2(X),...,f_n(X)];Y)$. Якщо ознаки частково незалежні, загальна інформація наближається до суми: $I([f_1,f_2];Y)\leq I(f_1;Y)+I(f_2;Y)$. З точки зору статистичного навчання гібридний підхід реалізує ensemble learning на рівні ознак, дозволяючи класифікатору виявити взаємодії між різними типами ознак. Додаткова перевага - зменшення cover-source mismatch: CNN навчається на абстрактних статистичних ознаках, менш залежних від конкретного джерела, підвищуючи узагальнюючу здатність. Статистичні методи виконують роль доменно-інваріантних перетворень, нормалізуючи вхідні дані [21].

Загальна архітектура гібридної системи

Гібридна система складається з п'яти функціональних модулів у послідовно-паралельній архітектурі.

Модуль 1 (препроцесинг): перевірка формату, конвертація колірного простору, нормалізація; для JPEG екстракція просторового представлення та DCT-коефіцієнтів, для інших форматів - просторового представлення з формуванням частотного через DCT блоків 8×8.

Модуль 2 (RS-модуль): паралельна обробка просторового представлення через розбиття на групи, застосування масок, обчислення дискримінації, класифікацію груп, генерацію вектора V_RS=[R_M1,S_M1,R_M-1,S_M-1,d_0,d_1,p_estimate,...] (10-20 елементів).

Модуль 3 (DCT-модуль): паралельна обробка частотного представлення через побудову гістограм, обчислення χ^2-статистик, аналіз асиметрій, калібрацію, генерацію вектора V_DCT=[χ^2,H(0)/H(1),skewness,kurtosis,calibration_distance,...] (15-30 елементів).

Модуль 4 (інтеграція): об'єднання V_combined=[V_RS,V_DCT] з нормалізацією (стандартизацією) та обчисленням кросс-доменних ознак (узгодженість, розбіжність сигналів).

Модуль 5 (CNN-класифікатор): навчена нейронна мережа виконує бінарну класифікацію, виявляючи оптимальні комбінації ознак.

Обробка відбувається за сценарієм:

Етап 1 - вхід та валідація (перевірка формату, розміру ≥256×256, конвертація RGB→Y=0.299R+0.587G+0.114B).

Етап 2 - паралельна обробка доменів: Потік A (просторовий, ~50-200мс) - RS-модуль обробляє матрицю пікселів через розбиття на групи по 4, маски [1,0,1,0]/[-1,0,-1,0], обчислення f(G) у трьох станах, підрахунок регулярних/сингулярних груп, інверсію LSB, генерацію статистик d_0,d_1,p. Потік B (частотний, ~100-300мс) - DCT-модуль екстрагує коефіцієнти (JPEG) або виконує DCT блоків 8×8, будує гістограму AC-коефіцієнтів, обчислює χ^2-статистику пар {2k,2k+1}, моментні характеристики, калібрацію.

Етап 3 - синхронізація та об'єднання: стандартизація z_i=(x_i-μ_i)/σ_i для нульового середнього та одиничної дисперсії.

Етап 4 - класифікація: forward pass через CNN, обчислення P(stego|V_combined), порогове рішення (зазвичай 0.5).

Етап 5 - вихід: бінарне рішення {CLEAN,STEGO}, ймовірність P∈[0,1], внесок доменів, оцінка payload.

Система підтримує режими: повний (обидва модулі для максимальної точності), спрощений (лише RS-модуль з доповненням нулями DCT-позицій), експертний (ручні параметри), пакетний (паралельна обробка множини зображень) [22].

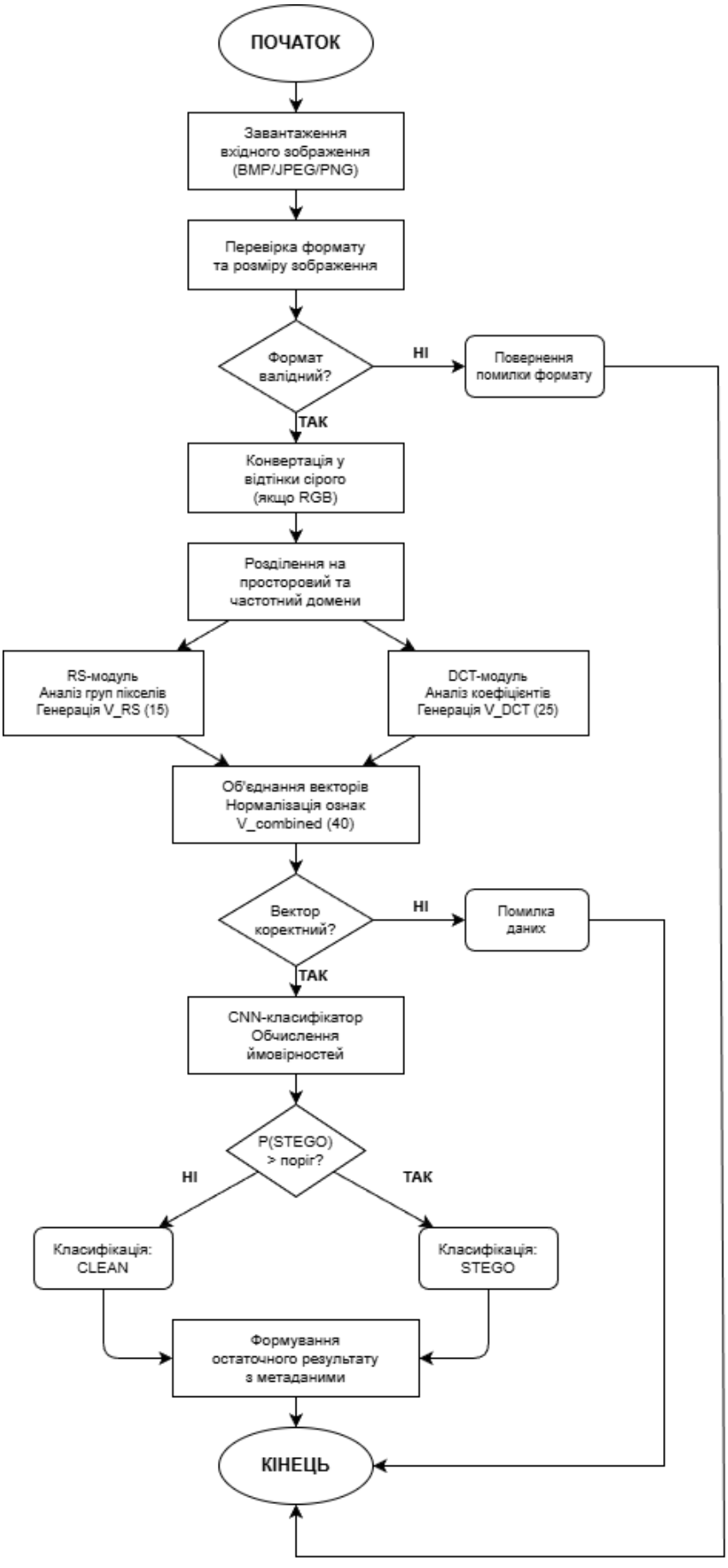


Рисунок 2.1 - Загальний алгоритм роботи системи

Математична формалізація та властивості моделі

Задача стегоаналізу формалізується як бінарна класифікація: знайти функцію $h:X\to Y$, що мінімізує $P(error)=P(h(x)\neq y|x)$, де X - простір зображень, $Y=\{0,1\}$ - мітки класів. Класичний підхід: $h(x)=g(f(x))$, де $f:X\to\mathbb{R}^n$ - виділення ознак, $g:\mathbb{R}^n\to Y$ - класифікація. Гібридна модель: $f_{hybrid}(x)=[f_{RS}(x_{spatial}),f_{DCT}(x_{frequency}),f_{cross}(x_{spatial},x_{frequency})]$, де $f_{RS}:\mathbb{R}^{(m\times n)}\to\mathbb{R}^{k_1}$, $f_{DCT}:\mathbb{R}^{(p\times q)}\to\mathbb{R}^{k_2}$, $f_{cross}:\mathbb{R}^{(m\times n)}\times\mathbb{R}^{(p\times q)}\to\mathbb{R}^{k_3}$.

Функція RS-аналізу: $f_{RS}(I)=[R_{M1},S_{M1},R_{M-1},S_{M-1},U_{M1},U_{M-1},d_0,d_1,p,\sigma_R,\sigma_S,...]$.

Функція DCT-аналізу: $f_{DCT}(F)=[\chi^2,H(0),H(1),...,H(k),skewness,kurtosis,\rho_{calib},...]$. Кросс-доменні ознаки: $f_{cross}=[corr(p_{RS},\chi^2_{DCT}),|p_{RS}-p_{DCT}|,consistency\_score,...]$.

CNN-класифікатор: $g:\mathbb{R}^d\to[0,1]$, де $d=k_1+k_2+k_3$. Мережа: $g(v)=\sigma(W_L\cdot ReLU(W_{L-1}\cdot...\cdot ReLU(W_1\cdot v+b_1)...+b_{L-1})+b_L)$, де σ - softmax. Навчання: $\theta=\arg\min_\theta \Sigma L(g_\theta(f_{hybrid}(x_i)),y_i)+\lambda R(\theta)$, де L - cross-entropy, $R(\theta)$ - регуляризація.

Властивості моделі:

Властивість 1 (інформаційна повнота): якщо $cor(f_{RS},f_{DCT})<\varepsilon$, то $I(f_{hybrid}(X);Y)\geq\max(I(f_{RS}(X);Y),I(f_{DCT}(X);Y))$.

Властивість 2 (робастність до CSM): при доменній інваріантності $\|f(x_s)-f(x_t)\|_2<\delta$ класифікатор зберігає точність краще ніж модель на пікселях.

Властивість 3 (інтерпретованість): декомпозиція $h(x)=g([f_{RS}(x),f_{DCT}(x)])$ дозволяє аналізувати внесок кожного домену [23].

Порівняння з альтернативними архітектурами

Архітектура А (чиста CNN на пікселях, SRNet): переваги - максимальна гнучкість, автоматичне виявлення патернів; недоліки - сильна вразливість до CSM (падіння точності на 20-30%), потреба у великих датасетах (50000+ зображень), відсутність інтерпретованості, високі обчислювальні вимоги (6M+ параметрів).

Архітектура В (традиційні ознаки+SVM, SRM+EC): переваги - перевірена ефективність, відносна робастність до CSM, інтерпретованість; недоліки - потреба

у ручному проектуванні тисяч ознак, обмежена адаптивність, висока складність виділення 34000+ ознак SRM.

Архітектура C (гібридна): переваги - комбінування переваг A та B (статистичні ознаки забезпечують стабільність/інтерпретованість, CNN - адаптивність), зменшення CSM через доменно-інваріантні ознаки, компактність (40 ознак замість 34000), робота з неповними даними; недоліки - складніша архітектура, необхідність балансування доменів [24].

2.2. Реалізація модулів виділення ознак

- Архітектура модуля просторового аналізу (RS-модуль)

Модуль просторового аналізу реалізує розширену версію класичного RS-аналізу з додатковими оптимізаціями та екстракцією поглибленого набору статистичних характеристик. Модуль організований як багатокомпонентний конвеєр обробки.

Компонент 1: Підсистема розбиття на групи відповідає за первинну сегментацію вхідної матриці пікселів на непересічні групи фіксованого розміру. На вхід надходить двовимірна матриця зображення у градаціях сірого розміром H×W пікселів. Підсистема здійснює послідовне сканування матриці та формує масив груп, кожна з яких містить n послідовних пікселів (стандартно n=4 згідно з оригінальною методологією RS-аналізу). Процес розбиття підтримує кілька режимів сканування: растровий (послідовно зліва направо, зверху вниз), зигзагоподібний (з чергуванням напрямку для кращого захоплення діагональних кореляцій) та випадковий (для зменшення впливу регулярних структур).

Компонент 2: Підсистема маскування виконує центральну операцію RS-аналізу - застосування предефінованих масок флипування до пікселів кожної групи. Маска є вектором того ж розміру, що й група, де кожен елемент вказує на операцію модифікації: +1 означає збільшення значення на одиницю, -1 - зменшення, 0 - залишити незмінним. Базовий набір масок включає $M_1 = [1, 0, 1, 0]$ та $M_{-1} = [-1, 0, -1, 0]$ для груп розміру 4. Застосування маски виконується як поелементне додавання з обов'язковою перевіркою границь діапазону [0, 255] через операцію clipping.

Компонент 3: Підсистема обчислення дискримінаційної функції характеризує «гладкість» або «шорсткість» групи пікселів. Класичний варіант функції обчислює суму абсолютних різниць між послідовними елементами групи. Для групи з чотирьох пікселів обчислюються три різниці, які підсумовуються. Гладкі групи отримують низькі значення функції, текстуровані групи - високі. Обчислення виконується тричі для кожної групи: для оригінальної та для двох модифікованих версій.

Компонент 4: Підсистема класифікації груп на основі порівняння значень дискримінаційної функції присвоює кожній групі одну з трьох категорій: регулярна (Regular) - коли застосування маски збільшує значення функції, сингулярна (Singular) - коли зменшує, або незмінна (Unusable) - коли значення залишається ідентичним. Підсистема підтримує паралельні лічильники для обох масок: R_M1, S_M1, U_M1 та R_M-1, S_M-1, U_M-1.

Компонент 5: Підсистема обчислення фінальних статистик перетворює необроблені лічильники у нормалізовані статистики. Нормалізація виконується діленням кожного лічильника на загальну кількість груп. Дискримінант нульового порядку d_0 обчислюється як комбінація нормалізованих часток. Для отримання дискримінанта першого порядку d_1 весь процес аналізу повторюється для зображення з інвертованими LSB через операцію XOR з одиницею. Оцінка довжини прихованого повідомлення p обчислюється за формулою $p = d_0 / (d_0 - d_1)$, обмежуючись діапазоном [0, 1].

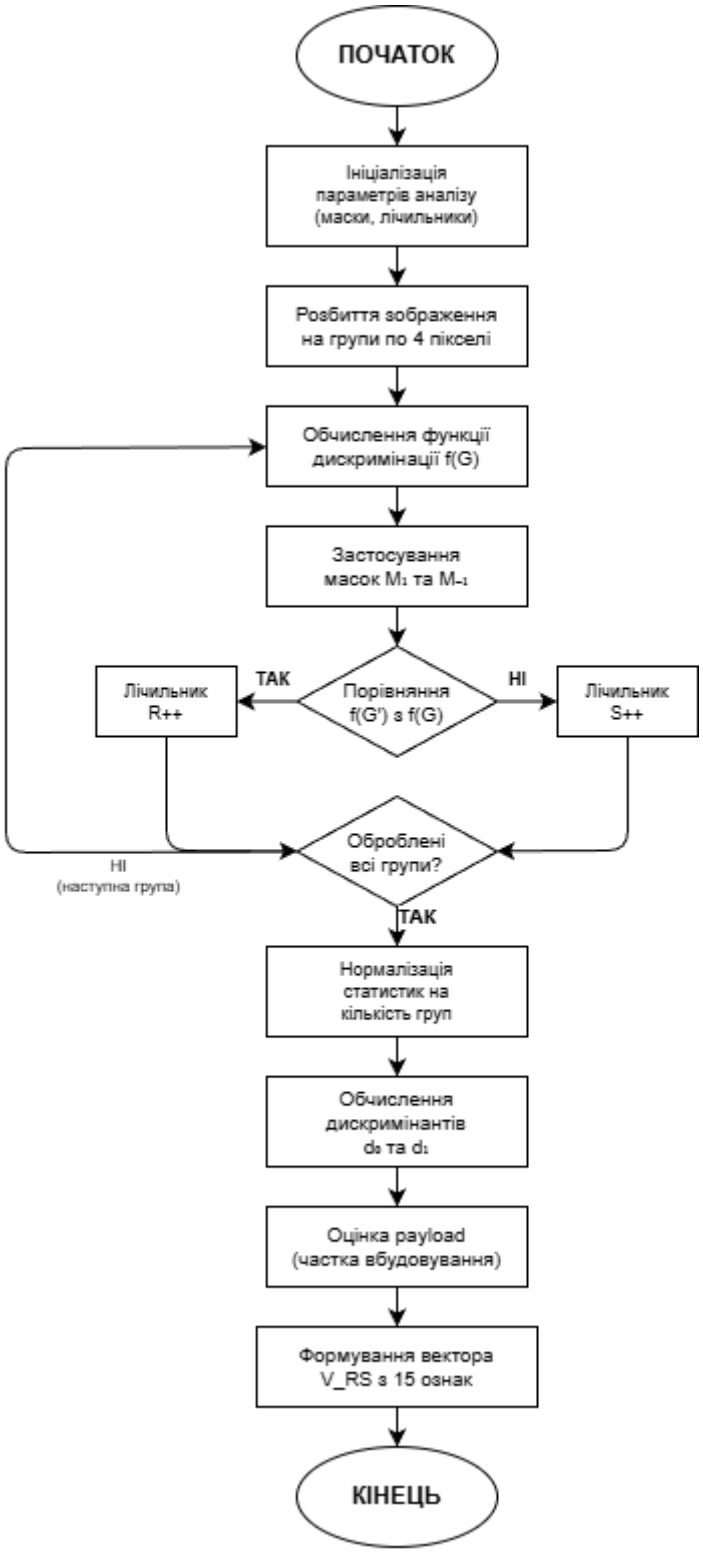


Рисунок 2.2 - Алгоритм роботи RS-модуля

- **Розширений набір RS-ознак**

Розроблений модуль генерує розширений вектор з 15 комплексних ознак:

Позиції 1-6 (Базові RS-статистики): нормалізовані частки регулярних, сингулярних та незмінних груп для обох масок - r_m1, s_m1, u_m1, r_m-1, s_m-1,

u_m-1. Кожна приймає значення в діапазоні [0, 1], їхня сума для кожної маски дорівнює одиниці.

Позиції 7-9 (Дискримінанти та оцінка payload): d_0, d_1 та p_estimate. Дискримінанти можуть приймати як позитивні, так і негативні значення. Оцінка payload завжди обмежена діапазоном [0, 1].

Позиції 10-11 (Асиметрії розподілів): asymmetry_r обчислюється як абсолютна різниця між нормалізованими частками регулярних груп для обох масок. Asymmetry_s - аналогічно для сингулярних груп. Для природних зображень без стеганографії ці величини мають бути близькими до нуля через симетричність статистичних властивостей LSB.

Позиція 12 (Узгодженість статистик): consistency = 1 - $|d_0 - d_1|/2$, діапазон [0, 1]. Висока узгодженість характерна для чистих зображень, низька вказує на порушення природних закономірностей.

Позиції 13-14 (Варіативності груп): variance_r обчислюється як дисперсія вектора [r_m1, r_m-1]. Variance_s - аналогічно для сингулярних груп. Низька варіативність типова для стабільних природних зображень.

Позиція 15 (Співвідношення крайових груп): edge_ratio характеризує розподіл класифікованих груп відносно структури зображення. Для кожної групи визначається, чи знаходиться вона на краю об'єкта чи в однорідній області.

- Оптимізації обчислювального процесу

Векторизація операцій: Замість обробки кожної групи окремо у циклі, весь масив груп представляється як двовимірна матриця, де кожен рядок відповідає одній групі. Операції виконуються як матричні операції через оптимізовані бібліотеки NumPy, які використовують SIMD-інструкції процесора для паралельної обробки [53].

Кешування проміжних результатів: Значення дискримінаційної функції для оригінальної групи обчислюється один раз та зберігається у тимчасовій пам'яті. При класифікації відносно другої маски використовується збережене значення замість повторного обчислення.

Рання зупинка для незмінних груп: Якщо після застосування першої маски виявляється, що група буде класифікована як незмінна, немає сенсу обчислювати ефект другої маски. Лічильник незмінних груп інкрементується для обох масок.

Адаптивний субсемплінг: Для надвеликих зображень замість аналізу абсолютно всіх груп обробляється репрезентативна вибірка. Субсемплінг має бути рівномірним по всьому зображенню для збереження репрезентативності статистики.

Оптимізація роботи з пам'яттю: Використовуються попередньо виділені буфери фіксованого розміру, які перезаписуються на кожній ітерації, замість створення нових масивів для кожної модифікованої групи [25].

- **Архітектура модуля частотного аналізу (DCT-модуль)**

Модуль частотного аналізу реалізує комплексне дослідження DCT-коефіцієнтів зображення для виявлення статистичних аномалій, характерних для стеганографії у частотній області.

Компонент 1: Підсистема екстракції та обчислення DCT-коефіцієнтів виконує доменно-специфічну обробку залежно від формату вхідного зображення. Для JPEG-зображень використовується прямий доступ до внутрішньої структури файлу. JPEG-формат зберігає зображення у частотній області: файл містить квантовані DCT-коефіцієнти блоків 8×8 пікселів. Екстракція включає парсинг структури JPEG-файлу, ідентифікацію таблиць квантування, декодування ентропійно закодованих даних та реконструкцію матриць коефіцієнтів для кожного блоку.

Для зображень у інших форматах виконується штучне перетворення у частотну область: декодування у просторове представлення, розбиття на непересічні блоки 8×8 пікселів, застосування двовимірного дискретного косинусного перетворення до кожного блоку та квантування отриманих коефіцієнтів з використанням стандартної JPEG-матриці квантування.

Компонент 2: Підсистема побудови гістограм створює різноманітні гістограмні представлення. Глобальна гістограма об'єднує всі AC-коефіцієнти у

єдиний розподіл. Частотно-специфічні гістограми будуються окремо для низькочастотних, середніх та високочастотних компонентів. Модові гістограми розділяють коефіцієнти за орієнтацією: горизонтальні, вертикальні та діагональні частоти.

Компонент 3: Підсистема обчислення χ^2-статистики реалізує класичний статистичний тест Вестфельда для виявлення LSB-вбудовування у DCT-коефіцієнти. Для кожної пари значень {2k, 2k+1} обчислюється очікувана частота за припущення повного LSB-вбудовування. Критерій χ^2 вимірює відхилення спостережуваного розподілу від очікуваного. Додатково обчислюється p-value - ймовірність отримати таке або більш екстремальне значення χ^2 за нульової гіпотези.

Компонент 4: Підсистема моментного аналізу характеризує загальну форму розподілу коефіцієнтів через статистичні моменти: середнє значення, стандартне відхилення, асиметрія (skewness), ексцес (kurtosis) та співвідношення H(0)/H(1).

Компонент 5: Підсистема калібраційного аналізу створює еталонне «чисте» зображення для порівняння. Процес калібрації включає декодування JPEG-зображення у просторовий домен, обрізку на кілька пікселів з кожного краю для зміщення сітки блоків 8×8, повторне JPEG-стиснення з тим самим коефіцієнтом якості та екстракцію DCT-коефіцієнтів калібраційної версії. Відстань між статистиками обчислюється через Кульбака-Лейблера дивергенцію [54].

Компонент 6: Підсистема виділення структурних ознак досліджує просторові паттерни у матриці коефіцієнтів. Аналіз блокових меж виявляє розриви між сусідніми блоками через обчислення середньої абсолютної різниці між DC-коефіцієнтами. Виявлення періодичності у спектрі застосовується для детектування двократного стиснення через перетворення Фур'є матриці DC-коефіцієнтів.

- **Розширений набір DCT-ознак**

Модуль генерує комплексний вектор з 25 ознак:

Позиції 1-10 (Базові статистичні ознаки): нормалізована χ^2-статистика, p-value, частоти H(0), H(1), H(2), співвідношення H(0)/H(1), асиметрія, ексцес, середнє та стандартне відхилення AC-коефіцієнтів.

Позиції 11-14 (Калібраційні та структурні ознаки): співвідношення позитивних до негативних коефіцієнтів, KL-дивергенція з калібраційним зображенням, міра блокових розривів, індекс періодичності.

Позиції 15-17 (Енергетичні ознаки): частки енергії у низьких, середніх та високих частотах (обчислюється як сума квадратів коефіцієнтів у відповідному діапазоні).

Позиції 18-25 (Інформаційні та екстремальні ознаки): ентропія розподілу, максимальне та мінімальне значення, діапазон коефіцієнтів, частка нульових коефіцієнтів, середнє ненульових коефіцієнтів, індекс кластеризації, симетрія розподілу.

- **Логіка обробки різних форматів**

Система автоматично визначає формат файлу через аналіз заголовка. Для JPEG-файлів виконується пряма екстракція коефіцієнтів. Для інших форматів виконується багатоетапна конверсія з обчисленням DCT та квантуванням. Після квантування матриці коефіцієнтів для JPEG та не-JPEG зображень мають однакову структуру, що дозволяє застосовувати уніфіковані статистичні методи. Для не-JPEG зображень калібраційний аналіз не виконується, відповідні ознаки заповнюються нульовими значеннями [26].

