Харківський національний університет імені В.Н. Каразіна
Факультет комп'ютерних наук
Безпека інформаційних систем і технологій

«Допущено до захисту»
Зав.кафедрою БІСТ
Сватовський І.І._____
« » червня 2023р.

Пояснювальна записка

до кваліфікаційної роботи бакалавра
спеціальність: 125 Кібербезпека

на тему: «Розробка методики порівняння властивостей стеганографічних контейнерів-зображень з різними характеристиками»

оцінка « » Керівник д. т. н. Олійников Р.В.
(прізвище та ініціали/підпис)

Голова ЕК Рецензент д. ф. к. н. Родінко М.Ю.
(прізвище та ініціали/підпис)

Лемешко О.В._____ Виконавець студент групи КБ-44

Скіпенко М. М.
(прізвище та ініціали/підпис)

Харків – 2023

РЕФЕРАТ

Пояснювальна записка викладена на 86 сторінках, містить 33 рисунки, 5 розділів, 3 таблиці, 30 джерел в переліку посилань.

Метою цього дослідження було розробка методики та програмного застосунку яким було б можливо оцінити придатність контейнера згідно з його характеристик, та провести аналіз які контейнери були б найбільш придатними для різних алгоритмів стеганографії.

Об'єктом дослідження є стеганографічні контейнери-зображення з різними характеристиками, такими як формат файлу, розмір зображення, вміст зображення та інші параметри, що можуть вплинути на процес вбудовування і відновлення прихованої інформації.

Предметом розробки є методика порівняння властивостей стеганографічних контейнерів-зображень з різними характеристиками. Це включає програмну реалізацію