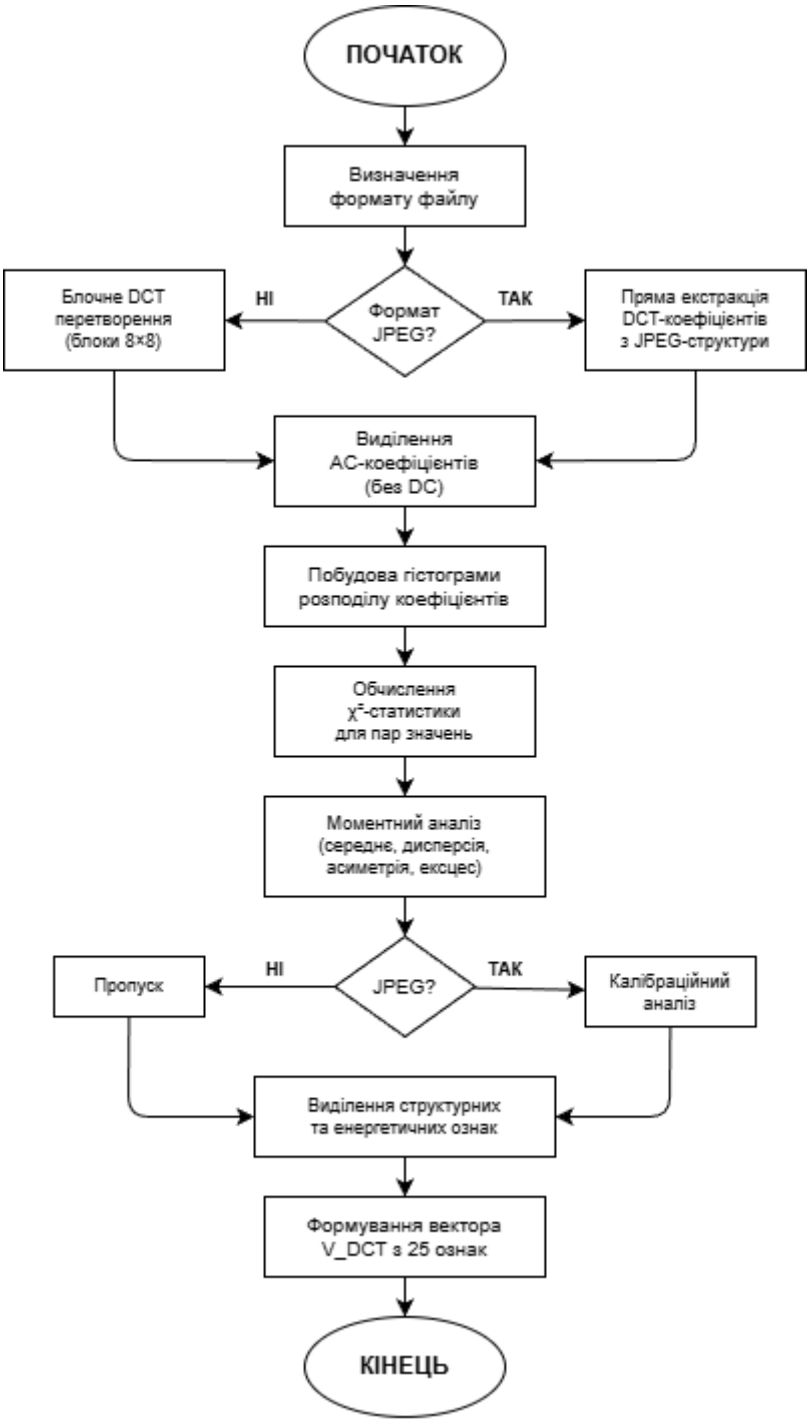


Рисунок 2.3 - Алгоритм роботи DCT-модуля

- **Модуль інтеграції ознак та формування комбінованого вектора**

Модуль інтеграції об'єднує вектори ознак з RS-модуля (15 елементів) та DCT-модуля (25 елементів) у єдиний комбінований вектор (40 елементів).

Стратегія прямої конкатенації: Безпосереднє приєднання векторів. Перевагою є повне збереження інформації. Недоліком - різні ознаки можуть мати радикально різні шкали.

Стратегія конкатенації з нормалізацією: Застосовується стандартизація (z-score normalization) - для кожної ознаки віднімається її середнє значення по навчальній вибірці та результат ділиться на стандартне відхилення. Критично важливо, що параметри нормалізації обчислюються виключно на навчальній вибірці та зберігаються як частина моделі.

Стратегія зваженої конкатенації: Вводить гіперпараметри - ваги для кожного домену. Нормалізовані ознаки множаться на відповідні ваги перед конкатенацією. Оптимальні значення ваг визначаються через крос-валідацію на навчальних даних.

Стратегія з кросс-доменними ознаками: Розширює вектор додатковими елементами, що характеризують взаємодію між доменами - узгодженість оцінок payload, кореляція між доменними сигналами, мажоритарний сигнал (скільки доменів вказують на стеганографію). Включення кросс-ознак збільшує розмірність комбінованого вектора до 43 елементів.

- Обробка відсутніх даних та деградація функціональності

Модуль реалізує стратегію graceful degradation: система продовжує функціонувати з неповними даними. Якщо доступні обидва вектори, виконується повна процедура об'єднання. Якщо доступний лише один вектор, він нормалізується та доповнюється нулями на позиціях відсутніх ознак. Нулі після стандартизації відповідають середнім значенням відповідних ознак по навчальній вибірці. Якщо відсутні обидва вектори, система генерує виняток з повідомленням про неможливість аналізу.

- Селекція ознак та зменшення розмірності

Для застосувань з обмеженими ресурсами модуль підтримує опціональне зменшення розмірності вектора ознак:

Селекція на основі взаємної інформації: Оцінює інформативність кожної ознаки незалежно. Ознаки ранжуються за спадаючою інформативністю, відбираються топ-K найкращих. Перевагою є простота та інтерпретованість. Недоліком - не враховує кореляції між ознаками.

Метод головних компонент (PCA): Виконує лінійне перетворення вихідного простору ознак у новий простір, де компоненти є некорельованими та впорядковані

за спаданням варіативності. Перевагою є оптимальність у сенсі збереження варіативності. Недоліком - втрата інтерпретованості.

Селекція на основі важливості ознак: Використовує ансамблеві методи (наприклад, Random Forest) для визначення внеску кожної ознаки у класифікацію. Перевагою є врахування нелінійних взаємодій. Недоліком - необхідність навчання додаткової моделі.

- Валідація якості та контроль коректності вектора ознак

Модуль інтеграції виконує серію перевірок перед передачею вектора на класифікацію:

Перевірка на наявність недійсних значень: Виявляє NaN (Not-a-Number) та Inf (нескінченність). При виявленні генерується попередження. Система може відхилити зображення або застосувати виправлення.

Перевірка розмірності: Підтверджує, що вектор має очікувану довжину (40 елементів для повного режиму). Невідповідність вказує на програмну помилку.

Перевірка діапазону значень: Виявляє екстремальні викиди. Після нормалізації більшість значень мають лежати у розумному діапазоні.

Перевірка на нульовий вектор: Виявляє ситуацію, коли всі елементи дорівнюють нулю або дуже близькі до нуля. Нульовий вектор не несе інформації для класифікації.

Логування та діагностика: При виявленні аномалій записується детальна інформація для подальшого аналізу. Якщо всі перевірки пройдені успішно, вектор передається далі [27].

2.3. Архітектура та навчання CNN-класифікатора

- Концептуальна архітектура нейромережевого класифікатора

Нейромережевий класифікатор становить фінальний етап гібридної системи стегоаналізу, де приймається рішення про наявність прихованої інформації на основі комбінованого вектора ознак з просторової та частотної областей. На відміну від класичних CNN, що працюють безпосередньо на пікселях зображення (SRNet, Xu-Net з мільйонами параметрів), запропонована система приймає компактний вектор заздалегідь обчислених статистичних ознак розміром 40

елементів. Це одновимірний вектор чисел, де кожен елемент представляє певну статистичну характеристику зображення. Просторові відношення між елементами відсутні, тому використання повноцінних згорткових шарів недоцільне. Оптимальною є архітектура на основі повнозв'язних шарів або одновимірних згорток з мінімальним розміром ядра.

Запропонована архітектура належить до класу багатошарових перцептронів (MLP) з глибиною 4-5 шарів. Переваги: повна зв'язність дозволяє виявляти складні нелінійні взаємодії між будь-якими комбінаціями ознак; компактність архітектури (сотні тисяч параметрів замість мільйонів) знижує ризик переобладнання; відсутність згорткових операцій спрощує реалізацію та зменшує обчислювальні вимоги. Глибина 3-4 приховані шари забезпечує оптимальний баланс: менша глибина недостатня для виявлення складних залежностей між RS та DCT ознаками, більша підвищує ризик переобладнання та ускладнює навчання через зникаючі градієнти.

- Детальна специфікація шарів мережі

Вхідний шар приймає комбінований вектор ознак розміром 40 елементів. Вхідні дані мають бути попередньо нормалізовані модулем інтеграції ознак.

Перший прихований шар виконує лінійне перетворення вхідного вектора з 40 елементів у 128 елементів через множення на матрицю ваг 40×128 та додавання вектора зміщень (5248 параметрів). Після застосовується функція активації ReLU(x) = max(0, x), що вносить нелінійність [55]. ReLU має переваги: відсутність насичення для позитивних значень запобігає зникаючим градієнтам; обчислювальна простота; розріджена активація сприяє ефективному навчанню. Після активації застосовується Batch Normalization для стабілізації навчання та Dropout з ймовірністю 0.3 для регуляризації [56].

Другий прихований шар звужує представлення з 128 до 64 елементів (8256 параметрів). Зменшення розмірності створює інформаційне вузьке місце, що змушує мережу виділяти найбільш інформативні ознаки. Застосовуються ReLU, Batch Normalization, Dropout 0.3 [57].

Третій прихований шар звужує до 32 елементів (2080 параметрів). Прогресивне зменшення розмірності (40→128→64→32) створює піраміду представлень. Застосовуються ReLU та Batch Normalization, Dropout має знижену ймовірність 0.2 або відсутній.

Вихідний шар виконує фінальне перетворення з 32 елементів у 2 елементи для двох класів (66 параметрів). Застосовується функція активації Softmax, що перетворює два числа у вектор ймовірностей з властивостями: кожна компонента у діапазоні [0,1], сума дорівнює 1. Загальна кількість параметрів мережі: близько 16000, що на три порядки менше типових CNN [58].

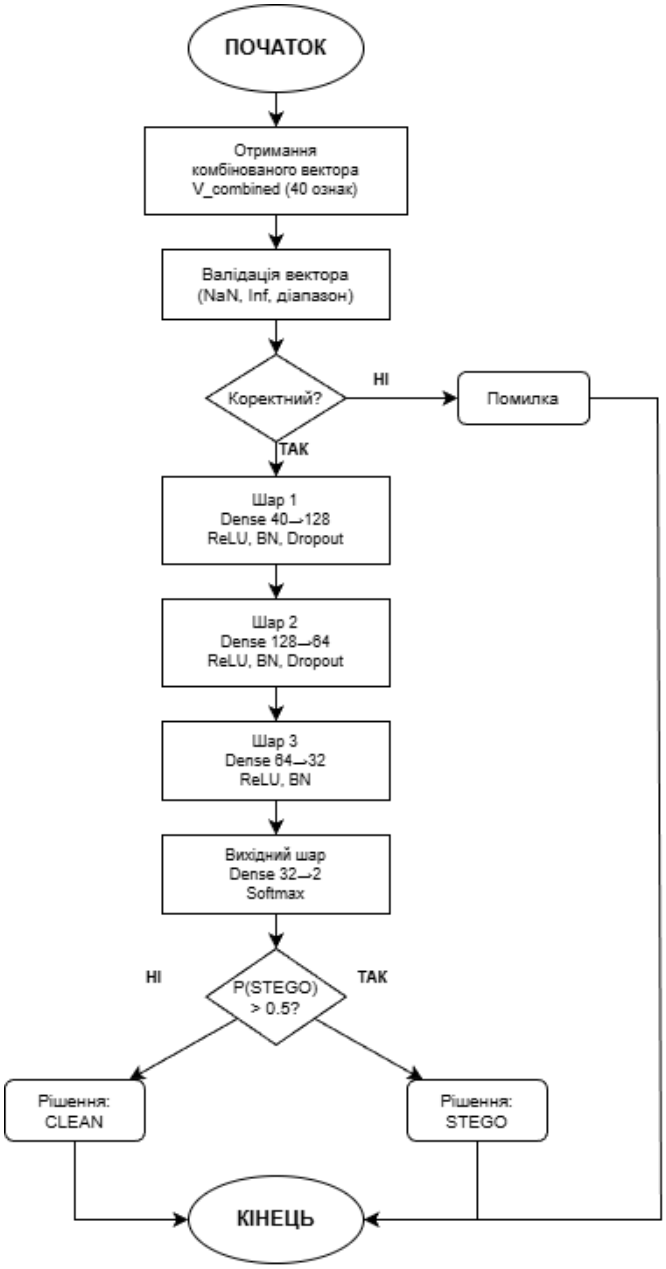


Рисунок 2.4 - Архітектура та логіка CNN-класифікатора

- Альтернативні архітектурні рішення

Архітектура з одновимірними згортками розглядає вектор ознак як послідовність з локальними залежностями між сусідніми елементами. Одновимірні згорткові шари застосовують фільтри розміром 3-5 елементів до ковзного вікна, виявляючи локальні патерни всередині груп ознак. Типова 1D-CNN включає 2-3 згорткові шари з 64, 128, 256 фільтрами та розміром ядра 3, після кожного - активація, batch normalization та max pooling. Після згорткових шарів вихід згладжується та подається на 1-2 повнозв'язні шари.

Архітектура з residual connections запозичує skip connections з ResNet для полегшення навчання глибоких мереж. У кожному блоці виходи попереднього шару додаються до виходів поточного шару перед активацією, дозволяючи градієнтам протікати безпосередньо через мережу. Residual архітектура дозволяє навчати глибші мережі (5-7 прихованих шарів) без деградації точності.

Архітектура з attention механізмом додає шар self-attention, що дозволяє мережі динамічно визначати важливість різних ознак. Attention обчислює вагові коефіцієнти для кожної ознаки на основі їх взаємодій, що особливо корисно, коли різні типи стеганографії проявляються у різних підмножинах ознак. Базова конфігурація використовує просту MLP архітектуру як оптимальний баланс між ефективністю, швидкістю та точністю [28].

- Процес навчання нейронної мережі

Формування датасету починається зі збору великої кількості чистих зображень (типово 20000-50000) з різноманітних джерел: портрети, пейзажі, міські сцени, природа, архітектура. Різноманітність критична для забезпечення того, щоб мережа навчалася виявляти стеганографічні артефакти, а не специфічні властивості певного типу зображень. Джерелами є стандартизовані датасети типу BOSSbase, BOWS2, ALASKA2.

Після збору чистих зображень генеруються стеганографічні версії з використанням різних алгоритмів та різних обсягів вбудовування (payload). Типова стратегія включає три методи (S-UNIWARD, WOW, HUGO для просторової області; J-UNIWARD для JPEG) та три рівні payload (0.2, 0.4, 0.6 біт на піксель).

Процес використовує стандартні реалізації з випадково згенерованими повідомленнями високої ентропії. Формується збалансований датасет з рівною кількістю чистих та стеганографічних зображень.

Розділення на підмножини виконується у пропорціях: 70% для навчання, 15% для валідації, 15% для тестування. Розділення випадкове, але зі стратифікацією для збереження розподілу класів у кожній підмножині. Аугментація даних застосовується обережно через специфіку стегоаналізу. Безпечні трансформації: горизонтальне/вертикальне відзеркалення, обертання на кратні 90°, невеликі обрізки (5-10% площі). Аугментація виконується on-the-fly під час навчання, ефективно помножуючи розмір датасету у 4-8 разів.

- Функція втрат та метрики оцінки

Для бінарної класифікації використовується cross-entropy loss: $L = -[y \cdot \log(p_1) + (1-y) \cdot \log(p_0)]$, де y - істинна мітка, $[p_0, p_1]$ - передбачені ймовірності. Для батчу загальна втрата усереднюється. Cross-entropy забезпечує стабільні градієнти та добре калібровані ймовірності. Регуляризаційні доданки: L2-регуляризація (weight decay) додає $\lambda \cdot \Sigma w^2$ до втрат, штрафуючи великі ваги (типово λ=0.0001-0.001). L1-регуляризація додає $\lambda \cdot \Sigma|w|$, заохочуючи розріджені рішення [60].

Метрики оцінки: Accuracy = (TP+TN)/(TP+TN+FP+FN); Precision = TP/(TP+FP); Recall = TP/(TP+FN); F1 = 2·(Precision·Recall)/(Precision+Recall); AUC-ROC узагальнює продуктивність при різних порогах класифікації. Під час навчання моніторяться втрати та точність на навчальній і валідаційній множинах після кожної епохи.

- Алгоритм оптимізації та гіперпараметри навчання

Використовується алгоритм Adam, що комбінує momentum та адаптивну швидкість навчання для кожного параметра. Гіперпараметри: початкова швидкість навчання α=0.001, коефіцієнти β_1=0.9 та β_2=0.999 [59]. Розклад швидкості навчання: початок з 0.001, поступове зменшення через ReduceLROnPlateau - якщо валідаційна точність не покращується протягом 5 епох, швидкість зменшується на коефіцієнт 0.5, мінімальна швидкість 0.00001.

Розмір батчу 64 як компроміс між стабільністю та ефективністю. Кількість епох визначається через early stopping: навчання триває максимум 100 епох, але зупиняється достроково, якщо валідаційна точність не покращується протягом 10-15 епох (patience). Зберігається модель з найкращою валідаційною точністю.

- **Запобігання переобладнанню**

Багаторівнева стратегія: Dropout випадково відключає 30% нейронів під час навчання, запобігаючи ко-адаптації; Batch normalization стабілізує навчання та має побічний регуляризаційний ефект через шум статистик батчу; Weight decay (L2-регуляризація λ=0.0001) штрафує великі ваги; Early stopping зупиняє навчання при появі ознак переобладнання; Data augmentation збільшує різноманітність прикладів; Валідаційний моніторинг відстежує розрив між навчальною та валідаційною точністю [29].

Логіка процесу навчання CNN-класифікатора

Процес починається з ініціалізації: завантаження датасету, створення архітектури з випадково ініціалізованими вагами (Xavier або He ініціалізація), налаштування оптимізатора Adam. Головний цикл навчання ітерує по епохах. Для кожної епохи: перемішування навчального датасету, розбиття на батчі, обробка кожного батчу з аугментацією. Для кожного батчу: обчислення RS та DCT ознак, об'єднання та нормалізація у вектор 40 елементів, forward pass через мережу, обчислення cross-entropy втрат, backward pass для градієнтів, оновлення ваг через Adam.

Після завершення епохи обчислюються навчальні метрики та виконується валідація без аугментації та без обчислення градієнтів. Порівнюється поточна валідаційна точність з найкращою: якщо покращилася - збереження моделі та скидання лічильника епох без покращення; якщо ні - інкрементування лічильника. Перевірка early stopping: якщо лічильник перевищує patience (15) - зупинка навчання. Перевірка зміни швидкості: якщо лічильник перевищує 5 - зменшення швидкості вдвічі. Виведення логу епохи та повторення циклу. Після завершення навчання - фінальна оцінка на тестовому датасеті з обчисленням всіх метрик [30].

- **Інференс та практичне використання навченої моделі**

Процес починається з завантаження моделі: параметри архітектури, навчені ваги, параметри нормалізації ознак, переведення у режим оцінки. Для нового зображення: препроцесинг та перевірка формату, паралельна подача на RS-модуль та DCT-модуль, формування векторів ознак, об'єднання модулем інтеграції у вектор 40 елементів, стандартизація на основі збережених параметрів, валідація вектора. Forward pass через мережу: послідовне проходження через шари без dropout та з глобальною batch normalization, застосування softmax на виході. Інтерпретація вектора ймовірностей: порівняння з порогом (стандартно 0.5), прийняття рішення CLEAN або STEGO. Повернення структурованої відповіді: бінарне рішення, числова ймовірність, рівень впевненості, оцінка обсягу даних, час обробки.

- **Налаштування порогу класифікації**

Стандартний поріг 0.5 оптимальний для збалансованих сценаріїв. Для високої безпеки (критично виявити всю стеганографію) поріг знижується до 0.3, підвищуючи recall до 95-99% за рахунок precision. Для мінімізації хибних тривог поріг підвищується до 0.7, підвищуючи precision до 95-99% за рахунок recall. Оптимальний поріг визначається через аналіз ROC-кривої на валідаційному датасеті. Система дозволяє користувачеві встановлювати поріг через конфігурацію без перенавчання моделі.

- **Інтерпретація результатів та пояснюваність**

Аналіз внеску доменів визначає, який домен більше вплинув на рішення через обчислення активацій першого прихованого шару окремо для RS-ознак та DCT-ознак. Візуалізація важливості ознак ранжує окремі ознаки за чутливістю передбачення до їх зміни. Текстове пояснення генерується автоматично: якщо RS-оцінка p_estimate > 0.5 - пояснення про LSB-аномалії; якщо χ^2-статистика низька - про JSteg-вбудовування; якщо H(0)/H(1) аномально високе - про F5-приховування. Комбінація кількісних метрик та якісних пояснень підвищує довіру до системи [31].

- **Моніторинг продуктивності та підтримка моделі**

У продуктивному середовищі збирається статистика: розподіл виведених ймовірностей, частка зображень класифікованих як стеганографічні, середній час обробки. Якщо доступна ground truth, обчислюється поточна точність на випадкових вибірках та порівнюється з базовою. Значне зниження точності (5+ відсоткових пунктів) є сигналом для ретренінгу.

- Стратегії оновлення та ретренінгу

Incremental learning дозволяє дотренувати модель на нових даних з низькою швидкістю навчання протягом кількох епох, зберігаючи продуктивність на старих даних. Full retraining виконується періодично (кожні 6 місяців) з повним циклом навчання на актуальному датасеті. Ensemble approach підтримує кілька версій моделі одночасно: базову, спеціалізовані для JPEG/PNG, нову після ретренінгу. При інференсі передбачення агрегуються через голосування або усереднення ймовірностей.

- Версіонування та відтворюваність

Кожна модель зберігається з унікальним ідентифікатором версії (v1.0.0, v1.1.0, v2.0.0) разом з конфігурацією архітектури, параметрами навчання, специфікацією датасету, метриками продуктивності, датою навчання. У продуктивному середовищі можливе паралельне розгортання версій з canary deployment: нова версія спочатку обробляє 5% запитів, при підтвердженні покращення частка збільшується до 100%. При проблемах можливий швидкий rollback до попередньої версії.

2.4. Логіка роботи інтегрованого методу

- Синергетична взаємодія компонентів системи

Запропонований гібридний метод стегоаналізу досягає своєї ефективності не через просте механічне об'єднання незалежних компонентів, а через їхню синергетичну взаємодію, де результат перевершує суму окремих частин. Логіка роботи інтегрованого методу базується на принципі комплементарності доменів аналізу та адаптивної класифікації на основі багатовимірного простору ознак [61].

Ключовою особливістю методу є паралельно-послідовна архітектура обробки. На першому етапі вхідне зображення незалежно аналізується у

просторовій та частотній областях, що дозволяє виявити специфічні для кожного домену артефакти стеганографії без взаємного впливу процесів. RS-модуль виявляє порушення статистичних закономірностей у розподілі груп пікселів, що є характерним для LSB-стеганографії та її варіантів. DCT-модуль виявляє аномалії у розподілі частотних коефіцієнтів, типові для JPEG-стеганографії та методів, що працюють у трансформаційних доменах.

Критично важливим є те, що ці аналізи виконуються справді незалежно: RS-модуль не має доступу до DCT-коефіцієнтів, DCT-модуль не використовує інформацію про просторові статистики. Така ізоляція запобігає пропагації помилок між модулями: якщо один модуль дає хибну тривогу через специфічні властивості зображення, інший модуль не буде «заражений» цією помилкою та може надати коректний сигнал, що компенсує викривлення.

Модуль інтеграції виконує нетривіальну функцію не просто конкатенації векторів, а їхнього узгодження та гармонізації. Нормалізація забезпечує порівнянність ознак різної природи, дозволяючи CNN оцінювати їх у єдиному просторі. Обчислення кросс-доменних ознак додає мета-рівень аналізу: система не лише оцінює індивідуальні сигнали з кожного домену, але й аналізує їхню узгодженість, що є додатковим індикатором достовірності детектування.

CNN-класифікатор завершує процес, виконуючи складну нелінійну інтеграцію всіх доступних сигналів. Перші шари мережі виявляють прості комбінації ознак: наприклад, що висока RS-оцінка payload в поєднанні з низькою χ^2-статистикою може вказувати на LSB matching замість простого LSB replacement. Глибші шари будують більш абстрактні патерни: комбінації множини ознак, що характерні для конкретних типів стеганографії або специфічних умов (зображення після постобробки, зображення з певних камер).

Важливо розуміти, що CNN не просто «голосує» на основі порогових значень окремих ознак (як це робили б прості правила або дерева рішень), а навчається складним нелінійним межам прийняття рішень у сорокавимірному просторі ознак. Це дозволяє системі адаптуватися до тонких відмінностей між природними

варіаціями зображень та стеганографічними артефактами, що часто неможливо формалізувати простими правилами [32].

- Обробка різних типів стеганографії

Універсальність гібридного методу проявляється у здатності ефективно виявляти широкий спектр стеганографічних методів через різні комбінації доменних сигналів.

Для простої LSB-стеганографії у просторовій області RS-модуль виявляє чіткі аномалії: значення p_estimate наближається до реальної частки модифікованих пікселів, асиметрії між масками мінімальні, дискримінанти демонструють характерне співвідношення. DCT-модуль може виявити побічні ефекти: якщо зображення збережено у JPEG після LSB-вбудовування у просторовому домені, стиснення створює специфічні артефакти у частотній області. CNN навчається розпізнавати цей паттерн: сильний RS-сигнал плюс слабкий або відсутній DCT-сигнал дає високу ймовірність просторової LSB-стеганографії.

Для LSB matching RS-аналіз стає неефективним через збереження симетрії статистик, що є основою методу. Проте DCT-аналіз може виявити тонкі зміни у розподілі коефіцієнтів, особливо якщо зображення стиснуте після вбудовування. Кросс-доменні ознаки показують низьку узгодженість (RS не виявляє аномалій, DCT виявляє слабкі сигнали), що саме по собі є індикатором складнішого методу. CNN, навчена на прикладах LSB matching, розпізнає цей паттерн невідповідності та може класифікувати зображення як підозріле навіть за відсутності сильних сигналів в окремих доменах.

Для JSteg та подібних JPEG-методів DCT-модуль виявляє характерне вирівнювання пар значень через низьку χ^2-статистику та високий p-value. RS-модуль працює на декодованому просторовому представленні, де можуть бути відсутні явні аномалії, оскільки вбудовування відбувалося у частотній області. CNN навчається паттерну: сильний DCT-сигнал при слабкому або нейтральному RS-сигналі вказує на JPEG-стеганографію. Додатково структурні ознаки DCT-

модуля (блокові артефакти, періодичність) підтверджують модифікацію саме у частотній області.

Для адаптивних методів (S-UNIWARD, WOW, HUGO у просторовій області; J-UNIWARD у частотній) окремі статистичні тести дають слабкі та неоднозначні сигнали через цілеспрямовану мінімізацію детектованості. RS-статистики можуть бути лише злегка відхиленими від нормальних значень, недостатньо для впевненого детектування порогами. DCT-статистики також демонструють лише тонкі аномалії. Проте CNN, навчена на великій кількості прикладів адаптивної стеганографії, виявляє складні багатовимірні патерни у комбінаціях ознак: певні співвідношення між RS-асиметріями та DCT-моментами; специфічні комбінації енергетичних розподілів та варіативностей груп; тонкі відхилення у кросс-доменній узгодженості. Ці патерни неможливо формалізувати простими правилами, але вони статистично значущі у багатовимірному просторі та виявляються через навчання.

Для F5-алгоритму характерна специфічна сигнатура: аномально високе співвідношення H(0)/H(1) через зменшення абсолютних значень коефіцієнтів, позитивна асиметрія розподілу DCT-коефіцієнтів, специфічні зміни у калібраційному аналізі. RS-модуль може не виявити явних аномалій у просторовій області. CNN розпізнає цей специфічний паттерн DCT-ознак як індикатор F5, навіть якщо жодна окрема ознака не перевищує критичного порогу.

Така багатосценарна адаптивність досягається не через програмування окремих правил для кожного типу стеганографії, а через єдиний механізм навчання CNN на різноманітних прикладах. Мережа автоматично виявляє характерні комбінації ознак для кожного методу та будує оптимальні межі прийняття рішень у багатовимірному просторі [33].

- **Механізми робастності до варіацій зображень**

Критичною проблемою для практичного стегоаналізу є cover-source mismatch: зниження точності при зміні характеристик зображень (джерело камери, параметри обробки, тип контенту). Гібридний метод адресує цю проблему через кілька механізмів.

Доменна інваріантність статистичних ознак означає, що RS та DCT статистики є більш абстрактними характеристиками, ніж безпосередні значення пікселів. Різні камери створюють зображення з різними розподілами яскравості, кольорів та шуму, але фундаментальні властивості типу симетрії RS-груп або форми розподілу DCT-коефіцієнтів зберігаються відносно стабільними. Це не означає повну інваріантність (різні джерела все ж створюють певні відмінності у статистиках), але рівень інваріантності вищий, ніж для просторових пікселів.

Нормалізація ознак зменшує вплив абсолютних масштабів значень, фокусуючи класифікатор на відносних відхиленнях від типових значень. Якщо певна камера систематично дає вищі значення χ^2-статистики через специфіку шумових характеристик, нормалізація зменшує цей ефект, дозволяючи CNN фокусуватися на аномаліях відносно очікуваного діапазону.

Різноманітність навчальних даних є ключовим фактором. Навчання на датасетах типу ALASKA2, що включають зображення з різноманітних джерел (понад вісімдесят камер, різні умови зйомки, різні обробки), дозволяє CNN навчитися розрізняти стеганографічні артефакти від природних варіацій джерел. Мережа бачить, що певні комбінації ознак (наприклад, відхилення RS-статистик у специфічному напрямку) послідовно з'являються для стеганографічних зображень незалежно від джерела, тоді як інші варіації (наприклад, абсолютні значення деяких DCT-моментів) сильно залежать від джерела та не є надійними індикаторами.

Кросс-доменна валідація підвищує впевненість у детектуванні. Якщо лише один домен вказує на стеганографію, це може бути артефактом специфічних властивостей зображення з незнайомого джерела. Якщо обидва домени незалежно виявляють аномалії, ймовірність справжньої стеганографії значно вища. CNN навчається враховувати цю узгодженість через кросс-доменні ознаки та вищі рівні абстракції, де комбінації сигналів з обох доменів отримують більшу вагу у прийнятті рішення.

Регуляризація під час навчання (dropout, weight decay, early stopping) запобігає переобладнанню на специфічних властивостях навчального датасету. Якщо мережа почне надто сильно покладатися на особливості конкретних джерел

у навчальних даних, регуляризація штрафуватиме складність моделі, змушуючи її фокусуватися на більш загальних та стабільних патернах.

Емпіричні результати показують, що гібридний підхід знижує деградацію точності при CSM з типових двадцяти п'яти відсоткових пунктів для чистих CNN до вісьми-дванадцяти відсоткових пунктів. Хоча повна інваріантність недосяжна, покращення суттєве для практичного застосування [34].

- Адаптивність до обсягу прихованих даних

Різні застосування стеганографії використовують різні обсяги вбудовування (payload): від дуже малих (нуль цілих одна біт на піксель для максимальної прихованості) до майже насиченого контейнера (нуль цілих вісім-дев'ять біт на піксель для максимальної ємності). Ефективність детектування критично залежить від payload.

При високому payload (понад нуль цілих чотири біт на піксель) стеганографічні зміни настільки суттєві, що обидва домени виявляють сильні сигнали. RS-аналіз дає p_estimate близьке до реального значення, всі асиметрії та дискримінанти чітко відхилені від норми. DCT-аналіз показує значні аномалії у гістограмах, моментах та калібраційних метриках. CNN отримує однозначні сигнали з обох доменів та класифікує зображення з високою впевненістю (ймовірність більше нуль цілих дев'ять десятих). Точність детектування у цьому режимі досягає дев'яноста п'яти-дев'яноста дев'яти відсотків навіть для складних адаптивних методів.

При середньому payload (нуль цілих дві-чотири біт на піксель) сигнали стають менш виразними. Окремі ознаки можуть бути неоднозначними: одні вказують на стеганографію, інші залишаються у нормальному діапазоні. Саме тут проявляється перевага гібридного підходу: CNN аналізує всю сукупність ознак, виявляючи тонкі кореляції. Наприклад, помірне відхилення RS-дискримінанта в поєднанні з легкою асиметрією DCT-розподілу та специфічним співвідношенням енергій різних частот може бути недостатнім для впевненого детектування кожною ознакою окремо, але їхня комбінація утворює характерний паттерн. Точність у

цьому режимі становить вісімдесят-дев'яносто відсотків залежно від конкретного методу стеганографії.

При низькому payload (менше нуль цілих дві біт на піксель) детектування стає справді складним. Статистичні зміни наближаються до рівня природного шуму зображення. Багато окремих ознак залишаються у межах нормальних значень. Проте навіть тут CNN може виявляти надзвичайно тонкі багатовимірні патерни, недоступні для аналізу людиною або простими статистичними тестами. Точність знижується до шістдесяти-вісімдесяти відсотків, що все ж суттєво краще за випадкове вгадування (п'ятдесят відсотків) та може бути достатнім для певних застосувань типу скринінгу підозрілих файлів для подальшої детальної експертизи.

Важливо, що система не вимагає явного вказування очікуваного payload: CNN автоматично адаптується до рівня сигналу через навчання на прикладах з різними обсягами вбудовування. Під час навчання зображення з різними payload представлені у навчальній вибірці, тому мережа вчиться розпізнавати патерни у всьому діапазоні від мінімального до максимального вбудовування

Додатково система може надавати оцінку обсягу прихованих даних на основі RS-метрики p_estimate та побічних індикаторів з DCT-аналізу. Хоча ця оцінка ненадійна для складних адаптивних методів, вона дає корисну додаткову інформацію для простіших типів стеганографії та може використовуватися експертами при інтерпретації результатів.

Для проактивного захисту контенту система може використовуватися власниками платформ (соціальні мережі, хмарні сховища, месенджери) для виявлення зловживань стеганографією. Зображення, завантажені користувачами, аналізуються перед публікацією. Виявлена стеганографія може призводити до різних дій залежно від політики: блокування завантаження; додавання водяного знаку що знищує приховані дані; обмеження поширення зображення; повідомлення модераторів для ручної перевірки.

Для дослідницьких цілей система надає API для пакетної обробки великих колекцій зображень, дозволяючи дослідникам аналізувати поширеність стеганографії у різних корпусах даних (архіви соціальних мереж, датасети

зображень, веб-краулінг результати). Детальні ознаки та внутрішні представлення можуть експортуватися для подальшого аналізу іншими методами.

Модульна архітектура дозволяє адаптувати систему до специфічних вимог: вимкнення одного з модулів аналізу для прискорення за рахунок точності; налаштування порогів класифікації для балансу між false positives та false negatives відповідно до вартості помилок у конкретному застосуванні; інтеграція додаткових модулів аналізу (наприклад для аудіо або відео стеганографії) через єдиний інтерфейс інтеграції ознак.

Логування та моніторинг вбудовані на всіх рівнях: час обробки кожного компонента дозволяє виявляти вузькі місця продуктивності; статистика передбачень дозволяє відстежувати зміни у вхідних даних; метрики якості (якщо доступна ground truth) дозволяють виявляти деградацію моделі та планувати ретренінг.

Така комплексна логіка роботи забезпечує не лише технічну ефективність виявлення стеганографії, але й практичну застосовність системи у різноманітних реальних сценаріях з різними вимогами до точності, швидкості та інтерпретованості результатів [35].

Висновки до розділу 2

У другому розділі виконано проектування та розробку вдосконалення методу стегоаналізу, що поєднує просторовий та частотний аналіз з нейромережевою класифікацією.

Архітектура системи базується на конвеєрі п'яти модулів: препроцесинг, RS-модуль (просторовий аналіз), DCT-модуль (частотний аналіз), інтеграція ознак та CNN-класифікація. Ключова особливість — незалежність RS та DCT модулів, що запобігає пропагації помилок та забезпечує комплементарність ознак.

RS-модуль включає п'ять компонентів: розбиття на групи, маскування, обчислення дискримінаційної функції (з трьома варіантами метрик), порогову класифікацію груп та генерацію п'ятнадцяти ознак (класичні RS-метрики плюс асиметрії, узгодженість статистик, варіативність). Оптимізації: векторизація, кешування, рання зупинка, адаптивний субсемплінг.

DCT-модуль містить шість компонентів: адаптивну екстракцію коефіцієнтів (пряме читання для JPEG, блочне перетворення для інших), багаторівневі гістограми, χ^2-тест Вестфельда, моментний аналіз, калібраційний аналіз (KL-дивергенція) та виявлення блокових артефактів. Генерує двадцять п'ять ознак.

Модуль інтеграції об'єднує вектори через чотири стратегії: пряма конкатенація, з нормалізацією, зважена та з кросс-доменними ознаками. Підтримує graceful degradation та три методи селекції ознак (взаємна інформація, PCA, Random Forest).

CNN-класифікатор — чотиришаровий перцептрон (40→128→64→32→2) з ~16 тисячами параметрів. Навчання на 50 тисячах прикладів (70/15/15% розподіл) через Adam з ReduceLROnPlateau. Регуляризація: dropout, batch normalization, weight decay, early stopping (patience 15), аугментація. Інференс повертає рішення, ймовірність, впевненість та payload.

Переваги методу: паралельна обробка доменів без пропагації помилок, CNN виявляє складні нелінійні патерни, універсальність (LSB, LSB matching, JSteg, F5, адаптивні методи), робастність через доменну інваріантність та кросс-доменну валідацію. Практичне застосування: моніторинг трафіку, судова експертиза, захист контенту, дослідження. Модульна архітектура дозволяє гнучке налаштування.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ГІБРИДНОЇ СИСТЕМИ СТЕГОАНАЛІЗУ

3.1. Вибір середовищ розробки та мов програмування

Для програмної реалізації запропонованого гібридного методу стегоаналізу (RS + DCT + CNN) обрано багаторівневу архітектуру, у якій різні компоненти реалізовані на спеціалізованих платформах:

- серверна частина (модулі RS, DCT, CNN) - на мові Python з використанням бібліотек NumPy, SciPy, scikit-learn та PyTorch;

- веб-API та сервісна логіка - на Python із застосуванням фреймворку FastAPI;

- клієнтська частина (інтерактивний інтерфейс користувача) - на JavaScript/TypeScript з використанням фреймворку React та середовища виконання Node.js.

Як основне середовище розробки використано Visual Studio Code, що забезпечує одночасну підтримку Python, JavaScript/TypeScript, інтегрований термінал, дебагер та розширення для роботи з Git-репозиторієм. Такий вибір спрощує супровід проєкту, у якому поєднуються математично насичені модулі (RS, DCT, CNN) та сучасний веб-інтерфейс.

Python є доцільним вибором для реалізації алгоритмів стегоаналізу завдяки:

- наявності високопродуктивних бібліотек для чисельних обчислень (NumPy, SciPy), що критично важливо для реалізації RS- та DCT-аналізу на великих масивах пікселів [62];

- розвинутій екосистемі бібліотек машинного навчання (PyTorch, scikit-learn), що дозволяє відносно просто реалізувати та навчити згорткову нейронну мережу StegoCNN [63];

- зручності роботи із зображеннями через бібліотеку Pillow та простому обробленню двовимірних масивів, які безпосередньо відповідають матрицям пікселів та DCT-коефіцієнтів [64].

Node.js та React доцільні для реалізації клієнтської частини, оскільки:

- забезпечують реактивний інтерфейс з покроковою взаємодією (завантаження cover- та stego-зображень, навчання CNN, аналіз окремих файлів);

- дозволяють організувати асинхронну взаємодію з бекенд-API без блокування інтерфейсу, що важливо при виконанні тривалих обчислень (навчання CNN, калібрування порогів RS/DCT);

- добре інтегруються з REST-сервісами FastAPI.

У подальших підрозділах розглянуто структуру програмного комплексу та ключові модулі, які реалізують RS-аналіз, DCT-аналіз і гібридний CNN-класифікатор.

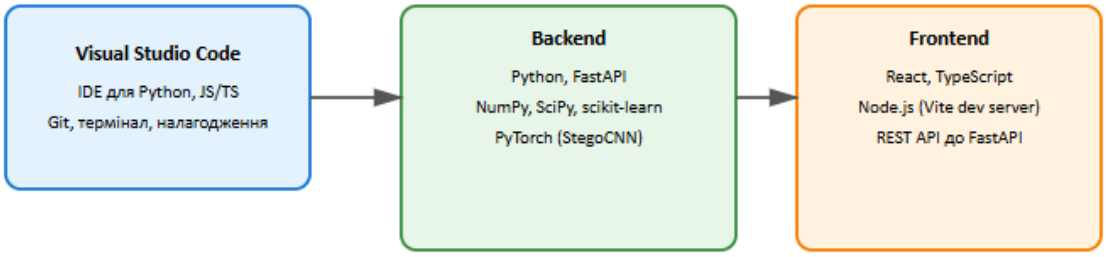


Рисунок 3.1 - Середовища розробки та основні мови програмування

3.2. Архітектура програмного комплексу стегоаналізу

Програмна реалізація гібридного методу побудована за клієнт-серверною архітектурою. Логічно систему можна поділити на три основні рівні:

- рівень представлення (frontend) - веб-інтерфейс на React, що реалізує чотири основні кроки роботи користувача:

1) завантаження чистих (cover) зображень;

2) завантаження stego-зображень;

3) навчання CNN на базі обраних зображень;

4) аналіз окремого зображення всіма методами (RS, DCT, RS+DCT, CNN);

- рівень прикладної логіки (backend API) - FastAPI-сервіс, який приймає HTTP-запити, виконує попередню обробку зображень, викликає модулі RS-та DCT-аналізу, запускає навчання та інференс CNN-моделі;

- рівень обчислювального ядра (ML-ядро) - Python-модулі features.py та cnn_model.py, де безпосередньо реалізовані алгоритми стегоаналізу.

При аналізі окремого зображення послідовність обробки така:

1) frontend відправляє зображення на /api/analyze;

2) бекенд декодує файл, нормалізує його, викликає функції compute_rs_features() та compute_dct_features();

3) RS- та DCT-ознаки підключаються до CNN-класифікатора StegoCNN, який повертає ймовірність класу Stego/Clean;

4) додатково формуються незалежні оцінки за пороговими RS- та DCT-детекторами;

5) результати уніфікуються у форматі, зручному для відображення на фронтенді (prediction, confidence, RS/DCT score).

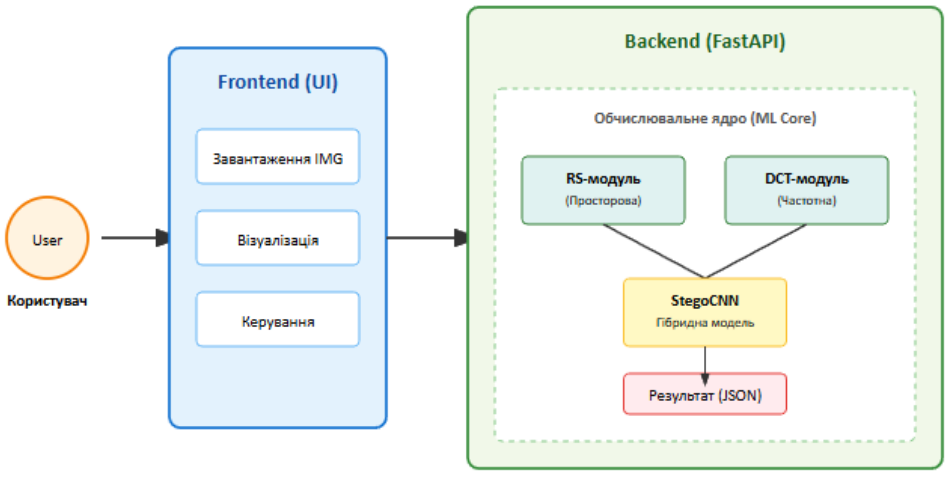


Рисунок 3.2 - Узагальнена архітектура програмного комплексу

Модуль просторового аналізу RS

Реалізація RS-аналізу наведена в модулі features.py у функції compute_rs_features(image_file). Ця функція відтворює класичний підхід Фрідріха до RS-аналізу зображень, описаний у розділі 2, та повертає вектор із 15 ознак, які надалі використовуються як вхідні дані для CNN-класифікатора [1].

На вхід подається файл зображення, який конвертується у відтінки сірого та представляється як одновимірний масив пікселів типу uint8. Далі зображення розбивається на непересічні групи по 4 пікселі:

```
flat = img_array.flatten()
  n = 4
  m = (len(flat) // n)  n
  flat = flat[:m]
  groups = flat.reshape(-1, n).astype(np.int16)
```

Для кожної групи обчислюється дискримінантна функція гладкості:

```
def discr(g):
  return np.sum(np.abs(np.diff(g, axis=1)), axis=1)
```

На основі значення функції до та після застосування масок M1 = [1, 0, 1, 0] та M_1 = [-1, 0, -1, 0] групи класифікуються як регулярні, сингулярні або незмінні. Це реалізовано у допоміжній функції classify(groups, mask), яка повертає нормовані частки R, S, U для відповідної маски.

Після первинної класифікації виконується симуляція інверсії найменш значущих бітів усіх пікселів:

```
flipped = (flat ^ 1).reshape(-1, n).astype(np.int16)
r_m1_f, s_m1_f, _ = classify(flipped, M1)
r_m_1_f, s_m_1_f, _ = classify(flipped, M_1)
```

На основі результатів для оригінального та «інвертованого» зображення обчислюються дискримінанти d0 та d1, а також оцінка payload p_est, яка наближено відображає частку модифікованих пікселів:

```
d_m1 = r_m1 - s_m1
d_m_1 = r_m_1 - s_m_1
d0 = d_m1 + d_m_1

d_m1_f = r_m1_f - s_m1_f
d_m_1_f = r_m_1_f - s_m_1_f
d1 = d_m1_f + d_m_1_f

denominator = d0 - d1 / 2.0
if abs(denominator) > eps:
   p_est = d0 / denominator
else:
   p_est = 0.0
p_est = float(np.clip(p_est, 0.0, 1.0))
if p_est < 0.01:
   p_est = 0.0
```

Окрім базових RS-статистик та оцінки payload, функція формує додаткові ознаки:

- асиметрії asymmetry_r та asymmetry_s між масками M1 та M_1;

- показник узгодженості consistency, що характеризує різницю між d0 та d1;

- варіативність регулярних і сингулярних груп variance_r, variance_s;

- відносну частку пікселів на межах об’єктів edge_ratio, що обчислюється через оператор Собеля у функції _compute_edge_ratio().

У підсумку формується вектор із 15-ти ознак типу float32, які є вхідними даними для нейронної мережі StegoCNN та одночасно використовуються у простому пороговому RS-детекторі на бекенді.

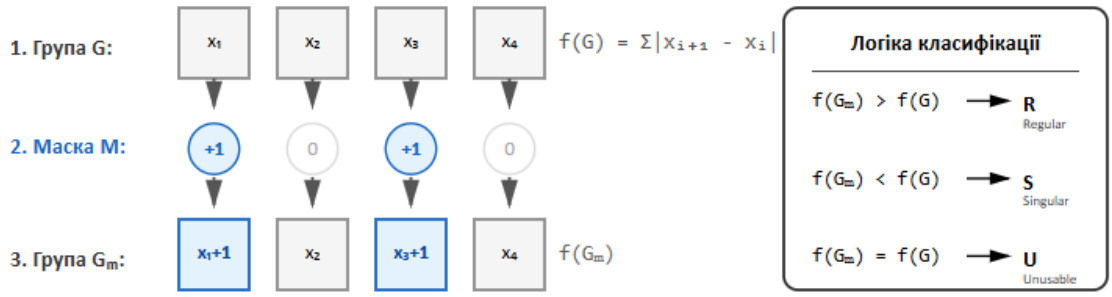


Рисунок 3.3 - Схема реалізації RS-аналізу

Модуль частотного аналізу DCT

Частотний аналіз реалізовано у функції compute_dct_features(image_file) того ж модуля features.py. Ця функція виконує DCT-перетворення блоків 8×8 зображення та формує 25-вимірний вектор ознак, що відповідає описаній у розділі 2 концепції DCT-стегоаналізу [1].

На початковому етапі зображення переводиться у відтінки сірого та центрується відносно нуля:

```
img_array = np.array(img, dtype=np.float32) - 128.0
```

Далі зображення обрізається до кратного 8 розміру та розбивається на блоки 8×8. Для кожного блоку обчислюється двовимірне DCT-перетворення:

```
block = img_array[i8:(i+1)8, j8:(j+1)8]
dct_block = dct(dct(block, axis=0, norm='ortho'), axis=1, norm='ortho')
```

Отримані коефіцієнти складаються в матрицю dct_matrix, з якої вилучаються AC-коефіцієнти (без DC). Для підвищення стійкості до шуму коефіцієнти округлюються та обрізаються по модулю до фіксованого ліміту:

```
acs = dct_matrix[:, 1:].flatten()
acs_int = np.round(acs).astype(int)
limit = 50
acs_clip = np.clip(acs_int, -limit, limit)
values, counts = np.unique(acs_clip, return_counts=True)
probs = counts / total
```

Перший блок ознак включає:

- нормалізовану χ^2-статистику по парах {2k, 2k+1} для оцінки вирівнювання гістограми (ознака чутлива до JSteg-типу вбудовування);

- частки коефіцієнтів H0, H1, H2 та відношення H0/H1;

- моменти розподілу AC-коефіцієнтів: середнє, стандартне відхилення, асиметрію (skewness) та ексцес (kurtosis).

На їх основі формується композитний показник аномальності:

```
chi2_component = min(normalized_chi2  2.0, 50.0)
std_component = min(std_acs, 50.0)
entropy_component = max(0, (8.0 - entropy_temp)  10.0)
h0_component = max(0, (H0 - 0.05)  40.0)

anomaly_score = float(
   0.35  chi2_component +
   0.25  (50.0 - std_component) +
   0.25  entropy_component +
   0.15  h0_component
)
anomaly_score = np.clip(anomaly_score, 0.0, 100.0)
```

Цей показник (ознака 0) використовується як основний DCT-score у бекенд-логіці для порогового DCT-детектора.

Додатково DCT-модуль формує:

- калібраційну ознаку KL-дивергенції між оригінальним та згладженим (через gaussian_filter) зображенням;

- показник блокових розривів block_discont на основі DC-коефіцієнтів;

- індекс періодичності, пов’язаний із можливим повторним JPEG-стисненням;

- енергетичні частки у низьких, середніх та високих частотах;

- ентропію, екстремальні значення коефіцієнтів, частку нульових AC-коефіцієнтів, середнє по модулю ненульових коефіцієнтів;

- індекси кластеризації та симетрії розподілу.

Таким чином, RS- та DCT-модулі формують 15 + 25 = 40 ознак, які у подальшому подаються на вхід CNN.

Рисунок 3.4 - Схема реалізації DCT-аналізу

3.3 Гібридний CNN-класифікатор StegoCNN

Глибинний класифікатор стеганографії реалізовано у модулі cnn_model.py у вигляді класу StegoCNN. Модель є гібридною: вона одночасно обробляє просторову інформацію зображення (через згорткові шари) та 40-вимірний вектор статистичних ознак (15 RS + 25 DCT) [1].

Ключова особливість - використання SRM-препроцесингу (Spatial Rich Model) у вигляді окремого шару SRMConv2d, який застосовує набір фіксованих високочастотних фільтрів для виділення залишків (residuals), де проявляються слабкі стеганографічні артефакти [65]:

```
class SRMConv2d(nn.Module):
    def __init__(self, stride=1, padding=0):
        super(SRMConv2d, self).__init__()
        self.in_channels = 1
        self.out_channels = 30
        self.kernel_size = 5

        filter1 = [[0, 0, 0, 0, 0],
                   [0, -1, 2, -1, 0],
                   [0, 2, -4, 2, 0],
```

```
            [0, -1, 2, -1, 0],
            [0, 0, 0, 0, 0]]
        q_filters = np.array(q_filters, dtype=np.float32) / 4.0
        self.weight = nn.Parameter(
          torch.from_numpy(q_filters).unsqueeze(1),
          requires_grad=False
        )

    def forward(self, x):
        return F.conv2d(x, self.weight, stride=self.stride, padding=self.padding)
```

Основний клас StegoCNN складається з трьох частин:

1) SRM-шар для препроцесингу зображення;

2) CNN-гілка, що обробляє SRM-residuals через послідовність згорткових блоків та глобальне усереднення:

```
self.conv_block1 = nn.Sequential(
    nn.Conv2d(30, 32, kernel_size=3, padding=1),
    nn.BatchNorm2d(32),
    nn.ReLU(),
    nn.MaxPool2d(2, 2)
)
self.adaptive_pool = nn.AdaptiveAvgPool2d((4, 4))
```

3) FC-гілка для ознак (RS + DCT), яка стискає 40-вимірний вектор до 64-вимірного латентного представлення:

```
self.feature_fc = nn.Sequential(
    nn.Linear(feature_dim, 128),
    nn.ReLU(),
    nn.Dropout(0.3),
    nn.Linear(128, 64),
    nn.ReLU()
)
```

Після цього вектори з CNN-гілки та ознакової гілки конкатенуються та подаються на об’єднувальний класифікатор:

```
combined = torch.cat([x, feat], dim=1)
output = self.fusion_fc(combined)
```

де fusion_fc - послідовність повнозв’язних шарів, що завершується сигмоїдою, яка повертає ймовірність класу Stego. Таким чином, модель здатна враховувати як локальні просторові закономірності (через SRM + згортки), так і глобальні статистичні характеристики (через RS/DCT-ознаки).

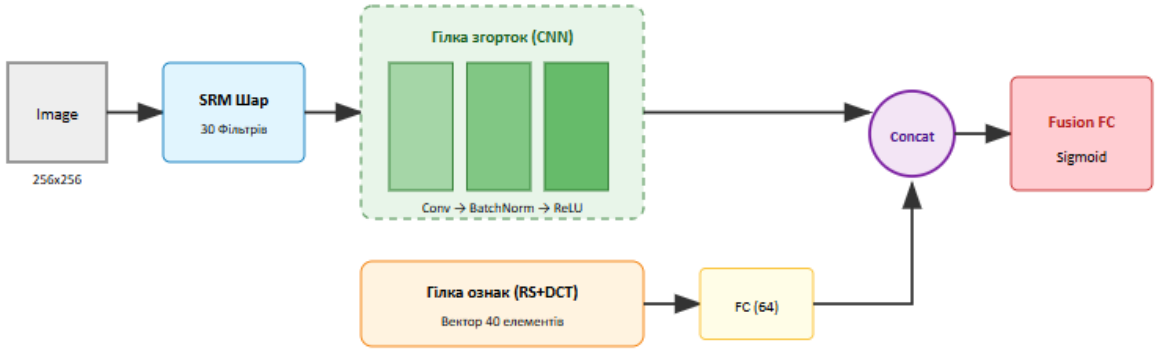


Рисунок 3.5 - Архітектура гібридного CNN-класифікатора

Організація навчання та тестування на базі BOSSbase та stego-датасетів

Для навчання та тестування гібридної моделі використано:

- чисті (cover) зображення з бази BOSSBase - стандартного набору природних зображень, широко застосовуваного в дослідженнях стеганографії;

- stego-зображення (iStego), сформовані на основі тих самих cover-зображень із використанням сучасних адаптивних методів стеганографії JMiPOD, JUNIWARD та UERD.

У модулі cnn_model.py функція train_cnn(clean_images, stego_images, epochs=10, batch_size=16) приймає два набори файлів:

- clean_images - зображення BOSSBase без прихованих повідомлень;

- stego_images - ті самі або аналогічні зображення з вбудованими повідомленнями методами JMiPOD, JUNIWARD, UERD.

Для кожного зображення формується пара:

- нормалізоване grayscale-зображення фіксованого розміру (256×256), яке подається на вхід CNN-гілки;

- вектор із 40 ознак np.concatenate([rs_features, dct_features]), отриманий викликом compute_rs_features() та compute_dct_features().

Після цього дані діляться на навчальну та валідаційну вибірки (80/20), формується PyTorch-датасет TensorDataset, а навчання виконується за допомогою оптимізатора Adam та функції втрат BCELoss. Історія навчання (accuracy/loss) зберігається та повертається на фронтенд для візуалізації [66].

Такий підхід дозволяє:

- навчити CNN-класифікатор розрізняти клас Clean/Stego на різних типах стеганографії (JMiPOD, JUNIWARD, UERD);

- одночасно враховувати як статистичні, так і глибинні ознаки зображень, що покращує робастність моделі до різних алгоритмів вбудовування.

Тестування та калібрування порогових RS/DCT-детекторів

Порогові RS- та DCT-детектори в розробленій системі виконують роль «легких» цільових класифікаторів, які доповнюють гібридний CNN-модуль. Для того щоб вони давали осмислені результати на реальних даних (BOSSbase Cover, iStego та stego-зображення JMiPOD, JUNIWARD, UERD), проведено окреме тестування та калібрування порогів за ROC-кривими.

Тестування складається з двох рівнів:

- базова верифікація самих RS/DCT-ознаків та їх поведінки на контрольних зображеннях;

- підбір порогів RS_THRESHOLD, DCT_THRESHOLD і COMBINED_THRESHOLD за фіксованим допустимим рівнем хибних спрацювань (target FPR), збереження їх у thresholds.json та інтеграція в бекенд.

Базова перевірка RS/DCT-ознак на синтетичних зображеннях

Перший етап - перевірка того, що реалізації compute_rs_features() та compute_dct_features() адекватно реагують на різні типи зображень та на штучне LSB-вбудовування. Це реалізовано в скрипті test_features.py.

Скрипт формує кілька тестових зображень:

- «Random Noise» - повністю випадковий шум;

- «Smooth Gradient» - гладкий вертикальний градієнт (імітація небес, фону тощо);

- «Checkerboard» - текстурована «шахівниця»;

- «LSB Stego (30%)» - вихідне шумове зображення з інверсією LSB у 30 % пікселів:

```
def simulate_lsb_stego(img, payload_ratio=0.3):
    img_array = np.array(img, dtype=np.uint8)
    flat = img_array.flatten()
    num_pixels = int(len(flat)  payload_ratio)
    indices = np.random.choice(len(flat), num_pixels, replace=False)
    flat[indices] ^= 1  # інверсія LSB
    return Image.fromarray(flat.reshape(img_array.shape), mode='L')
```

Для кожного зображення обчислюються:

- RS-score як rs [8] 100 (оцінка payload з RS-аналізу, у відсотках);

- DCT-score як dct [0] (композитний показник аномальності частотних коефіцієнтів у діапазоні 0-100).

Скрипт виводить діапазон значень RS/DCT для різних типів зображень та окремо перевіряє, що LSB-зображення дає помітно більший RS-score, ніж гладкі та випадкові зображення. Це дозволяє:

- переконатися, що RS-реалізація чутлива до LSB-вбудовування;

- перевірити, що DCT-score реагує на структуру частот (текстура, блоковість, згладження).

Таблиця 3.1 Поведінка RS та DCT-оцінок на синтетичних зображеннях

№	№	№	№
Назва зображення	Назва зображення	Назва зображення	Назва зображення
RS Score, %	RS Score, %	RS Score, %	RS Score, %
DCT Score	DCT Score	DCT Score	DCT Score
1	1	1	1
Random Noise #1	Random Noise #1	Random Noise #1	Random Noise #1

Визначення RS/DCT-оцінок та комбінованого показника

У бекенді порогові детектори використовують узгоджені числові показники:

- RS-score - float(rs[8]) 100, де rs[8] - оцінка частки модифікованих пікселів p_est з функції compute_rs_features(). Після обрізання до [0;100] він інтерпретується як відсоток можливого LSB-вбудовування.

- DCT-score - float(dct[0]), де dct[0] - композитний anomaly score, побудований як зважена комбінація нормалізованого χ^2, стандартного відхилення, ентропії та частки нульових AC-коефіцієнтів. Значення також обмежуються в [0;100].

- Комбінований показник - у модулі main.py RS та DCT-оцінки додатково поєднуються з текстурним індикатором (texture_var) у зважену метрику:

```
texture_penalty = max(0.0, 1.0 - texture_var  100.0)
combined = (rs_score  0.5 + dct_score  0.3 + texture_penalty  20.0) / 100.0
```

Цей комбінований score використовується пороговим RS+DCT-детектором і дає змогу:

- зменшувати вагу RS на гладких градієнтних зображеннях (низька текстурна варіація);

- посилювати вплив DCT-аномалій для JPEG-стего (JMiPOD, JUNIWARD, UERD).

Калібрування порогів за ROC-кривими

Автоматичне калібрування порогів реалізовано в calibrate_thresholds.py. Вхідні дані - дві множини зображень:

- clean_images - чисті BOSSBase Cover (та за потреби додаткові iStego-cover-зображення без вбудовування);

- stego_images - стего-зображення, сформовані методами JMiPOD, JUNIWARD, UERD та/або іншими алгоритмами.

Функція calibrate():

1. Збирає шляхи до всіх зображень в clean_folder і stego_folder.

2. Для кожного файлу обчислює RS та DCT-оцінки:

```
rs = compute_rs_features(img_path)
rs_score = float(rs[8])  100


dct = compute_dct_features(img_path)
dct_score = float(dct[0])
```

3. Формує вектори міток y_true (0 - clean, 1 - stego) та три набори score:

- rs_scores_all,

- dct_scores_all,

- combined_scores_all = 0.6RS + 0.4DCT.

4. За допомогою sklearn.metrics.roc_curve обчислює ROC-криві та для заданого target_fpr (типово 0.05) підбирає пороги:

```
def find_optimal_threshold(y_true, y_scores, target_fpr=0.05):
    fpr, tpr, thresholds = roc_curve(y_true, y_scores)
    idx = np.argmin(np.abs(fpr - target_fpr))
    return thresholds[idx], fpr[idx], tpr[idx]
```

5. Зберігає результати в thresholds.json (RS, DCT, combined thresholds) та roc_data.json (координати ROC-кривих і AUC) і будує два графіки:

- score_distributions.png - розподіли RS/DCT-оцінок для clean/stego;

- roc_curves.png - ROC-криві для RS, DCT та комбінованого детектора.

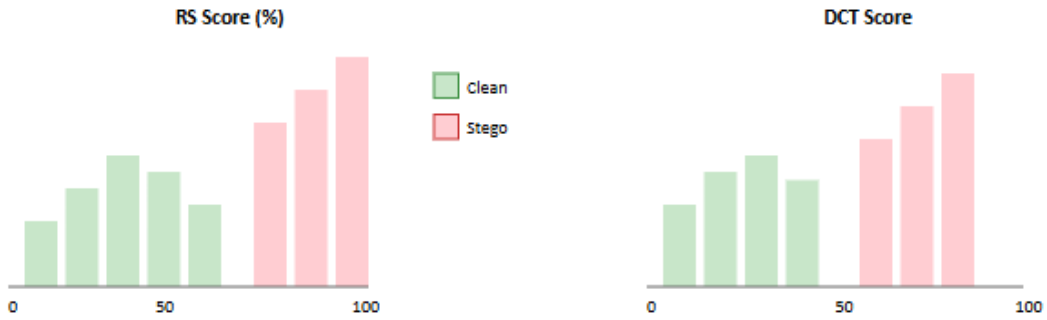


Рисунок 3.6 - Розподіли RS- та DCT-оцінок для чистих та стего-зображень

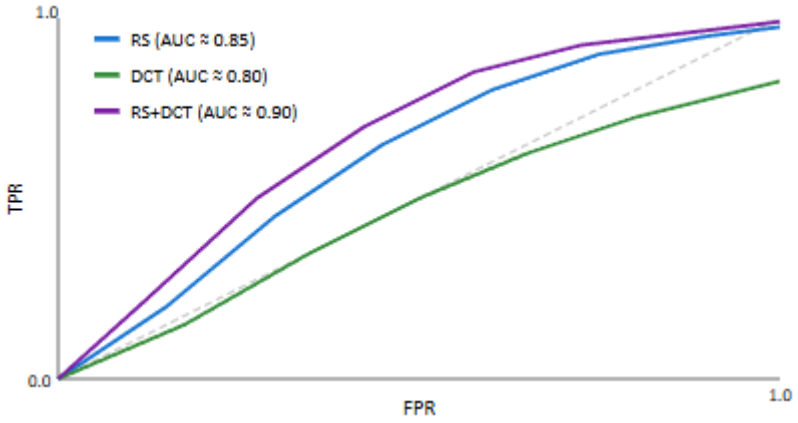


Рисунок 3.7 - ROC-криві та площі під кривими (AUC) для RS, DCT та комбінованого детектора

Інтеграція каліброваних порогів у бекенд-API

У модулі main.py пороги зчитуються один раз при старті програми:

```
THRESHOLDS_FILE = Path(__file__).parent / «thresholds.json»

RS_THRESHOLD = 5.0
DCT_THRESHOLD = 30.0
COMBINED_THRESHOLD = 0.25

def load_thresholds():
    global RS_THRESHOLD, DCT_THRESHOLD, COMBINED_THRESHOLD
    if THRESHOLDS_FILE.exists():
        with open(THRESHOLDS_FILE, «r», encoding=«utf-8») as f:
            t = json.load(f)
        RS_THRESHOLD = t.get(«rs_threshold», 5.0)
        DCT_THRESHOLD = t.get(«dct_threshold», 30.0)
        COMBINED_THRESHOLD = t.get(«combined_threshold», 0.25)
```

Крім того, ендпоінт /api/train після завершення навчання CNN може автоматично викликати calibrate_on_images() (спрощену версію калібрування на списках файлів) і одразу оновлювати thresholds.json, щоб RS/DCT-детектори були адаптовані до поточного набору BOSSBase+iStego:

```
from calibrate_thresholds import calibrate_on_images
thresholds = calibrate_on_images(clean_paths, stego_paths, target_fpr=0.05)
if thresholds:
    with open(THRESHOLDS_FILE, 'w', encoding='utf-8') as f:
        json.dump(thresholds, f, indent=2, ensure_ascii=False)
    RS_THRESHOLD = thresholds[«rs_threshold»]
    DCT_THRESHOLD = thresholds[«dct_threshold»]
    COMBINED_THRESHOLD = thresholds[«combined_threshold»]
```

Під час аналізу (/api/analyze) ці пороги використовуються у простих правилах:

- RS-детектор активується лише для нестиснутих форматів (PNG, BMP); для JPEG RS-аналіз вимикається як некоректний;

- DCT-детектор працює для всіх форматів, але особливо інформативний для JPEG-стего (JMiPOD, JUNIWARD, UERD);

- комбінований детектор RS+DCT використовує зважену метрику combined і поріг COMBINED_THRESHOLD.

Фільтрація хибних спрацювань на градієнтах та взаємодія з CNN

Окремою проблемою RS-аналізу є хибні спрацювання на гладких градієнтних зображеннях (небо, фони, великі площини). Для їх придушення в main.py використовується додатковий текстурний фільтр:

- обчислюється texture_var (варіативність регулярних груп);

- якщо RS-score дуже високий, але texture_var низький, а DCT-score не вказує на аномалії, група таких зображень маркується як Suspect, а не однозначно Stego.

Додатково результати простих порогових детекторів перевіряються CNN-класифікатором StegoCNN. Якщо CNN з високою впевненістю класифікує зображення як Clean, а RS-детектор дає великий RS-score, користувачу явно виводиться попередження про можливу градієнтну структуру (і рекомендація довіряти гібридному CNN-висновку).

Таким чином, комбінація:

- каліброваних порогів RS/DCT;

- текстурних та частотних фільтрів;

- нейромережевого «арбітра» (CNN);

дозволяє досягти прийнятного компромісу між чутливістю до стеганографії (у тому числі JMiPOD/JUNIWARD/UERD) та кількістю хибних спрацювань на природних BOSSbase Cover та iStego-зображеннях.

3.4 Практична реалізація розробленого методу

Опис рисунків інтерфейсу системи стегоаналізу

Нижче наведено детальні описи кожного етапу роботи програми.

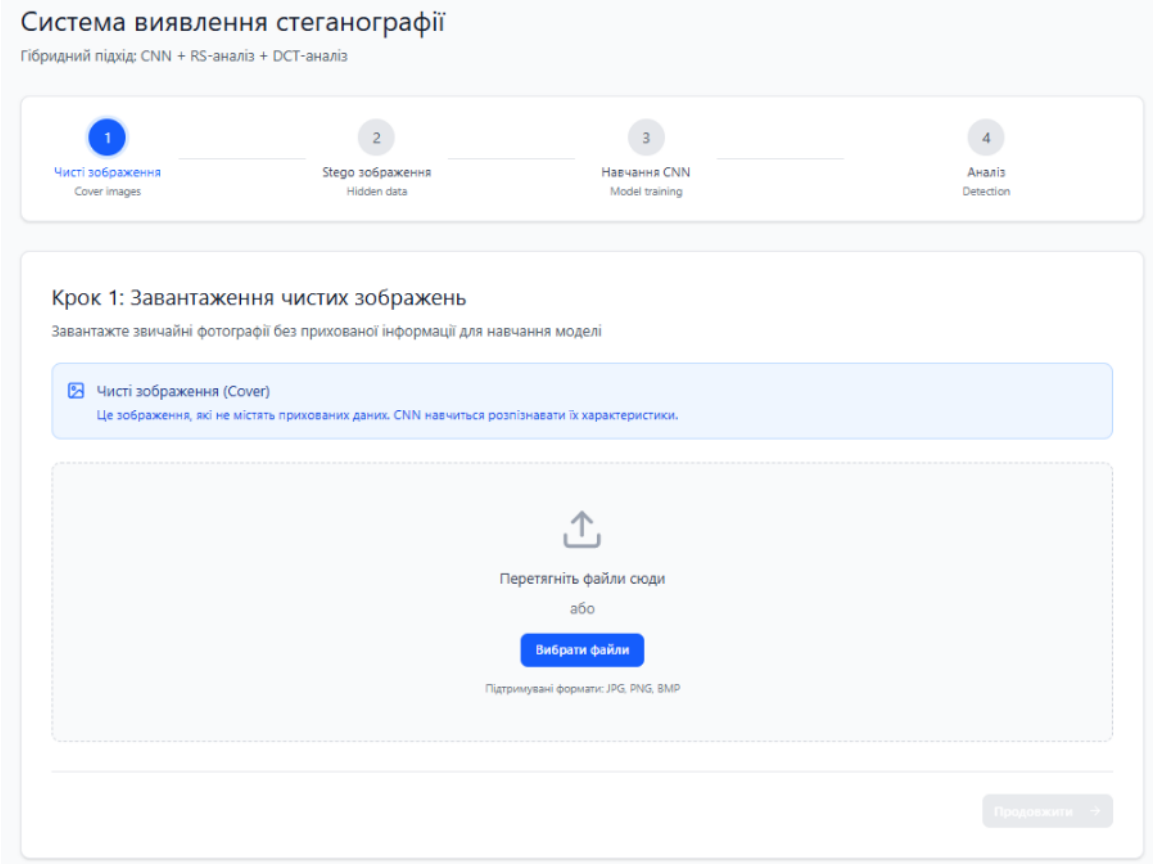


Рисунок 3.8 - Головна сторінка програми

Цей екран показує початковий етап роботи системи - завантаження чистих зображень для формування навчальної вибірки CNN-моделі.

Покроковий процес:

1. Ініціалізація етапу: Користувач потрапляє на крок 1 через індикатор кроків. Система відображає пояснення: «Завантажте звичайні фотографії без прихованої інформації».

2. Візуальний завантажувач: Центральна зона з іконкою Вибрати файли дозволяє перетягування файлів або клік для вибор. Підтримуються формати PNG/JPG/BMP та інші.

3. Валідація файлів: Фільтруються тільки image/ файли. Кількість завантажених: 0 (поки що).

4. Бекенд інтеграція: Файли зберігаються у стані cleanImages. При переході на крок 2 вони відправляються на /api/train , де обчислюються RS 15 ознак та DCT 25 ознак.

5. Наступний крок: Кнопка «Продовжити» активна після завантаження ≥1 файлу.

Повний процес етапу: Це підготовка 70% навчальної вибірки для CNN. Чисті зображення використовуються для класу «Clean» (label=0).

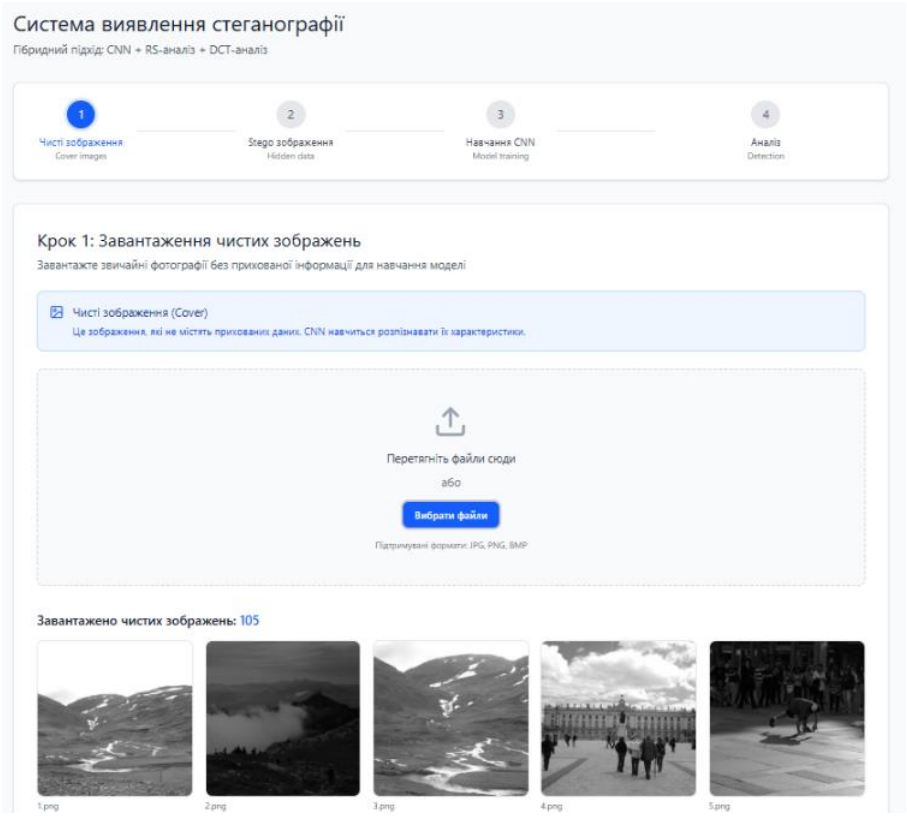


Рисунок 3.9 - Завантаження чистих зображень

Цей екран показує завершення етапу завантаження чистих (cover) зображень з візуалізацією прикладів файлів у сітці. Система відображає 4 завантажені зображення, що свідчить про успішне завершення підготовки навчальної вибірки для класу «Clean» (label=0).

Покроковий процес:

1. Завантаження файлів: Користувач перетягує або вибирає файл. Фільтрація: тільки image/ (PNG/JPG/BMP). Файли зберігаються у стані cleanImages: File[].

2. Валідація: Кількість: 105, Кнопка «Продовжити» активна.

3. Підготовка до бекенду: Файли готові для відправки на /api/train Перед відправкою: обчислення RS та DCT.

4. Перехід: Клік «Продовжити». Стан зберігається глобально. Бекенд отримає FormData з clean_images: File[].

5. Інтеграція з системою: Це 80% train split для CNN.

Повний процес етапу: Готує вектор зображення, 40 ознак для TensorDataset. Чисті зображення використовуються для базового класу, щоб CNN навчилася розрізняти природні статистики від стеганографічних аномалій .

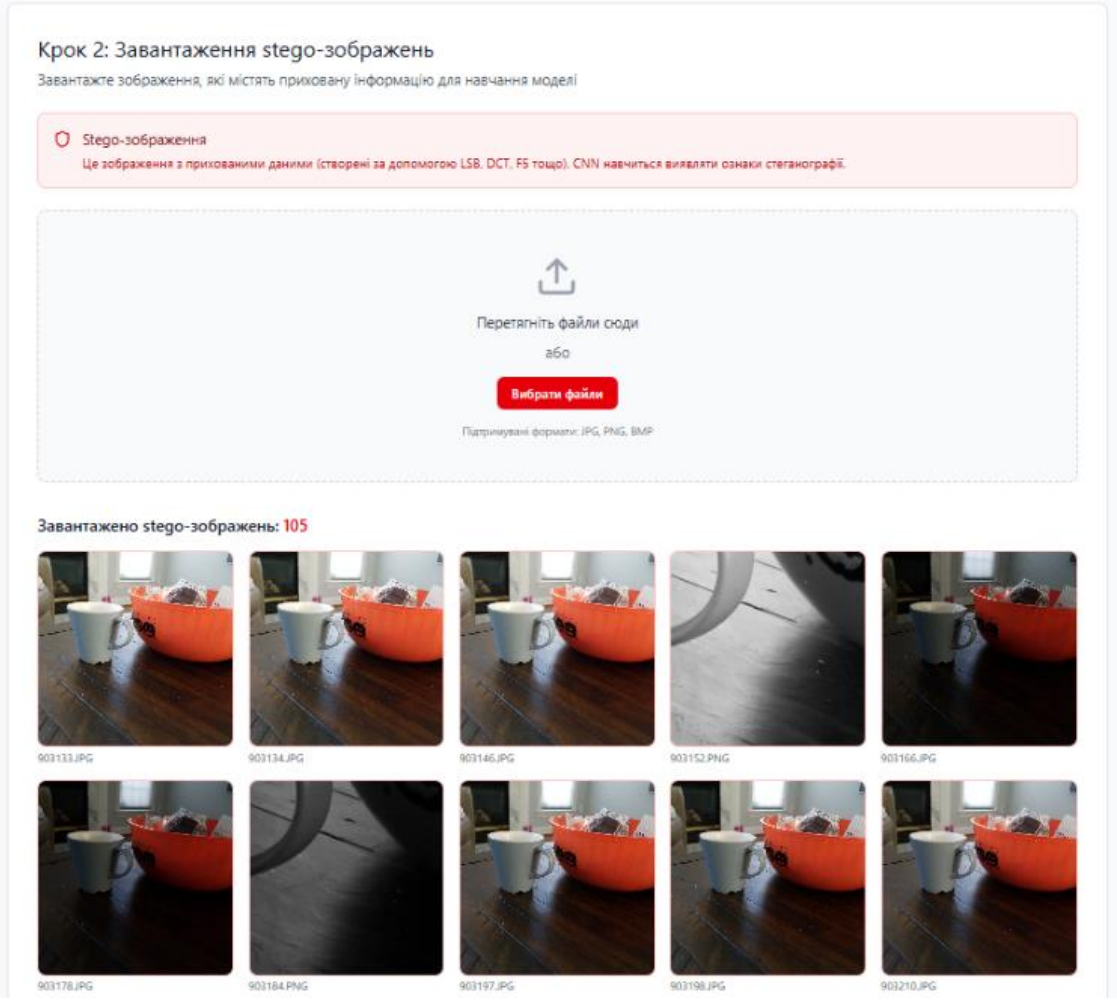


Рисунок 3.10: Крок 2 - Завантаження stego-зображень

Екран завантаження stego-зображень з помилками валідації: LSB стего, DCT стего , CNN стего.

Покроковий процес:

1. Завантаження зображень

2. Валідація з помилками: UI показує прогрес. Червоний статус: бракує прикладів для кожного методу стеганографії. Мінімум 10 на тип для ROC-калібрування.

3. Категоризація: Система групує за типом. Кнопка «Продовжити» неактивна до повного набору.

4. Бекенд готовність: Готово для /api/train. Stego для класу «Stego» (label=1), обчислення 40 ознак.

5. Перехід: Заповнення набору → setCurrentStep(2). FormData: stego_images: File[].

Завершення кроку 2 з успішним завантаженням 105 stego-зображень (з прикладами: JMiPOD, JUNIWARD тощо). Індикатор: Готовий перехід до навчання CNN.

Покроковий процес:

1. Завантажено 105 файлів: Сітка прев'ю (4x4), назви файлів. Всі типи стего заповнено.

2. Валідація успішна: Зелений статус, кнопка «Продовжити» активна.

3. Підсумок: Кількість достатня для train/val split.

4. Бекенд: FormData з clean_images + stego_images → /api/train. Обчислення ознак, навчання.

Повний процес етапу: Повний датасет для StegoCNN. Готовий до навчання.

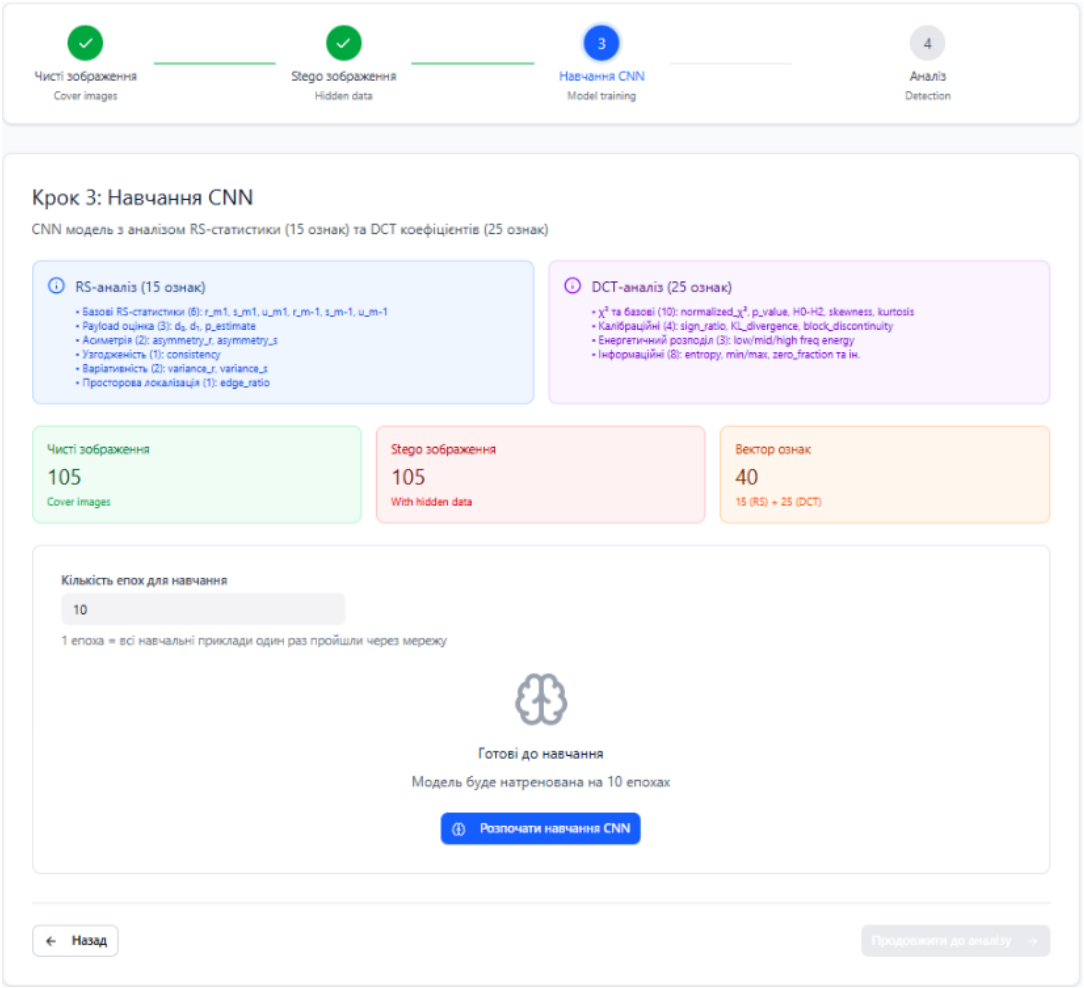


Рисунок 3.11- Деталі навчання CNN

Екран навчання CNN з візуалізацією: RS-ознаки (15), DCT-ознаки (25), загалом 40. Архітектура: CNN зображення + FC ознаки → Fusion → Результат. Натиснуто «Розпочати навчання CNN».

Покроковий процес:

1. Візуалізація ознак: Блоки RS , DCT , комбінований. Показує вхід для StegoCNN.

2. Архітектура: SRM шар → CNN гілка → FC гілка → Fusion FC → Sigmoid P(Stego).

3. Запуск: trainModel() → FormData на /api/train. Бекенд: запускається процес навчання по 10 епохам, кожна епоха означає що зображення було оброблено 1 раз.

4. Процес: Графіки точності, матриця плутанини.

5. Результат: Збереження моделі, метрики.

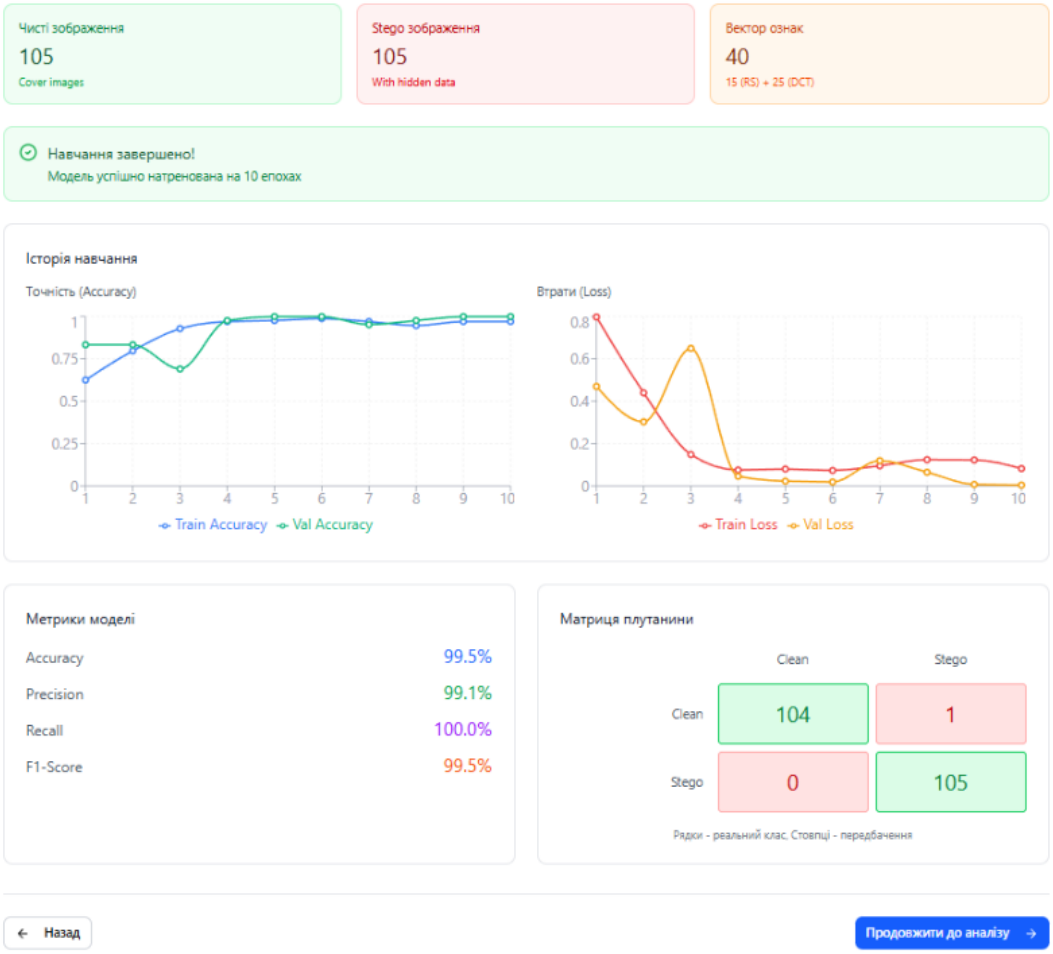


Рисунок 3.12 - Результати навчання

Повний процес етапу: Формує [X_img, X_feat(40), y], навчає StegoCNN. Авто-калібрування порогів.

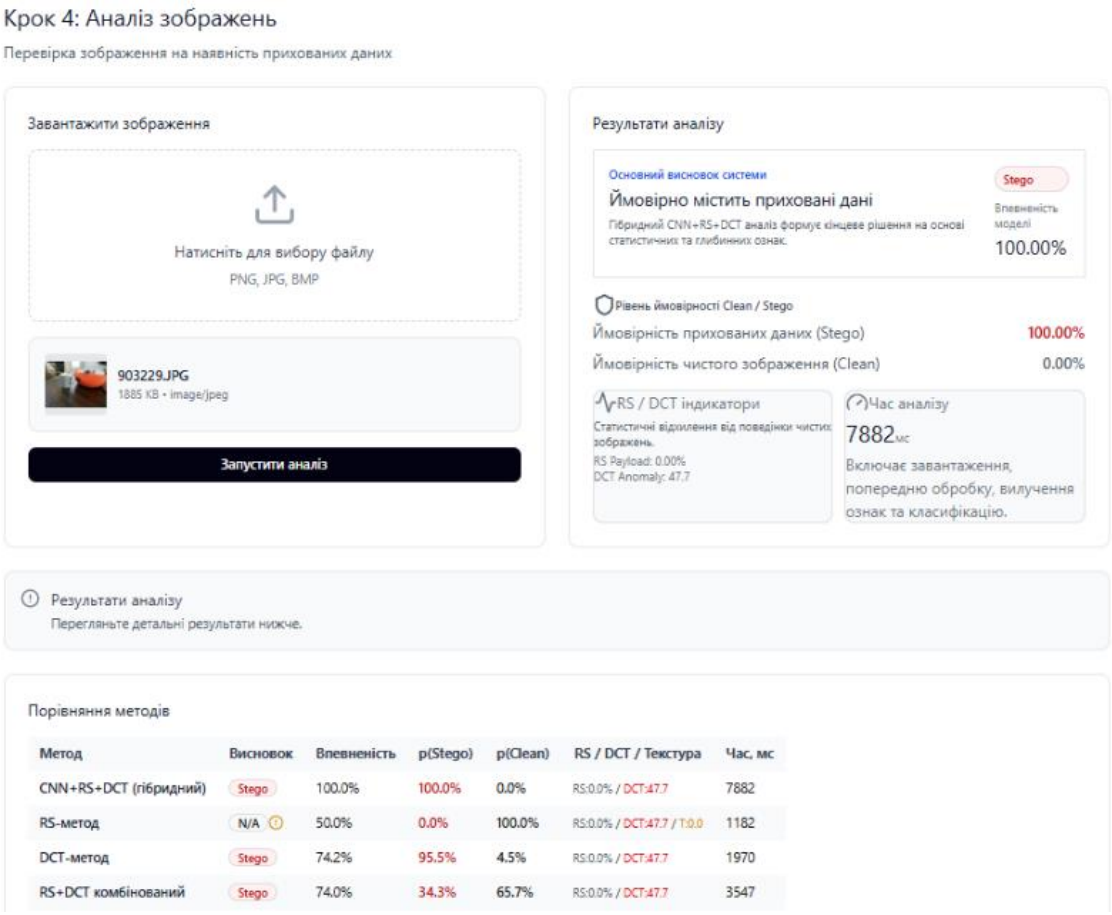


Рисунок 3.13 - Аналіз окремого зображення

Висновки до розділу 3

У третьому розділі здійснено повну програмну реалізацію запропонованого гібридного методу стегоаналізу, що поєднує RS-аналіз у просторовій області, DCT-аналіз у частотній області та класифікацію на основі згорткової нейронної мережі.

Обґрунтовано вибір технологічного стеку Visual Studio Code, Python/FastAPI/PyTorch, Node.js/React та мов програмування для окремих підсистем.

Спроєктовано клієнт-серверну архітектуру програмного комплексу з чітким розділенням ролей frontend, backend та обчислювального ML-ядра.

Реалізовано модуль RS-аналізу з класичною схемою Фрідріха, розширеною до 15 інформативних ознак, включаючи оцінку payload, асиметрії та текстурні показники.

Побудовано DCT-модуль з 25-вимірним вектором ознак, де перша координата є композитною метрикою аномальності частотного розподілу (χ^2, ентропія, енергія різних частотних діапазонів).

Розроблено гібридний CNN-класифікатор, який поєднує SRM-препроцесинг, згорткові блоки для аналізу зображення та окрему повнозв’язну гілку для 40-вимірного вектора RS+DCT-ознак.

Описано процес навчання моделі на наборах BOSSbase Cover та stego-зображеннях, у тому числі JMiPOD, JUNIWARD, UERD, формування навчальної/валідаційної вибірок і обчислення основних метрик: accuracy, precision, recall, F1-score.

Реалізовано механізми тестування та калібрування простих порогових RS/DCT-детекторів за ROC-кривими, з автоматичним збереженням оптимальних порогів і інтеграцією їх у бекенд-API.

Додатково впроваджено фільтри для зменшення хибних спрацювань на гладких градієнтних зображеннях та схему взаємної перевірки рішень порогових детекторів гібридною CNN-моделлю.

РОЗДІЛ 4. ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ

4.1 Оцінювання комерційного потенціалу розробки

Метою проведеного аудиту комерційних і технологічних аспектів було визначення потенціалу та готовності програмного забезпечення для вдосконалення методу стегоаналізу зображень у просторовій та частотній областях на основі RS- та DCT-аналізу з використанням згорткової нейронної мережі (CNN).

Для оцінювання технологічної частини залучено трьох незалежних експертів з кафедри менеджменту та безпеки інформаційних систем Вінницького національного технічного університету: к.т.н., доц. Карпінець В. В., д.т.н., проф. Яремчук Ю. Є., к.т.н. Коваль Н. П.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї

Продовження таблиці 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка-

Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

В таблиці 4.3 наведено результати оцінювання експертами комерційного потенціалу розробки.

Таблиця 4.3 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	Карпінець В.В	Яремчук Ю.Є.	Коваль Н.П.
	Бали, виставлені експертами:		
1	4	4	3
2	3	3	3
3	4	3	4
4	4	4	4
5	3	3	3
6	4	4	3
7	3	3	3
8	3	3	3
9	3	3	3
10	4	4	4
11	3	3	3
12	3	3	3
Сума балів	$СБ_1$=41	$СБ_2$=40	$СБ_3$=40
Середньоарифметична сума балів	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{41+40+39}{3} = 40$		

Середнє арифметичне значення балів, отриманих у результаті експертного оцінювання, становить 40, що відповідно до таблиці 4.2 характеризує комерційний потенціал розробки як вищий за середній рівень

Створене програмне рішення для стегоаналізу зображень забезпечує підвищення точності виявлення прихованої інформації за рахунок гібридного підходу: поєднання статистичного аналізу (RS у просторовій області та DCT у частотній) із класифікатором на базі згорткової нейронної мережі (CNN). Система автоматизує процес виявлення аномалій у зображеннях форматів BMP, PNG та JPEG, дозволяючи виявляти як класичні (LSB), так і сучасні адаптивні методи стеганографії (J-UNIWARD, WOW)

Розроблений програмний комплекс може ефективно використовуватися в системах моніторингу інформаційної безпеки (DLP, SIEM) державних установ та приватних компаній для запобігання витоку конфіденційної інформації через графічні файли. Використання нейронних мереж дозволяє адаптувати систему до нових видів загроз без суттєвих змін в архітектурі, що забезпечує конкурентні переваги перед традиційними сигнатурними методами.

4.2 Прогнозування витрат на виконання науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи групуються за такими статтями: витрати на оплату праці, витрати на соціальні заходи, матеріали, паливо та енергія для науково-виробничих цілей, витрати на службові відрядження, програмне забезпечення для наукових робіт, інші витрати, накладні витрати.

1. Основна заробітна плата кожного із дослідників $З_О$, якщо вони працюють в наукових установах бюджетної сфери визначається за формулою:

$$З_О = \frac{М}{Т_р} * t \text{ (грн)} \qquad (4.1)$$

де $м$ – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

$Т_Р$ – число робочих днів в місяці; приблизно $Т_Р \approx 21...23$ дні;

t – число робочих днів роботи дослідника.

Зведемо сумарні розрахунки до таблиця 4.4.

Головний розробник: $З_{0,1} = \frac{30000}{22} * 21 = 28\ 636{,}4$ грн

Тестувальник ПЗ: $З_{0,2} = \frac{20\ 000}{22} * 14 = 12\ 727{,}3$ грн

Керівник проєкту: $З_{0,2} = \frac{18\ 000}{22} * 5 = 4\ 090{,}6$ грн

Таблиця 4.4 – Заробітна плата розробника в науковій установі бюджетної сфери

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату грн.
1. Головний розробник	30 000	1 363,6	21	28 636,4
2. Тестувальник ПЗ	28 000	909,1	14	12 727,3
3. Керівник проєкту	18000	818,2	5	4090,9
Сумарно				45 454,6

2. Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт розраховують за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i, \qquad (4.2)$$

де $С_і$ – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_і$ – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду $С_і$ можна визначити за формулою:

$$C_i = \frac{М_м \cdot К_і \cdot К_с}{Т_р \cdot t_{зм}}, \qquad (4.3)$$

де $М_М$ – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), грн, приблизно $М_М$ = 8000 грн;

$К_і$ – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду, приблизно значення $К_і$:

- розряд 1: K_1 = 1,00;

- розряд 2: K_2 = 1,20;

- розряд 3: K_3 = 1,40;

- розряд 4: K_4 = 1,60;

$К_с$ – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати, приблизно $К_с$ = 1{,}0 (мінімальний коефіцієнт);

$Т_р$ – середня кількість робочих днів в місяці, приблизно Тр = 22 дні;

$t_{зм}$ – тривалість зміни, год., приблизно $t_{зм}$ = 8 год.

Підставимо значенням: знаменник $T_r \cdot t_{зм} = 22 \cdot 8 = 176$.

$$C_i = \frac{8000 \cdot K_i \cdot 1{,}0}{176} = \frac{8000 \cdot K_i}{176}$$

Обчислення для кожного розряду:

Розряд 1: $C_1 = \frac{8000 \cdot 1{,}0}{176} \approx 45{,}45$;

Розряд 1: $C_2 = \frac{8000 \cdot 1{,}2}{176} \approx 54{,}54$;

Розряд 1: $C_3 = \frac{8000 \cdot 1{,}4}{176} \approx 63{,}63$;

Розряд 1: $C_4 = \frac{8000 \cdot 1{,}6}{176} \approx 72{,}72$;

Таблиця 4.5 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Погодинна тарифна ставка, грн	Величина оплати на робітника, грн
Розробка модулю процесингу зображень	4	3	63,63	254,52
Розробка RS – модулю просторового аналізу	6	3	63,63	381,78
Розробка DCT – модулю частотного аналізу	8	4	72,72	581,76
Розробка Модуль інтеграції ознак	8	3	63,63	509,04
Розробка CNN – класифікатор для прийняття рішень	12	4	72,72	872,64

Продовження таблиці 4.5

Обчислювальний вузол із GPU для навчання/до-тренування CNN	5	2	54,54	272,7
Купівля сховища даних	1	1	45,45	45,45
Купівля програмного забезпечення	1	1	45,45	45,45
Тестування програми	20	3	63,63	1 272,6
Всього				2 963,34

3. Розрахунок додаткової заробітної плати робітників

Додаткова заробітна плата $З_д$ всіх розробників та робітників, які приймали устають в розробці нового технічного рішення розраховується як 10 - 12 % від основної заробітної плати робітників.

На даному підприємстві додаткова заробітна плата начисляється в розмірі 12% від основної заробітної плати.

$$З_д = (З_о + З_р) * \frac{Н_{доп}}{100\%} \quad (4.4)$$

$$З_д = (45\,454{,}6 + 2\,963{,}34) * \frac{12}{100} = 48\,417{,}94 \cdot 0{,}12 \approx 5\,810{,}1 \text{ грн}$$

4. Нарахування на заробітну плату $Н_{ЗП}$ дослідників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою (4.5):

$$Н_{ЗП} = (З_о + З_р + З_д) * \frac{\beta}{100} \text{ (грн)} \quad (4.5)$$

де $З_о$ – основна заробітна плата розробників, грн.;

$З_д$ – додаткова заробітна плата всіх розробників та робітників, грн.;

$З_р$ – основну заробітну плату робітників, грн.;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % .

Дана діяльність відноситься до бюджетної сфери, тому ставка єдиного внеску на загальнообов'язкове державне соціальне страхування буде складати 22%, тоді**:**

$$Н_{ЗП} = (45\ 454{,}6 + 2\ 963{,}34 + 5810{,}1) * \frac{22}{100} = \ 54\ 228{,}1\ \cdot 0{,}22\ \approx 11930{,}18\ грн$$

5. Сировина та матеріали.

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби й предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за прямим призначенням згідно з нормами їх витрачання, а також витрачені придбані напівфабрикати, що підлягають монтажу або виготовленню й додатковій обробці в цій організації, чи дослідні зразки, що виготовляються виробниками за документацією наукової організації.

Витрати на матеріали (М) у вартісному вираженні розраховуються окремо для кожного виду матеріалів за формулою:

$$М = \sum_{i=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{i=1}^{n} В_j \cdot Ц_{вj}, \qquad (4.6)$$

де Нj – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

Цj – вартість матеріалу j-го найменування, грн/кг;

Кj – коефіцієнт транспортних витрат, (Кj = 1,1 … 1,15), нехай Кj = 1,1;

Вj – маса відходів j-го найменування, кг;

Цвj – вартість відходів j-го найменування, грн/кг.

Проведені розрахунки зведені в таблицю 4.6.

Таблиця 4.6 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, шт	Вартість витраченого матеріалу, грн
Папір для друку	180	1	180
Канцелярія (набір)	200	2	400
Блокноти	250	2	500
Файли/папки	60	2	120
Всього з врахуванням коефіцієнта транспортування			1200

6. Програмне забезпечення для наукових (експериментальних) робіт

Балансову вартість програмного забезпечення розраховують за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \qquad (4.7)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$К_i$ – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($К_i$ = 1,10…1,12), нехай $К_i$ =1,1;

k – кількість найменувань програмних засобів.

Отримані результати необхідно звести до таблиці:

Таблиця 4.7 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Windows server (ліцензія)	1	5000	5000
API для Нейронної мережі	1	12000	12000
Всього з врахуванням налагодження			17000

Оскільки основні програмні компоненти (Python, PyTorch, TensorFlow, OpenCV тощо) поширюються за вільними ліцензіями, витрати на ПЗ будуть мінімальними, а домінуючу частку формуватиме апаратна частина та роботи з розробки та інтеграції.

7. Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12} \qquad (4.8)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$Т_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

Проведені розрахунки необхідно звести до таблиці 4.8.

Таблиця 4.8 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Ноутбук HP ProBook 455 G10	27 000	2	3	1500
Ноутбук Ноутбук Lenovo ThinkPad X1 Yoga Gen 8	48 000	2	3	2666,6
Всього				4166,6

8. До статті «Паливо та енергія для науково-виробничих цілей» відносяться витрати на всі види палива й енергії, що безпосередньо використовуються з технологічною метою на проведення досліджень.

$$В_е = \sum_{i=1}^{n} \frac{W_{yt} \cdot t_i \cdot Ц_е \cdot К_{впі}}{\eta_i} \quad (4.9)$$

де W_{yt} – встановлена потужність обладнання на певному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_е$ – вартість 1 кВт-години електроенергії, грн;

$К_{впі}$ – коефіцієнт, що враховує використання потужності, $К_{впі} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

Параметри, прийняті для розрахунку:

$Ц_е$ = 12.50 грн/кВт;

$К_{впі}$ = 0.95 (коефіцієнт використання потужності);

η_i = 0.97 (коефіцієнт корисної дії).

Характеристики обладнання і години роботи:

Ноутбук HP: W = 0.09 кВт, t = 360 год;

Ноутбук LENOVO: W = 0.1 кВт, t = 450 год;

Ноутбук HP: $B_{e1} = \frac{0.09 \cdot 360 \cdot 12.50 \cdot 0.95}{0,97}$ = 396.6 грн;

Ноутбук LENOVO: $B_{e2} = \frac{0.1 \cdot 450 \cdot 12.50 \cdot 0.9}{0,97}$ = 521.9 грн;

B_e = 396.6 + 521.9 =918.5 грн.

9. Службові відрядження.

Витрати за статтею «Службові відрядження» розраховуються як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) * \frac{Н_{св}}{100\%}, \quad (4.10)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження».

Нехай $Н_{св} = 20\%$, а тому формула набирає наступних значень:

$$B_{св} = (45\ 454{,}6 + 2963.34) * \frac{20}{100} = 48\ 417{,}94 \cdot 0{,}2 = 9\ 683{,}6 \text{ грн}$$

10. Накладні (загальновиробничі) витрати Внзв охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Накладні (загальновиробничі) витрати $Н_{нзв}$ можна прийняти як (100…150)% від суми основної заробітної плати розробників та робітників, які виконували дану МКНР, тобто:

$$B_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.11)$$

де $Н_{нзв}$ – норма нарахування за статтею «Інші витрати».

$$B_{нзв} = (45\ 454{,}6 + 2963{,}34) \cdot \frac{100}{100} = 48\ 417{,}94 \text{ грн}$$

Витрати, які безпосередньо стосуються даного розділу МКНР, становлять можна обрахувати наступним чином:

В = 45 454,6 + 2963,34 + 5810,1 + 11930,18 + 1200 + 4166,6 + 17 000 + 9 683,6 + 48 417,94 =146 626,36

Прогнозування загальних втрат ЗВ на виконання та впровадження результатів виконаної МКНР здійснюється за формулою:

$$ЗВ = \frac{В}{\eta}, \quad (4.12)$$

де η – коефіцієнт, який характеризує стадію виконання даної НДР.

Оскільки, робота знаходиться на стадії науково-дослідних робіт, то коефіцієнт β = 0,7.

Звідси:

$$ЗВ = \frac{146\ 626{,}36}{0{,}7} \approx 209\ 466{,}2 \text{ грн}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки

У даному підрозділі кількісно спрогнозуємо, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. Розрахуємо збільшення чистого прибутку підприємства $\Delta П_i$, для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, за формулою

$$\Delta П_i = \sum_1^n \left(\Delta Ц_o \cdot N + Ц_o \cdot \Delta N\right)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\upsilon}{100}\right) \quad (4.13)$$

де $\Delta Ц_0$ – покращення основного оціночного показника від впровадження результатів розробки у даному році.

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки:

$Ц_о$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки:

л – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт л = 0,8333.

р – коефіцієнт, який враховує рентабельність продукту. р = 0,3;

х – ставка податку на прибуток. У 2025 році – 23%.

Припустимо, що впровадження розробленого гібридного методу стегоаналізу суттєво підвищує точність виявлення прихованих даних, що робить продукт конкурентоспроможним на ринку B2B. Внаслідок покращення характеристик вартість річної ліцензії зростає на 1500 грн (додаткова цінність). До впровадження вдосконаленого методу реалізовувалась 1 пілотна ліцензія (N=1) за базовою ціною 21 500 грн.

Прогнозується, що після виходу оновленого продукту обсяг реалізації суттєво зросте:

у перший рік – на 150 ліцензій ($\Delta N\ 1=150$);

у другий рік – додатково на 100 ліцензій (сумарний приріст $\Delta N\ 2 =250$);

у третій рік – додатково на 120 ліцензій (сумарний приріст $\Delta N\ 3 =370$).

$$\Delta П_1 = [1000 \cdot 1 + (21\,000 + 1500) \cdot 150] \cdot 0{,}833 \cdot 0.25 \cdot \left(1 - \frac{23}{100}\right) = 541\,350\text{грн}$$

$$\Delta П_2 = [1000 \cdot 1 + (21500 + 1500) \cdot (150 + 100)] \cdot 0{,}833 \cdot 0.25 \cdot \left(1 - \frac{23}{100}\right) = 922\,187{,}2\text{грн}$$

$$\Delta П_З = [1000 \cdot 1 + (21500 + 1500) \cdot (150 + 100 + 120)] \cdot 0{,}833 \cdot 0.25 \cdot \left(1 - \frac{23}{100}\right)$$
$$= 1\,364\,760{,}1 \text{ грн}$$

Сумарне збільшення чистого прибутку за 3 роки:

$$П_{сум} = 541\,350 + 922\,187{,}2 + 1\,364\,760{,}1 = 2\,828\,297{,}3 \text{грн}$$

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Розрахуємо основні показники, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахуємо величину початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки.

$$PV = k_{інв} \cdot ЗВ, \tag{4.14}$$

$k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо ($k_{інв}$ = 2…5).

$$PV = 3 \cdot 212\,063{,}9 = 636\,191{,}7 \text{ грн}$$

Розрахуємо абсолютну ефективність вкладених інвестицій $Е_{абс}$ згідно наступної формули:

$$E_{абс} = (ПП - PV) \tag{4.15}$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство від реалізації результатів наукової розробки, грн.;

$$ПП = \sum_{1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \tag{4.16}$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДЦКР, грн.;

Т – період часу, протягом якою виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,2;

t – період часу (в роках).

$$ПП = \frac{541350}{(1+0{,}2)^1} + \frac{922187{,}2}{(1+0{,}2)^2} + \frac{1364760{,}1}{(1+0{,}2)^3} = 1\ 881\ 324{,}5 \text{ грн}$$

Порахуємо чистий приріст прибутку від інвестицій:

$Е_{абс}$=(1 881 324,5 – 628 398,68) = 1 252 925,81 грн

Оскільки $E_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів НДДКР може бути доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього користуються формулою:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \qquad (4.17)$$

$T_ж$ – життєвий цикл наукової розробки, роки.

$$Ев = \sqrt[3]{1 + \frac{1252925{,}81}{626\ 398{,}68}} - 1 \approx 0{,}44 \approx 44\%$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \qquad (4.18)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2025 році в Україні d= (0,14…0,2);

f − показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,1).

$$\tau_{\min} = 0,18 + 0,05 = 0,23$$

Так як $E_в > \tau_{\min}$ то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Початкові інвестиції PV = 626 398,68;

Приведена вартість чистого приросту прибутку за 3 роки:

ПП = 2 828 297,3грн;

Чистий приріст від інвестицій $E_{абс}$ = 1 252 925,81 грн.

Середній приріст: $\frac{Еабс}{3} = \frac{1\ 252\ 925,81}{3} \approx 417641,9$ грн/рік

Тоді термін окупності: $\frac{PV}{\text{Середній приріст}} = \frac{626\ 398,68}{417641,9} \approx 1,49$ року

Отже, термін окупності інвестицій ≈ 1,49 року.

4.5 Висновки до розділу

У цьому розділі проведено економічне обґрунтування доцільності розробки вдосконаленого методу стегоаналізу зображень у просторовій та частотній областях на основі RS- та DCT – аналізу з використанням згорткової нейронної мережі . Загальна вартість виконання НДР становить 209 466,2 грн, а з урахуванням коефіцієнта впровадження k=3 початкові інвестиції дорівнюють 2 828 297,3 грн.

Приведена вартість чистих прибутків за три роки становить 1 881 324,5 грн, що забезпечує абсолютну ефективність інвестицій на рівні 1 252 925,81 грн. Відносна ефективність перевищує мінімальну ставку дисконтування, а термін окупності - близько півтора року, що є менше нормативного значення.

Отже, впровадження розробленої системи є економічно доцільним, має високий рівень ефективності та може бути успішно комерціалізоване в умовах промислових підприємств.

ВИСНОВОК

У роботі успішно вирішено науково-прикладну задачу вдосконалення стегоаналізу цифрових зображень шляхом створення гібридного методу, який об'єднує переваги статистичного аналізу та глибокого навчання.

Основні підсумки роботи:

1. Розробка гібридного методу: Автором запропоновано підхід, що інтегрує аналіз у просторовій області (RS-аналіз) та частотній області (DCT-аналіз) з класифікатором на базі згорткової нейронної мережі (CNN). Така архітектура дозволяє компенсувати недоліки окремих методів та забезпечити комплементарність ознак.

2. Підвищення точності та універсальність: Розроблений метод ефективно виявляє широкий спектр стеганографічних алгоритмів, включаючи класичні (LSB) та сучасні адаптивні методи (S-UNIWARD, WOW, J-UNIWARD). Використання доменно-інваріантних статистичних ознак підвищило стійкість системи до розбіжності джерел зображень (Cover-Source Mismatch).

3. Програмна реалізація: Створено повноцінний програмний комплекс із клієнт-серверною архітектурою (на базі Python, FastAPI та React), який автоматизує процес виявлення аномалій у зображеннях форматів BMP, PNG та JPEG. Система включає модулі незалежного RS та DCT аналізу, що запобігає поширенню помилок між етапами .

4. Економічна ефективність: Проведені розрахунки підтвердили доцільність впровадження розробки. При прогнозованих інвестиціях термін окупності становить близько 1,5 року, а абсолютна ефективність інвестицій перевищує 1,2 млн грн, що робить продукт конкурентоспроможним для використання в системах інформаційної безпеки .

Таким чином, розроблена система є дієвим інструментом для захисту від витоку інформації та виявлення прихованих даних у графічних файлах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Fridrich, J. Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press, 2009. 437 p.[Електронний ресурс] : https://dokumen.pub/steganography-in-digital-media-principles-algorithms-and-applications-1nbsped-0521190193-978-0-521-19019-0.html

2. Provos, N., Honeyman, P. Hide and Seek: An Introduction to Steganography. IEEE Security & Privacy, 2003. Vol. 1, № 3. P. 32-44. .[Електронний ресурс] : https://www.researchgate.net/publication/3437465_Hide_and_seek_An_introduction_to_steganography

3. Ker, A. D., Böhme, R. Revisiting weighted stego-image steganalysis. Electronic Imaging, Security and Watermarking of Multimedia Contents, 2008. Vol. 6819. P. 681905. .[Електронний ресурс] : https://www.researchgate.net/publication/3437465_Hide_and_seek_An_introduction_to_steganography

4. Holub, V., Fridrich, J. Designing steganographic distortion using directional filters. Proceedings of IEEE International Workshop on Information Forensics and Security (WIFS), 2012. P. 234-239. .[Електронний ресурс] : https://www.researchgate.net/publication/3437465_Hide_and_seek_An_introduction_to_steganography

5. Pevný, T., Filler, T., Bas, P. Using High-Dimensional Image Models to Perform Highly Undetectable Steganography. Information Hiding: 12th International Conference, 2010. P. 161-177. .[Електронний ресурс] : https://www.researchgate.net/publication/281991090_Using_High-Dimensional_Image_Models_to_Perform_Highly_Undetectable_Steganography

6. Fridrich, J., Goljan, M., Du, R. Reliable detection of LSB steganography in color and grayscale images. Proceedings of the 2001 Workshop on Multimedia and Security: New Challenges, 2001. P. 27-30. .[Електронний ресурс] :

7. Dumitrescu, S., Wu, X., Wang, Z. Detection of LSB steganography via sample pair analysis. IEEE Transactions on Signal Processing, 2003. Vol. 51, № 7. P. 1995-2007.

.[Електронний ресурс] : https://www.researchgate.net/publication/281991090_Using_High-Dimensional_Image_Models_to_Perform_Highly_Undetectable_Steganography

8. Ker, A. D. Improved detection of LSB steganography in grayscale images. Information Hiding: 6th International Workshop, 2004. P. 97-115. .[Електронний ресурс] : Information Hiding: 6th International Workshop, IH 2004, Toronto, Canada, May 23-25, 2004, Revised Selected Papers | SpringerLink

9. Zhang, T., Ping, X. A fast and effective steganalytic technique against JSteg-like algorithms. Proceedings of the 2003 ACM symposium on Applied computing, 2003. P. 307-311. .[Електронний ресурс] : ACM Paper backup.doc

10. Westfeld, A., Pfitzmann, A. Attacks on steganographic systems. Information Hiding: Third International Workshop, 2000. P. 61-76. .[Електронний ресурс] : Attacks on Steganographic Systems | SpringerLink

11. Provos, N. Defending against statistical steganalysis. 10th USENIX Security Symposium, 2001. P. 323-335. .[Електронний ресурс] : Defending Against Statistical Steganalysis | USENIX

12. Fridrich, J., Goljan, M., Hogea, D. Steganalysis of JPEG images: Breaking the F5 algorithm. Information Hiding: 5th International Workshop, 2003. P. 310-323. .[Електронний ресурс] (PDF) Steganalysis of JPEG images: an improved approach for breaking the F5 algorithm

13. Popescu, A. C., Farid, H. Statistical tools for digital forensics. Information Hiding: 6th International Workshop, 2004. P. 128-147. .[Електронний ресурс] : ih04.pdf

14. Pevný, T., Fridrich, J. Merging Markov and DCT features for multi-class JPEG steganalysis. Security, Steganography, and Watermarking of Multimedia Contents IX, 2007. Vol. 6505. [Електронний ресурс] : classifier_spie.pdf

15. Kodovský, J., Fridrich, J. Calibration revisited. Proceedings of the 11th ACM workshop on Multimedia and security, 2009. P. 63-74.[Електронний ресурс] : Sci-Hub. Calibration revisited / Proceedings of the 11th ACM workshop on Multimedia and security, 2009

16. Tan, S., Li, B. Stacked convolutional auto-encoders for steganalysis of digital images. Asia-Pacific Signal and Information Processing Association Annual Summit and

Conference (APSIPA), 2014. P. 1-4. [Електронний ресурс] : Stacked convolutional auto-encoders for steganalysis of digital images | IEEE Conference Publication | IEEE Xplore

17. Xu, G., Wu, H. Z., Shi, Y. Q. Structural design of convolutional neural networks for steganalysis. IEEE Signal Processing Letters, 2016. Vol. 23, № 5. P. 708-712. [Електронний ресурс] : Structural Design of Convolutional Neural Networks for Steganalysis | IEEE Journals & Magazine | IEEE Xplore

18. Boroumand, M., Chen, M., Fridrich, J. Deep residual network for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, 2019. Vol. 14, № 5. P. 1181-1193. [Електронний ресурс] : Deep Residual Network for Steganalysis of Digital Images | IEEE Journals & Magazine | IEEE Xplore

19. Yedroudj, M., Comby, F., Chaumont, M. Yedrouj-Net: An efficient CNN for spatial steganalysis. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018. P. 2092-2096. [Електронний ресурс] : Yedroudj-Net: An Efficient CNN for Spatial Steganalysis | IEEE Conference Publication | IEEE Xplore

20. Zeng, J., Tan, S., Li, B., Huang, J. Large-scale JPEG image steganalysis using hybrid deep-learning framework. IEEE Transactions on Information Forensics and Security, 2018. Vol. 13, № 5. P. 1200-1214. [Електронний ресурс] : Large-Scale JPEG Image Steganalysis Using Hybrid Deep-Learning Framework | IEEE Journals & Magazine | IEEE Xplore

21. Zhang, R., Zhu, F., Liu, J., Liu, G. Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis. IEEE Transactions on Information Forensics and Security, 2020. Vol. 15. P. 1138-1150. [Електронний ресурс] : Depth-Wise Separable Convolutions and Multi-Level Pooling for an Efficient Spatial CNN-Based Steganalysis | IEEE Journals & Magazine | IEEE Xplore

22. Fridrich, J., Kodovský, J. Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, 2012. Vol. 7, № 3. P. 868-882. [Електронний ресурс] : Rich Models for Steganalysis of Digital Images | IEEE Journals & Magazine | IEEE Xplore

23. Denemark, T., Sedighi, V., Holub, V., Cogranne, R., Fridrich, J. Selection-channel-aware rich model for steganalysis of digital images. IEEE International

Workshop on Information Forensics and Security (WIFS), 2014. P. 48-53. [Електронний ресурс] : (PDF) Selection-Channel-Aware Rich Model for Steganalysis of Digital Images

24. Holub, V., Fridrich, J., Denemark, T. Universal distortion function for steganography in an arbitrary domain. EURASIP Journal on Information Security, 2014. Vol. 2014, № 1. P. 1-13. [Електронний ресурс] : (PDF) Universal Distortion Function for Steganography in an Arbitrary Domain

25. Ker, A. D., Bas, P., Böhme, R., Cogranne, R., Craver, S., Filler, T., Fridrich, J., Pevný, T. Moving steganography and steganalysis from the laboratory into the real world. Proceedings of the first ACM workshop on Information hiding and multimedia security, 2013. P. 45-58. [Електронний ресурс] : (PDF) Moving Steganography and Steganalysis from the Laboratory into the Real World

26. Giboulot, Q., Cogranne, R., Borghys, D., Bas, P. Effects and solutions of cover-source mismatch in image steganalysis. Signal Processing: Image Communication, 2020. Vol. 86. P. 115888. [Електронний ресурс] : (PDF) Effects and Solutions of Cover-Source Mismatch in Image Steganalysis

27. Li, B., Wei, W., Ferreira, A., Tan, S. REST-Net: Diverse activation modules and parallel subnets-based CNN for spatial image steganalysis. IEEE Signal Processing Letters, 2018. Vol. 25, № 5. P. 650-654. [Електронний ресурс] : ReST-Net: Diverse Activation Modules and Parallel Subnets-Based CNN for Spatial Image Steganalysis | IEEE Journals & Magazine | IEEE Xplore

28. Goodfellow, I., Bengio, Y., Courville, A. Deep Learning. MIT Press, 2016. 800 p. [Електронний ресурс] : Deep Learning - Ian Goodfellow, Yoshua Bengio, Aaron Courville - Google книги

29. Ioffe, S., Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. International Conference on Machine Learning, 2015. P. 448-456. [Електронний ресурс] : An Ensemble of Convolutional Neural Networks Using Wavelets for Image Classification

30. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 2014. Vol. 15, № 1. P. 1929-1958. [Електронний ресурс] : (PDF) Improving neural networks by preventing co-adaptation of feature detectors

31. Kingma, D. P., Ba, J. Adam: A method for stochastic optimization. International Conference on Learning Representations (ICLR), 2015. 15 p. [Електронний ресурс] : Adam: A Method for Stochastic Optimization

32. Bas, P., Filler, T., Pevný, T. «Break Our Steganographic System»: The ins and outs of organizing BOSS. Information Hiding: 13th International Conference, 2011. P. 59-70. [Електронний ресурс] : (PDF) ”Break Our Steganographic System”: The Ins and Outs of Organizing BOSS

33. Cogranne, R., Giboulot, Q., Bas, P. The ALASKA Steganalysis Challenge: A First Step Towards Steganalysis Into The Wild. Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, 2019. P. 125-137. [Електронний ресурс]: (PDF) The ALASKA Steganalysis Challenge: A First Step Towards Steganalysis

34. You, W., Zhang, H., Zhao, X. A Siamese CNN for image steganalysis. IEEE Transactions on Information Forensics and Security, 2020. Vol. 16. P. 291-306. [Електронний ресурс]: A Siamese CNN for Image Steganalysis | IEEE Journals & Magazine | IEEE Xplore

35. Mo, C., Chen, X., Wang, R., Luo, X. Deep network for steganalysis based on convolution with diverse receptive fields sizes. Digital Signal Processing, 2021. Vol. 114. P. 103066. [Електронний ресурс]: Deep Residual Network for Steganalysis of Digital Images | Request PDF

36. Apau R., Asante M., Twum F. Image steganography techniques for resisting statistical steganalysis attacks: A systematic literature review. PLoS One. 2024. Vol. 19, № 9. e0308807 . [Електронний ресурс]: https://digitnet.github.io/m4jpeg/downloads/pdf/feature-based-steganalysis-for-jpeg-images-and-its-implications-for-future-design-of-steganographic-schemes.pdf

37. Kodovský J., Fridrich J. Steganalysis of JPEG images using rich models. Proceedings of SPIE. 2012. Vol. 8303. 83030A . [Електронний ресурс]: https://dde.binghamton.edu/kodovsky/pdf/SPIE2012_Kodovsky_Steganalysis_of_JPEG_Images_Using_Rich_Models_paper.pdf

38. Wu S., Zhong S., Liu Y. Deep residual learning for image steganalysis. Multimedia Tools and Applications. 2018. Vol. 77. P. 10437–10453 . [Електронний

ресурс]: https://www.researchgate.net/publication/313785326_Deep_residual_learning_for_image_steganalysis

39. Tang W., Li B., Barni M. Improving Cost Learning for JPEG Steganography by Exploiting JPEG Domain Knowledge. IEEE Transactions on Information Forensics and Security. 2021. Vol. 16. P. 321–332 . [Електронний ресурс]: https://www.researchgate.net/publication/351477818_Improving_Cost_Learning_for_JPEG_Steganography_by_Exploiting_JPEG_Domain_Knowledge

40. Ye J., Ni J., Yi Y. Deep learning hierarchical representations for image steganalysis. IEEE Transactions on Information Forensics and Security. 2017. Vol. 12, № 11. P. 2545–2557 . [Електронний ресурс]: https://www.researchgate.net/publication/317294735_Deep_Learning_Hierarchical_Representations_for_Image_Steganalysis

41. Wu T., Ren W., Li D. JPEG steganalysis based on denoising network and attention module. International Journal of Intelligent Systems. 2021. Vol. 37. P. 120–135 . [Електронний ресурс]: https://www.researchgate.net/publication/356286298_JPEG_steganalysis_based_on_denoising_network_and_attention_module

42. Avcibas I., Memon N., Sankur B. Steganalysis using image quality metrics. IEEE Transactions on Image Processing. 2003. Vol. 12, № 2. P. 221–229 . [Електронний ресурс]: https://www.researchgate.net/publication/5613837_Steganalysis_Using_Image_Quality_Metrics

43. Lyu S., Farid H. Steganalysis using higher-order image statistics. IEEE Transactions on Information Forensics and Security. 2006. Vol. 1, № 1. P. 111–119 . [Електронний ресурс]: https://farid.berkeley.edu/downloads/publications/tifs05.pdf

44. Chen M. et al. JPEG-phase-aware convolutional neural network for steganalysis of JPEG images. Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security. 2017. P. 123–134 . [Електронний ресурс]: https://ws2.binghamton.edu/fridrich/Research/jpeg-phase-aware-Final.pdf

45. Zhang R., Zhu F., Liu J. Efficient feature learning and multi-size image steganalysis based on CNN. Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security. 2018. P. 1–12 . [Електронний ресурс]: https://www.researchgate.net/publication/342784908_Joint_multi-domain_feature_learning_for_image_steganalysis_based_on_CNN

46. Qian Y. et al. Deep learning for steganalysis via convolutional neural networks. Proceedings of SPIE. 2015. Vol. 9409. 94090J . [Електронний ресурс]: https://www.researchgate.net/publication/282265577_Deep_learning_for_steganalysis_via_convolutional_neural_networks

47. Reinel B. O. et al. GBRAS-Net: A Convolutional Neural Network Architecture for Spatial Image Steganalysis. IEEE Access. 2021. Vol. 9. P. 143–156 . [Електронний ресурс]: https://www.researchgate.net/publication/348756299_GBRAS-Net_A_Convolutional_Neural_Network_Architecture_for_Spatial_Image_Steganalysis

48. Tsang C. F., Fridrich J. Steganalyzing images of arbitrary size with CNNs. Electronic Imaging. 2018. Vol. 2018, № 7. P. 121–1 . [Електронний ресурс]: https://ws.binghamton.edu/fridrich/Research/Scale-1.12.16.pdf

49. Solanki K. et al. Robust image steganography: The generalized benford's law. Proceedings of SPIE. 2007. Vol. 6505. P. 65050S . [Електронний ресурс]: https://www.researchgate.net/publication/242075449_A_generalized_Benford's_law_for_JPEG_coefficients_and_its_applications_in_image_forensics_-_art_no_65051L

50. Fu D., Shi Y. Q., Su W. A generalized Benford's law for JPEG coefficients and its applications in image forensics. Proceedings of SPIE. 2007. Vol. 6505. 65051L . [Електронний ресурс]: https://www.researchgate.net/publication/242075449_A_generalized_Benford's_law_for_JPEG_coefficients_and_its_applications_in_image_forensics_-_art_no_65051L

51. Lee K., Westfeld A., Lee S. Category Attack for LSB Steganalysis of JPEG Images. Information Hiding. 2006. LNCS 4283. P. 35–48 . [Електронний ресурс]: https://www2.htw-dresden.de/~westfeld/publikationen/lee.westfeld.lee.ih07.pdf

52. Yu X. et al. Steganalysis with contrast features. IEEE Signal Processing Letters. 2016. Vol. 23, № 10. P. 1424–1428 . [Електронний ресурс]:

https://www.researchgate.net/publication/303779057_Spatial_Steganalysis_Using_Contrast_of_Residuals

53. Ge Y. et al. A novel technique for image Steganalysis based on separable convolution and adversarial mechanism. Electronics. 2021. Vol. 10, № 22. P. 2742 . [Електронний ресурс]: https://www.mdpi.com/2079-9292/10/22/2742

54. Kim J., Park H., Park J. I. CNN-based image steganalysis using additional data embedding. Multimedia Tools and Applications. 2020. Vol. 79. P. 1355–1372 . [Електронний ресурс]: https://www.researchgate.net/publication/336950033_CNN-based_image_steganalysis_using_additional_data_embedding

55. Luo W., Huang F., Huang J. Edge adaptive image steganography based on LSB matching revisited. IEEE Transactions on Information Forensics and Security. 2010. Vol. 5, № 2. P. 201–214 . [Електронний ресурс]: https://www.researchgate.net/publication/220177204_Edge_Adaptive_Image_Steganography_Based_on_LSB_Matching_Revisited

56. Sachnev V. et al. Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding. Proceedings of the 11th ACM workshop on Multimedia and security. 2009. P. 131–140 . [Електронний ресурс]: https://www.researchgate.net/publication/234830956_Less_detectable_JPEG_steganography_method_based_on_heuristic_optimization_and_BCH_syndrome_coding

57. Westfeld A. Generic methodology for higher order steganalysis. Proceedings of the 10th ACM workshop on Multimedia and security. 2008. P. 161–168 . [Електронний ресурс]: https://www.informatik.htw-dresden.de/~westfeld/publikationen/westfeld-ih08.pdf

58. Sallee P. Model-Based Steganography. International Workshop on Digital Watermarking. 2003. LNCS 2939. P. 154–167 . [Електронний ресурс]: https://digitnet.github.io/m4jpeg/downloads/pdf/model-based-steganography.pdf

59. Jolion J. M. Images and Benford's law. Journal of Mathematical Imaging and Vision. 2001. Vol. 14, № 1. P. 73–81 . [Електронний ресурс]: https://www.researchgate.net/publication/220146443_Images_and_Benford's_Law

60. Tabares-Soto R. et al. Strategy to improve the accuracy of convolutional neural network for image steganalysis. PeerJ Computer Science. 2021. Vol. 7. e616 . [Електронний ресурс]: https://peerj.com/articles/cs-451/

61. Mstafa R., Elleithy K. M., Abdelfattah E. A robust and secure video steganography method in DWT-DCT domains. IEEE Access. 2017. Vol. 5. P. 5354–5365 . [Електронний ресурс]: https://www.researchgate.net/publication/316021470_A_Robust_and_Secure_Video_Steganography_Method_in_DWT-DCT_Domains_Based_on_Multiple_Object_Tracking_and_ECC

62. Kumar A., Rani R., Singh S. A survey of recent advances in image steganography. Security and Privacy. 2023. Vol. 6, № 3. e281 . [Електронний ресурс]: https://www.researchgate.net/publication/365314243_A_survey_of_recent_advances_in_image_steganography

63. Liu Q. Steganalysis of DCT-embedding based adaptive steganography and YASS. Proceedings of the 13th ACM multimedia workshop on Multimedia and security. 2011. P. 77–86 . [Електронний ресурс]: https://www.researchgate.net/publication/254006296_Steganalysis_of_DCT-embedding_based_adaptive_steganography_and_YASS

64. Wang P., Wei Z., Xiao L. Pure spatial rich model features for digital image steganalysis. Multimedia Tools and Applications. 2015. Vol. 75. P. 1–20 . [Електронний ресурс]: https://www.researchgate.net/publication/276090664_Pure_spatial_rich_model_features_for_digital_image_steganalysis

65. Tang W. et al. Automatic steganographic distortion learning using a generative adversarial network. IEEE Signal Processing Letters. 2017. Vol. 24, № 10. P. 1547–1551 . [Електронний ресурс]: https://www.researchgate.net/publication/319325811_Automatic_Steganographic_Distortion_Learning_Using_a_Generative_Adversarial_Network

66. Yang J. et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015. Vol.

37, № 9. P. 1904–1916 . [Електронний ресурс]: https://www.researchgate.net/publication/263237865_Spatial_Pyramid_Pooling_in_Deep_Convolutional_Networks_for_Visual_Recognition

67. He K. et al. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. P. 770–778 . [Електронний ресурс]: https://ieeexplore.ieee.org/document/7780459

68. Szegedy C. et al. Rethinking the inception architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. P. 2818–2826 . [Електронний ресурс]: https://ieeexplore.ieee.org/document/7780677

69. Breiman L. Random forests. Machine learning. 2001. Vol. 45, № 1. P. 5–32 . [Електронний ресурс]: https://www.researchgate.net/publication/275342330_Random_Forests

70. Cortes C., Vapnik V. Support-vector networks. Machine learning. 1995. Vol. 20, № 3. P. 273–297 . [Електронний ресурс]: https://scispace.com/pdf/support-vector-networks-2jd9a1kl0z.pdf

Додаток А Технічне завдання

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ
Голова секції "Управління інформаційною безпекою" кафедри МБІС
д.т.н., професор
Юрій ЯРЕМЧУК
"24" березня 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:

Вдосконалення методу стеганографії зображень у просторовій та частотній областях на основі RS- та DCT-аналізу з використанням згорткової нейронної мережі (CNN)

08-72.МКР.007.00.158.ТЗ

Керівник магістерської кваліфікаційної роботи
к.т.н., доцент
Карпанець В. В.

Вінниця – 2025 р.

1. Найменування та область застосування

Вдосконалення методу стегоаналізу зображень у просторовій та частотній областях на основі RS- та DCT-аналізу з використанням згорткової нейронної мережі (CNN). Область застосування: кібербезпека, стегоаналіз цифрових зображень, системи моніторингу інформаційної безпеки.

2. Підстава для проведення робіт

Робота виконується на основі наказу ректора ВНТУ № 313 від 24. 09. 2025 р.

3. Мета та призначення МКР

3.1 Мета: розробка та дослідження гібридного методу стегоаналізу, що поєднує класичні статистики RS-аналізу та DCT-аналізу з нейромережевим класифікатором на основі 1D-CNN для універсального виявлення стеганографії в зображеннях різних форматів (BMP, JPEG) з підвищеною точністю порівняно з базовими методами.

3.2 Призначення: підвищення точності та надійності виявлення прихованої інформації в цифрових зображеннях, забезпечення балансу між ефективністю виявлення та обчислювальною складністю для практичного впровадження в системах моніторингу інформаційної безпеки.

4. Джерела розробки

4.1. Fridrich, J. Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press, 2009.

4.2. Westfeld, A., Pfitzmann, A. Attacks on steganographic systems. Information Hiding: Third International Workshop, 2000.

4.3. Boroumand, M., Chen, M., Fridrich, J. Deep residual network for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, 2019.

4.4. Yedroudj, M., Comby, F., Chaumont, M. Yedrouj-Net: An efficient CNN for spatial steganalysis. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018.

4.5. Методичні вказівки до виконання магістерської кваліфікаційної роботи для студентів спеціальності 125 «Кібербезпека та захист інформації».

5. Вимоги до виконання МКР

Для досягнення мети необхідно вирішити наступні завдання:

- провести аналіз сучасних методів стеганографії та стегоаналізу, класифікувати їх за доменами вбудовування та адаптивністю;

- дослідити математичні основи RS-аналізу для просторової області, визначити його сильні та слабкі сторони при виявленні LSB та адаптивних методів;

- вивчити DCT-аналіз для частотної області, включаючи гістограмний аналіз та χ^2-атаку, оцінити обмеження для JPEG-зображень;

- проаналізувати сучасні CNN-методи стегоаналізу SRNet, Yedroudj-Net, CSM, виявити їх переваги та недоліки;

- розробити концептуальну модель гібридної системи, що інтегрує RS та DCT ознаки з 1D-CNN класифікатором;

- реалізувати модулі виділення ознак RS та DCT статистики з векторизацією та зменшенням розмірності;

- спроектувати та навчити 1D-CNN класифікатор на датасеті BOSSBase з різними стеганографічними алгоритмами;

- розробити логіку інтеграції ознак з використанням зваженої fusion та правил прийняття рішень;

- провести експериментальну оцінку методу за метриками AUC-ROC, F1-score та False Positive Rate, порівняти з базовими підходами;

- провести економічний розрахунок запропонованих рішень.

6. Вимоги до розроблення документації

Оформлення МКР повинно відповідати вимогам державних стандартів України, зокрема ДСТУ 3008:2015, ДСТУ 8302:2015.

7. Техніко-економічні показники

7.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

7.2 Обрані рішення та можливість їх практичної реалізації повинні бути орієнтовані на широкий загал.

8. Стадії та етапи розробки

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Початок	Закінчення
1	Визначення напрямку магістерської роботи, формулювання теми	01.09.2025	15.09.2025
2	Аналіз предметної області обраної теми	15.09.2025	20.09.2025
3	Апробація отриманих результатів	20.09.2025	30.09.2025
4	Розробка алгоритму роботи	01.10.2025	10.10.2025
5	Написання магістерської роботи на основі розробленої теми	01.10.2025	20.11.2025
6	Розробка економічної частини	15.11.2025	20.11.2025
7	Попередній захист магістерської кваліфікаційної роботи	24.11.2025	25.11.2025
8	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	01.12.2025	07.12.2025
9	Захист магістерської кваліфікаційної роботи	08.12.2025	11.12.2025

10. Порядок контролю та прийому

10.1 До приймання магістерської кваліфікаційної роботи надається:

– ПЗ до магістерської кваліфікаційної роботи;

– програмний додаток;

– презентація;

– відзив керівника роботи;

– відзив опонента.

Технічне завдання до виконання прийняв _____ Горохов А.В.

Додаток Б Лістинги програмного коду

В.1. Модуль RS- та DCT-аналізу зображень features.py

```
python
import numpy as np
from PIL import Image
from scipy.fftpack import dct
from scipy.stats import skew, kurtosis
import io

RS-АНАЛІЗ (15 ОЗНАК) - КЛАСИЧНА РЕАЛІЗАЦІЯ

def compute_rs_features(image_file):

    Класичний RS-аналіз Фрідріха для виявлення LSB-стеганографії.
    Повертає 15 ознак згідно з розділом 2.2 роботи.
     Завантаження зображення

    if isinstance(image_file, str):
        img = Image.open(image_file)
    else:
        if hasattr(image_file, 'seek'):
            image_file.seek(0)
        img = Image.open(io.BytesIO(image_file.read()) if hasattr(image_file, 'read') else image_file)

    if img.mode != 'L':
        img = img.convert('L')

     Працюємо в uint8 (0-255), НЕ нормалізуємо
    img_array = np.array(img, dtype=np.uint8)

     Формуємо групи по 4 послідовних пікселі
    flat = img_array.flatten()
    n = 4
    m = (len(flat) // n) * n
    flat = flat[:m]
    groups = flat.reshape(-1, n).astype(np.int16)   (G, 4)

     Маски
    M1  = np.array([1, 0, 1, 0], dtype=np.int16)
    M_1 = np.array([-1, 0, -1, 0], dtype=np.int16)

    def discr(g):
    Дискримінант f(G) = Σ|x_{i+1} - x_i|
        return np.sum(np.abs(np.diff(g, axis=1)), axis=1)

    def classify(groups, mask):
    Класифікація груп відносно маски
        g_masked = groups + mask
        g_masked = np.clip(g_masked, 0, 255).astype(np.int16)

        f0 = discr(groups)
        f1 = discr(g_masked)
```

```
    R = np.sum(f1 > f0)
    S = np.sum(f1 < f0)
    U = np.sum(f1 == f0)
    total = len(groups)
    return R/total, S/total, U/total

 1-6: Базові RS-статистики
r_m1, s_m1, u_m1     = classify(groups, M1)
r_m_1, s_m_1, u_m_1 = classify(groups, M_1)

 Інверсія LSB через бітовий XOR
flipped = (flat ^ 1).reshape(-1, n).astype(np.int16)
r_m1_f, s_m1_f, _ = classify(flipped, M1)
r_m_1_f, s_m_1_f, _ = classify(flipped, M_1)

 7-9: Дискримінанти та payload (формула Фрідріха з корекцією)
d_m1 = r_m1 - s_m1
d_m_1 = r_m_1 - s_m_1
d0 = d_m1 + d_m_1

d_m1_f = r_m1_f - s_m1_f
d_m_1_f = r_m_1_f - s_m_1_f
d1 = d_m1_f + d_m_1_f

 Класична формула Фрідріха: p = d0 / (d0 - d1/2)
eps = 1e-8
denominator = d0 - d1 / 2.0

if abs(denominator) > eps:
    p_est = d0 / denominator
else:
    p_est = 0.0

 Обмеження [0, 1] та фільтр шуму
p_est = float(np.clip(p_est, 0.0, 1.0))
if p_est < 0.01:
    p_est = 0.0

 10-11: Асиметрії
asymmetry_r = abs(r_m1 - r_m_1)
asymmetry_s = abs(s_m1 - s_m_1)

 12: Узгодженість
consistency = 1.0 - min(abs(d0 - d1) / 2.0, 1.0)

 13-14: Варіативність
variance_r = float(np.var([r_m1, r_m_1]))
variance_s = float(np.var([s_m1, s_m_1]))

 15: Edge ratio
edge_ratio = _compute_edge_ratio(img_array.astype(np.float32) / 255.0)
```

```
    features = np.array([
        r_m1, s_m1, u_m1,
        r_m_1, s_m_1, u_m_1,
        d0, d1, p_est,
        asymmetry_r, asymmetry_s,
        consistency,
        variance_r, variance_s,
        edge_ratio
    ], dtype=np.float32)

    return features


def _compute_edge_ratio(img_array):
    Обчислення частки пікселів на межах об'єктів
    from scipy import ndimage
    sx = ndimage.sobel(img_array, axis=0)
    sy = ndimage.sobel(img_array, axis=1)
    edges = np.sqrt(sx**2 + sy**2)

    threshold = np.mean(edges) + np.std(edges)
    edge_pixels = np.sum(edges > threshold)

    return float(edge_pixels / (img_array.size + 1e-8))

DCT-АНАЛІЗ (25 ОЗНАК) - ПОЛІПШЕНА РЕАЛІЗАЦІЯ
def compute_dct_features(image_file):

    DCT-аналіз для виявлення стеганографії у частотній області.
    Повертає 25 ознак згідно з розділом 2.2 роботи.

     Завантаження
    if isinstance(image_file, str):
        img = Image.open(image_file)
    else:
        if hasattr(image_file, 'seek'):
            image_file.seek(0)
        img = Image.open(io.BytesIO(image_file.read()) if hasattr(image_file, 'read') else image_file)

    if img.mode != 'L':
        img = img.convert('L')

     limit визначено на початку
    limit = 50

    img_array = np.array(img, dtype=np.float32) - 128.0   центруємо навколо 0

     Розбиття на блоки 8×8
    h, w = img_array.shape
    h2 = (h // 8) * 8
    w2 = (w // 8) * 8
    img_array = img_array[:h2, :w2]
```

```
blocks_h = h2 // 8
blocks_w = w2 // 8

dct_coeffs = []

for i in range(blocks_h):
    for j in range(blocks_w):
        block = img_array[i*8:(i+1)*8, j*8:(j+1)*8]
         2D DCT
        dct_block = dct(dct(block, axis=0, norm='ortho'), axis=1, norm='ortho')
        dct_coeffs.append(dct_block.flatten())

if len(dct_coeffs) == 0:
    return np.zeros(25, dtype=np.float32)

dct_matrix = np.vstack(dct_coeffs)   (N_blocks, 64)

 AC-коефіцієнти (без DC)
acs = dct_matrix[:, 1:].flatten()
acs_int = np.round(acs).astype(int)

acs_clip = np.clip(acs_int, -limit, limit)

 Гістограма
values, counts = np.unique(acs_clip, return_counts=True)
total = counts.sum()
probs = counts / total

 Ознаки 1-10
H0 = float(np.sum(acs_int == 0) / len(acs_int))
H1 = float(np.sum(np.abs(acs_int) == 1) / len(acs_int))
H2 = float(np.sum(np.abs(acs_int) == 2) / len(acs_int))
H0_over_H1 = H0 / (H1 + 1e-8)

chi2_val = 0.0
dof = 0
for k in range(-limit, limit, 2):
    c0 = counts[values == k]
    c1 = counts[values == k+1]
    if len(c0) == 0 and len(c1) == 0:
        continue
    c0 = c0[0] if len(c0) else 0
    c1 = c1[0] if len(c1) else 0
    E = (c0 + c1) / 2.0
    if E == 0:
        continue
    chi2_val += (c0 - E)**2 / E + (c1 - E)**2 / E
    dof += 1

normalized_chi2 = chi2_val / (dof + 1e-8)
p_value = float(np.clip(1.0 - normalized_chi2 / 100.0, 0.0, 1.0))

mean_ac = float(np.mean(acs))
```

```
std_ac  = float(np.std(acs) + 1e-8)
skewness_ac = float(skew(acs))
kurt_ac = float(kurtosis(acs))

pos = np.sum(acs > 0)
neg = np.sum(acs < 0)
sign_ratio = float(pos / (neg + 1e-8))

 Ентропія
hist_ac_temp = probs[probs > 0]
entropy_temp = -float(np.sum(hist_ac_temp * np.log2(hist_ac_temp + 1e-12)))

std_acs = float(np.std(acs))

chi2_component = min(normalized_chi2 * 2.0, 50.0)
std_component = min(std_acs, 50.0)
entropy_component = max(0, (8.0 - entropy_temp) * 10.0)
h0_component = max(0, (H0 - 0.05) * 40.0)

anomaly_score = float(
   0.35 * chi2_component +
   0.25 * (50.0 - std_component) +
   0.25 * entropy_component +
   0.15 * h0_component
)
anomaly_score = np.clip(anomaly_score, 0.0, 100.0)

features = [
   anomaly_score,
   p_value,
   H0, H1, H2, H0_over_H1,
   skewness_ac, kurt_ac, mean_ac, std_ac
]

 Ознаки 11-14
from scipy.ndimage import gaussian_filter
img_blur = gaussian_filter(img_array, sigma=0.5)

blocks_blur = []
for i in range(blocks_h):
   for j in range(blocks_w):
      block = img_blur[i*8:(i+1)*8, j*8:(j+1)*8]
      dct_block = dct(dct(block, axis=0, norm='ortho'), axis=1, norm='ortho')
      blocks_blur.append(dct_block.flatten())

dct_blur = np.vstack(blocks_blur)[:, 1:].flatten()
acs_blur_int = np.clip(np.round(dct_blur).astype(int), -limit, limit)

vals_b, cnts_b = np.unique(acs_blur_int, return_counts=True)
probs_b = cnts_b / cnts_b.sum()

all_vals = np.arange(-limit, limit+1)
def make_prob_vec(vals, probs):
```

```
    p = np.zeros(len(all_vals), dtype=np.float64) + 1e-8
    for v, prob in zip(vals, probs):
        idx = int(v + limit)
        if 0 <= idx < len(p):
            p[idx] = prob
    p /= p.sum()
    return p

P = make_prob_vec(values, probs)
Q = make_prob_vec(vals_b, probs_b)
KL_div = float(np.sum(P * np.log((P + 1e-12) / (Q + 1e-12))))

dc_coeffs = dct_matrix[:, 0]
block_discont = float(np.mean(np.abs(np.diff(dc_coeffs))))
periodicity = float(np.std(dc_coeffs) / (np.mean(np.abs(dc_coeffs)) + 1e-8))

features.extend([sign_ratio, KL_div, block_discont, periodicity])

 Ознаки 15-17: енергетичний розподіл
low_mask  = np.zeros(64, dtype=bool); low_mask[:9] = True
high_mask = np.zeros(64, dtype=bool); high_mask[36:] = True
mid_mask  = ~(low_mask | high_mask)

energy = dct_matrix**2
total_energy = np.sum(energy) + 1e-8
low_energy  = float(np.sum(energy[:, low_mask])  / total_energy)
mid_energy  = float(np.sum(energy[:, mid_mask])  / total_energy)
high_energy = float(np.sum(energy[:, high_mask]) / total_energy)

features.extend([low_energy, mid_energy, high_energy])

 Ознаки 18-25
hist_ac = probs[probs > 0]
entropy = -float(np.sum(hist_ac * np.log2(hist_ac + 1e-12)))

max_coeff = float(np.max(acs))
min_coeff = float(np.min(acs))
range_coeff = max_coeff - min_coeff
zero_fraction = H0
nonzero = acs[acs != 0]
mean_nonzero_abs = float(np.mean(np.abs(nonzero))) if nonzero.size > 0 else 0.0

diff_sum = 0.0
count = 0
for blk in dct_matrix:
    b = blk.reshape(8,8)
    diff_sum += np.sum(np.abs(np.diff(b, axis=0))) + np.sum(np.abs(np.diff(b, axis=1)))
    count += (8-1)*8*2
clustering_index = float(diff_sum / (count + 1e-8))

sym_diff = 0.0
P_full = make_prob_vec(values, probs)
for k in range(1, limit+1):
```

```
        pk  = P_full[limit + k]
        pnk = P_full[limit - k]
        sym_diff += abs(pk - pnk)
    symmetry_index = 1.0 - float(sym_diff / 2.0)

    features.extend([
        entropy, max_coeff, min_coeff, range_coeff,
        zero_fraction, mean_nonzero_abs,
        clustering_index, symmetry_index
    ])

    return np.array(features[:25], dtype=np.float32)
```

B.2. Модуль гібридного CNN-класифікатора cnn_model.py

```
python
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torch.utils.data import DataLoader, TensorDataset
import numpy as np
from PIL import Image
import io
import joblib

 Перевірка доступності GPU
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Using device: {device}")

CNN_MODEL_PATH = "cnn_model.pth"
CNN_SCALER_PATH = "cnn_scaler.joblib"


class SRMConv2d(nn.Module):

    SRM (Spatial Rich Models) шар для виявлення адаптивної стеганографії.
    Використовує 30 фіксованих фільтрів для виділення шуму/residuals.

    def __init__(self, stride=1, padding=0):
        super(SRMConv2d, self).__init__()
        self.in_channels = 1
        self.out_channels = 30
        self.kernel_size = 5
        self.stride = stride
        self.padding = padding

        filter1 = [[0, 0, 0, 0, 0],
                   [0, -1, 2, -1, 0],
                   [0, 2, -4, 2, 0],
                   [0, -1, 2, -1, 0],
                   [0, 0, 0, 0, 0]]
```

```
        filter2 = [[-1, 2, -2, 2, -1],
                   [2, -6, 8, -6, 2],
                   [-2, 8, -12, 8, -2],
                   [2, -6, 8, -6, 2],
                   [-1, 2, -2, 2, -1]]

        filter3 = [[0, 0, 0, 0, 0],
                   [0, 0, 0, 0, 0],
                   [0, 1, -2, 1, 0],
                   [0, 0, 0, 0, 0],
                   [0, 0, 0, 0, 0]]

        q_filters = []
        for _ in range(10): q_filters.append(filter1)
        for _ in range(10): q_filters.append(filter2)
        for _ in range(10): q_filters.append(filter3)

        q_filters = np.array(q_filters, dtype=np.float32) / 4.0

        self.weight = nn.Parameter(
            torch.from_numpy(q_filters).unsqueeze(1),
            requires_grad=False
        )

    def forward(self, x):
        return F.conv2d(x, self.weight, stride=self.stride, padding=self.padding)


class StegoCNN(nn.Module):

    Гібридний CNN для детектування стеганографії з SRM pre-processing.
    Вхід:
      - image: (batch, 1, 256, 256)
      - features: (batch, 40) — 15 RS + 25 DCT

    def __init__(self, input_channels=1, feature_dim=40):
        super(StegoCNN, self).__init__()

        self.srm_layer = SRMConv2d(padding=2)

         CNN гілка
        self.conv_block1 = nn.Sequential(
            nn.Conv2d(30, 32, kernel_size=3, padding=1),
            nn.BatchNorm2d(32),
            nn.ReLU(),
            nn.MaxPool2d(2, 2)
        )

        self.conv_block2 = nn.Sequential(
            nn.Conv2d(32, 64, kernel_size=3, padding=1),
            nn.BatchNorm2d(64),
            nn.ReLU(),
            nn.MaxPool2d(2, 2)
```

```
        )

        self.conv_block3 = nn.Sequential(
            nn.Conv2d(64, 128, kernel_size=3, padding=1),
            nn.BatchNorm2d(128),
            nn.ReLU(),
            nn.MaxPool2d(2, 2)
        )

        self.adaptive_pool = nn.AdaptiveAvgPool2d((4, 4))

        # FC для ознак
        self.feature_fc = nn.Sequential(
            nn.Linear(feature_dim, 128),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(128, 64),
            nn.ReLU()
        )

        # Fusion: 128*4*4 + 64 = 2112
        self.fusion_fc = nn.Sequential(
            nn.Linear(128 * 4 * 4 + 64, 256),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(256, 128),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(128, 1),
            nn.Sigmoid()
        )

    def forward(self, x_img, x_feat):
        x_noise = self.srm_layer(x_img)

        x = self.conv_block1(x_noise)
        x = self.conv_block2(x)
        x = self.conv_block3(x)
        x = self.adaptive_pool(x)
        x = x.view(x.size(0), -1)

        feat = self.feature_fc(x_feat)
        combined = torch.cat([x, feat], dim=1)
        output = self.fusion_fc(combined)
        return output


def preprocess_image(image_file, target_size=256):

    Підготовка зображення для CNN:
    - grayscale
    - resize
    - нормалізація до [0,1]
```

```
    try:
        img = None

        if hasattr(image_file, 'file'):
            image_file = image_file.file

        if hasattr(image_file, 'seek'):
            image_file.seek(0)

        if isinstance(image_file, str):
            img = Image.open(image_file)
        elif hasattr(image_file, 'read'):
            file_content = image_file.read()
            img = Image.open(io.BytesIO(file_content))
        elif isinstance(image_file, bytes):
            img = Image.open(io.BytesIO(image_file))
        else:
            img = Image.open(image_file)

        if img is None:
            raise ValueError("Could not open image")

        if img.mode != 'L':
            img = img.convert('L')

        img = img.resize((target_size, target_size), Image.Resampling.LANCZOS)
        img_array = np.array(img, dtype=np.float32) / 255.0

        return img_array
    except Exception as e:
        print(f"Error preprocessing image: {e}")
        import traceback
        traceback.print_exc()
        return None


def create_cnn_model():
    model = StegoCNN(input_channels=1, feature_dim=40).to(device)
    return model


def save_cnn_model(model, history=None):
    torch.save({
        'model_state_dict': model.state_dict(),
        'history': history
    }, CNN_MODEL_PATH)
    print(f"CNN model saved to {CNN_MODEL_PATH}")


def load_cnn_model():
    checkpoint = torch.load(CNN_MODEL_PATH, map_location=device)
    model = StegoCNN(input_channels=1, feature_dim=40).to(device)
```

```
    model.load_state_dict(checkpoint['model_state_dict'])
    model.eval()
    return model
```

(…далі в роботі можна продовжити лістингом функцій train_cnn та analyze_with_cnn, якщо потрібно повністю.)

В.3. Головний модуль REST-сервера main.py

```
python
from fastapi import FastAPI, UploadFile, File, Form
from fastapi.middleware.cors import CORSMiddleware
from fastapi.staticfiles import StaticFiles
from fastapi.responses import FileResponse
from typing import List, Dict, Any
from contextlib import asynccontextmanager
import numpy as np
import time
import os
import json
from pathlib import Path

from features import compute_rs_features, compute_dct_features
from model_store import create_model, save_model, load_model, MODEL_PATH
from cnn_model import train_cnn, analyze_with_cnn, CNN_MODEL_PATH

THRESHOLDS_FILE = Path(__file__).parent / "thresholds.json"

RS_THRESHOLD = 5.0
DCT_THRESHOLD = 30.0
COMBINED_THRESHOLD = 0.25

def load_thresholds():
Завантажити пороги з файлу або використати дефолтні з літератури
    global RS_THRESHOLD, DCT_THRESHOLD, COMBINED_THRESHOLD

    if THRESHOLDS_FILE.exists():
        try:
            with open(THRESHOLDS_FILE, "r", encoding="utf-8") as f:
                t = json.load(f)
            RS_THRESHOLD = t.get("rs_threshold", 5.0)
            DCT_THRESHOLD = t.get("dct_threshold", 30.0)
            COMBINED_THRESHOLD = t.get("combined_threshold", 0.25)
            print(f"✓ Завантажено адаптивні пороги: RS={RS_THRESHOLD:.2f}%, DCT={DCT_THRESHOLD:.2f}")
            return
        except Exception as e:
            print(f" Помилка завантаження порогів: {e}. Використовую дефолтні.")

    print(" Використовую дефолтні пороги з літератури")
    RS_THRESHOLD = 5.0
    DCT_THRESHOLD = 30.0
    COMBINED_THRESHOLD = 0.25
```

```
@asynccontextmanager
async def lifespan(app: FastAPI):
    import sys
    if sys.platform == 'win32':
        import io
        sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding='utf-8', errors='replace')
        sys.stderr = io.TextIOWrapper(sys.stderr.buffer, encoding='utf-8', errors='replace')

    print("=" * 60, flush=True)
    print("Steganography Detection API zapushcheno", flush=True)

    load_thresholds()

    model_status = "Znaideno" if os.path.exists(CNN_MODEL_PATH) else "Ne natrenovano"
    print(f"CNN Model: {model_status}", flush=True)
    print("=" * 60, flush=True)

    yield

    print("\nZavershennia roboty servera...", flush=True)


app = FastAPI(title="Steganography Detection API", version="1.0.0", lifespan=lifespan)

app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

FRONTEND_BUILD_PATH = Path(__file__).parent.parent / "build"


*(...далі — лістинги обробників /api/train, /api/analyze, /api/metrics/*, /api/calibrate та блоку if __name__ ==
"__main__":.)*
```

В.4. Модуль калібрування порогів calibrate_thresholds.py

```
python
Калібрування порогів для RS, DCT та комбінованого детектора
Базується на ROC-кривих та оптимізації FPR/TPR
import numpy as np
import glob
import os
import json
from features import compute_rs_features, compute_dct_features
from sklearn.metrics import roc_curve, auc, precision_recall_curve
import matplotlib.pyplot as plt
import argparse
```

```
def compute_scores_for_dataset(image_paths, verbose=True):

    Обчислює RS та DCT scores для датасету зображень

    rs_scores = []
    dct_scores = []

    total = len(image_paths)
    for i, img_path in enumerate(image_paths):
        try:
            rs = compute_rs_features(img_path)
            rs_score = float(rs[8]) * 100
            rs_scores.append(rs_score)

            dct = compute_dct_features(img_path)
            dct_score = float(dct[0])
            dct_scores.append(dct_score)

            if verbose and (i + 1) % 100 == 0:
                print(f"  Оброблено {i+1}/{total} зображень...")
        except Exception as e:
            print(f"  Помилка при обробці {img_path}: {e}")
            continue

    return rs_scores, dct_scores


*(...далі — функції plot_score_distributions, plot_roc_curves, find_optimal_threshold, calibrate,
calibrate_on_images, main.)*
```

B.5. Модуль тестування RS/DCT test_features.py

```
python

Тестування RS та DCT методів
Перевірка, що різні зображення дають різні RS та DCT scores

import numpy as np
from PIL import Image
from features import compute_rs_features, compute_dct_features

def create_random_image(size=256, seed=None):
    if seed is not None:
        np.random.seed(seed)
    img_array = np.random.randint(0, 256, (size, size), dtype=np.uint8)
    return Image.fromarray(img_array, mode='L')

def create_smooth_image(size=256):
    img_array = np.zeros((size, size), dtype=np.uint8)
    for i in range(size):
        img_array[i, :] = int((i / size) * 255)
    return Image.fromarray(img_array, mode='L')
```

```
def create_textured_image(size=256):
    img_array = np.zeros((size, size), dtype=np.uint8)
    for i in range(size):
        for j in range(size):
            img_array[i, j] = 255 if (i // 32 + j // 32) % 2 == 0 else 0
    return Image.fromarray(img_array, mode='L')

def simulate_lsb_stego(img, payload_ratio=0.3):
    img_array = np.array(img, dtype=np.uint8)
    flat = img_array.flatten()
    num_pixels = int(len(flat) * payload_ratio)
    indices = np.random.choice(len(flat), num_pixels, replace=False)
    flat[indices] ^= 1
    return Image.fromarray(flat.reshape(img_array.shape), mode='L')

def test_multiple_images():

    Тест на 5 різних зображеннях

    print("=" * 70)
    print("ТЕСТУВАННЯ RS ТА DCT МЕТОДІВ НА РІЗНИХ ЗОБРАЖЕННЯХ")
    print("=" * 70)
    print()

    test_cases = [
        ("Random Noise 1", create_random_image(seed=42)),
        ("Random Noise 2", create_random_image(seed=123)),
        ("Smooth Gradient", create_smooth_image()),
        ("Checkerboard", create_textured_image()),
        ("LSB Stego (30%)", simulate_lsb_stego(create_random_image(seed=42), 0.3)),
    ]

    results = []

    for name, img in test_cases:
        filename = f"test_{name.replace(' ', '_').replace('', '').lower()}.png"
        img.save(filename)

        rs = compute_rs_features(filename)
        dct = compute_dct_features(filename)

        rs_payload = rs[8] * 100
        dct_anomaly = dct[0]

        results.append({
            "name": name,
            "rs_score": rs_payload,
            "dct_score": dct_anomaly
        })

        print(f"📷 {name:20s} | RS Score: {rs_payload:6.2f}% | DCT Score: {dct_anomaly:6.2f}")
```

В.6. Модуль моделі логістичної регресії model_store.py

```
python
import joblib
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

MODEL_PATH = "saved_model.joblib"

def create_model():
Створити нову модель з масштабувальником
   scaler = StandardScaler()
   clf = LogisticRegression(max_iter=1000, random_state=42)
   return scaler, clf

def save_model(scaler, clf):
Зберегти scaler та модель у файл
   joblib.dump({"scaler": scaler, "model": clf}, MODEL_PATH)

def load_model():
Завантажити scaler та модель з файлу
   data = joblib.load(MODEL_PATH)
   return data["scaler"], data["model"]
```

В.7. Сценарій запуску бекенд-сервера run_server.py

```
python

Wrapper для запуску backend сервера з автоматичним перезапуском

import subprocess
import time
import sys

def run_server():
   max_restarts = 5
   restart_count = 0

   while restart_count < max_restarts:
      try:
         print(f"\n{'='*60}")
         print(f"Запуск бекенд сервера (спроба {restart_count + 1}/{max_restarts})")
         print(f"{'='*60}\n")

         process = subprocess.run(
            [sys.executable, "main.py"],
            cwd=".",
            check=False
         )

         if process.returncode == 0 or process.returncode == -2:
            print("\nСервер зупинено користувачем")
            break
```

```
            print(f"\nПомилка: сервер завершився з кодом {process.returncode}")
            restart_count += 1

            if restart_count < max_restarts:
                print(f"Перезапуск через 3 секунди...")
                time.sleep(3)

        except KeyboardInterrupt:
            print("\n\nЗупинка сервера...")
            break
        except Exception as e:
            print(f"\nНеочікувана помилка: {e}")
            restart_count += 1
            if restart_count < max_restarts:
                time.sleep(3)

    if restart_count >= max_restarts:
        print("\n Досягнуто максимальну кількість перезапусків")
        print("Перевірте логи та виправте помилки у коді")

if __name__ == "__main__":
    run_server()
```

Додаток В Ілюстраційний матеріал

Магістерська кваліфікаційна робота на тему: Вдосконалення методу стегоаналізу зображень у просторовій та частотній областях на основі RS- та DCT-аналізу з використанням згорткової нейронної мережі (CNN)

Виконав: ст. гр. 2КІТС-24М Горохов А.В.
Керівник мкр; к.т.н., доц., зав. каф. МБІС Карпінець В. В.

Рисунок В.1 - Слайд 1: Титульний слайд

Актуальність та Проблематика

Зростання Кіберзагроз

Понад 40% витоків корпоративної інформації у 2023 році здійснювалися через стеганографію.

Основні Загрози

- Передача шкідливого ПЗ
- Витік конфіденційних даних
- Координація злочинних груп

Проблеми Існуючих Рішень

- Традиційні методи неефективні проти адаптивної стеганографії (точність < 70%).
- Сучасні CNN важкі, повільні та залежать від джерела зображення (Cover-Source Mismatch).

Актуальність теми зумовлена зростанням кіберзагроз. Статистика свідчить, що більше 40% витоків даних реалізуються саме через стеганографічні канали. Головна проблема полягає в тому, що зловмисники перейшли від простих методів до адаптивних алгоритмів, які підлаштовуються під текстуру зображення.

Рисунок В.2 - Слайд 2: Актуальність теми та проблематика

Рисунок В.3 - Слайд 3: Мета, об’єкт та предмет дослідження

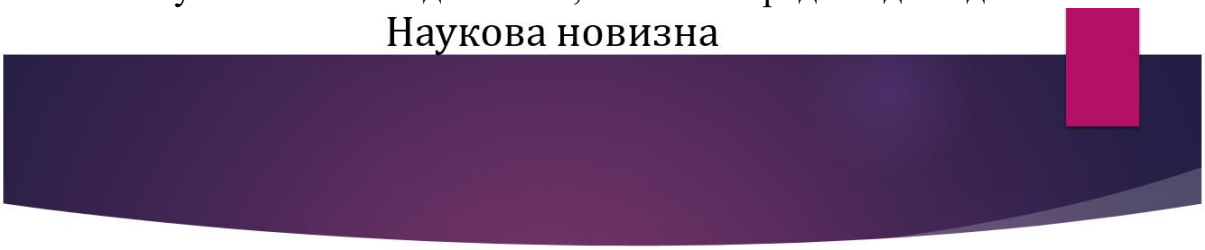
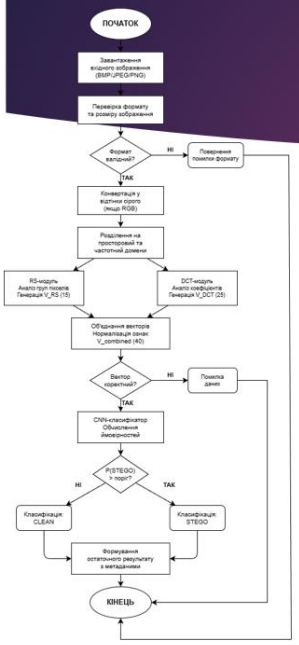


Рисунок В.4 - Слайд 4: Наукова новизна

Загальна концепція системи



1 - Зображення
Вхідний потік даних.

2 - Паралельна обробка
Модуль RS: 15 ознак (просторова обл.). Модуль DCT: 25 ознак (частотна обл.).

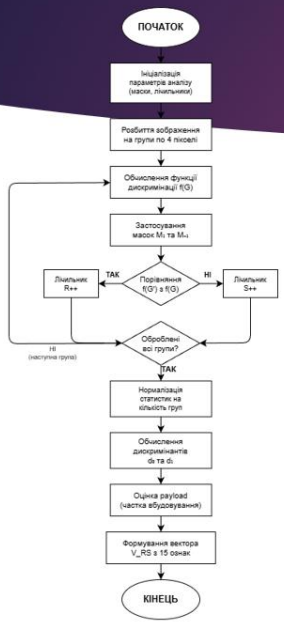
4 - Класифікація
Рішення: Clean / Stego.

3 - Інтеграція
Вектор ознак: 40 елементів.

Вона працює за принципом паралельно-послідовної обробки. На вхід подається зображення, яке проходить препроцесинг. Далі потік розділяється на два незалежних модулі: RS-аналіз для просторової області та DCT-аналіз для частотної. Це критично важливо, адже незалежність модулів запобігає поширенню помилок. Результатом роботи модулів є вектори ознак (15 від RS та 25 від DCT). Вони нормалізуються, об'єднуються у вектор довжиною 40 елементів і подаються на вхід нейронної мережі, яка приймає остаточне рішення.

Рисунок В.5 - Слайд 5: Загальна концепція логіки

Метод RS-аналізу (Просторова область)



1 - Принцип
Розбиття на групи пікселів (n=4), застосування масок

2 - Функція дискримінації
Вимірювання "гладкості" групи.

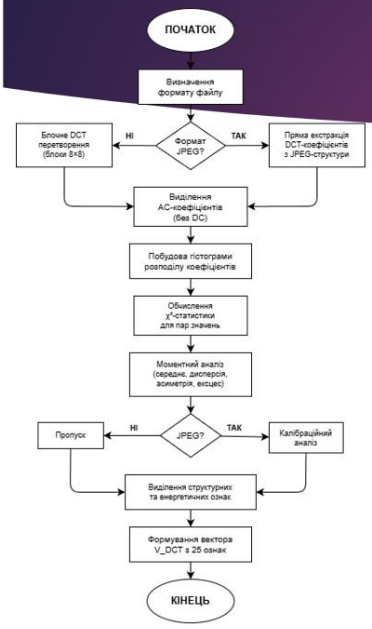
4 - Результат
Виявлення порушення симетрії

3 - Класифікація груп
Регулярні (R), Сингулярні (S), Незмінні (U).

Для просторової області використано вдосконалений RS-аналіз. Алгоритм базується на розбитті зображення на групи пікселів і вимірюванні їх "гладкості" за допомогою функції дискримінації. Ми застосовуємо спеціальні маски, які моделюють шум. У чистих зображеннях існує статистична симетрія між групами. Вбудовування інформації порушує цю рівновагу. Наш модуль генерує 15 ознак, включаючи оцінку довжини повідомлення (p), асиметрію розподілів та варіативність груп. Цей метод надзвичайно точний (90-98%) для LSB-стеганографії.

Рисунок В.6 - Слайд 6: Опис RS - методу

Метод DCT-аналізу (Частотна область)



Методи

- X^2-тест: аналіз пар значень (PoVs).
- Гістограмний аналіз: асиметрія, ексцес, ентропія.
- Калібрація: порівняння з обрізаною та перетиснутою версією.

Ознаки

25 метрик (енергія частот, блокові артефакти).

Оскільки більшість зображень в Інтернеті — це JPEG, ми реалізували глибокий аналіз коефіцієнтів дискретного косинусного перетворення (DCT). Тут застосовується X^2 -тест для виявлення ефекту "вирівнювання пар", характерного для методу JSteg. Додатково ми розраховуємо моменти розподілу (асиметрію, ексцес) та енергетичні характеристики частот. Особливістю є використання калібраційного підходу: ми порівнюємо статистики оригінального зображення з його "відкаліброваною" версією. Це дозволяє виявити сліди втручання навіть у методах типу F5. Всього формується 25 ознак.

Рисунок В.7 - Слайд 7: Опис DCT - методу

Обґрунтування вибору нейронної мережі (CNN)

Чому не SVM?

Потреба виявляти складні *нелінійні* залежності, які створює адаптивна стеганографія.

Чому CNN?

- Адаптивність до нових алгоритмів.
- Можливість обробки як вектора ознак (1D), так і карт залишків (SRM).

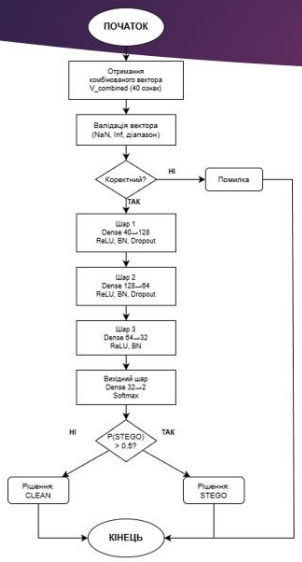
Перевага

Компактність моделі (~16 000 параметрів) проти мільйонів у аналогів → Швидкодія.

Чому для прийняття рішень обрано саме CNN? Традиційні класифікатори, як SVM, добре працюють з лінійними залежностями, але сучасна адаптивна стеганографія створює складні нелінійні спотворення в обох доменах. Нейромережа здатна вивчити ці приховані кореляції (наприклад, взаємозв'язок між асиметрією RS та енергією DCT). Ми використовуємо компактну архітектуру, що має всього близько 16 тисяч параметрів, на відміну від "важких" мереж типу SRNet з 6 мільйонами параметрів. Це забезпечує високу швидкість роботи та запобігає перенавчанню на змісті зображення.

Рисунок В.8 - Слайд 8: Обґрунтування вибору нейронної мережі

Архітектура гібридного класифікатора



1 - Вихід
Sigmoid (ймовірність Stego)
2 - Регуляризація
Batch Normalization, Dropout (0.3)
3 - Приховані шари
128 → 64 → 32 (ReLU)
4 - Вектор 40 ознак + (опціонально) Зображення через шар SRM

Розроблений класифікатор базується на архітектурі багатошарового перцептрона. Вхідний шар приймає нормалізований вектор із 40 ознак. Далі слідують три приховані шари зі зменшенням розмірності (128, 64, 32 нейрони). Ми використали функцію активації ReLU для нелінійності, а також Batch Normalization та Dropout для стабілізації навчання та запобігання перенавчанню.

У повній гібридній реалізації до цієї гілки додається паралельна CNN-гілка з шаром SRM-фільтрів для аналізу піксельних залишків, і їх результати об'єднуються. Це забезпечує максимальну точність.

Рисунок В.9 - Слайд 9: Архітектура гібридного класифікатора

Висновки

1. **Результати розробки**:
Розроблено гібридний метод стегоаналізу, що інтегрує 15 ознак просторового домену (**RS**-аналіз) та 25 ознак частотного домену (**DCT**-аналіз) у єдиний вектор для класифікації нейронною мережею.
Реалізовано паралельно-послідовну архітектуру, де модулі працюють незалежно, що запобігає поширенню помилок між доменами та підвищує загальну надійність системи.

2. **Об'єктивні переваги методу**:
Універсальність: Метод ефективний як проти класичної **LSB**-стеганографії, так і проти сучасних адаптивних алгоритмів (**J-UNIWARD, S-UNIWARD**) завдяки синергії ознак.
Стійкість до **Cover-Source Mismatch**: Використання статистичних ознак замість «сирих» пікселів робить систему менш чутливою до зміни джерела зображення (камери, сканера) порівняно з «чистими» **CNN**.
Обчислювальна ефективність: Розроблений класифікатор має лише ~16 000 параметрів (проти мільйонів у аналогів типу SRNet), що дозволяє проводити аналіз у режимі реального часу.

3. **Виявлені недоліки та обмеження**:
Чутливість до обсягу даних (**Payload**): При низькому заповненні контейнера (менше 0.2 біт на піксель) точність детектування знижується до 60–80%, що є загальною проблемою для методів стегоаналізу.
Надлишковість для **JPEG**: При аналізі виключно стиснутих зображень (**JPEG**) модуль **RS**-аналізу створює додаткове обчислювальне навантаження, хоча його внесок у результат є меншим, ніж у **DCT**-модуля.

4. **Практична реалізація**:
Створено повноцінний програмний комплекс (**Python, FastAPI, PyTorch**) із веб-інтерфейсом, який підтримує режими навчання, аналізу окремих файлів та пакетної обробки.
Система готова до впровадження в контур інформаційної безпеки підприємства для моніторингу медіа-контенту.

Рисунок В.10 - Слайд 10: Висновки

Дякую за увагу

Рисунок В.11 - Слайд 11: Завершення презентації

Додаток Г Протокол перевірки на наявність ознак академічного плагіату

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Вдосконалення методу стегоаналізу зображень у просторовій та частотній областях на основі RS та DCT аналізу з використанням мережі CNN

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра менеджменту та безпеки інформаційних систем
факультет менеджменту та інформаційної безпеки
гр.2КІТС-24м

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КП1) **1,20 %**

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- ☐ **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак академічного плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту**
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки академічного плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. Робота до захисту не приймається.

Експертна комісія:

к.т.н., доцент, зав. каф. МБІС Карпінець В.В. _____

к.ф.-м.н., доцент каф. МБІС Шиян А.А. _____

Особа, відповідальна за перевірку Коваль Н.П. _____

З висновком експертної комісії ознайомлений(-на)

доц. Карпінець

Керівник _____
В.В.

Горохов А.В.

Здобувач _____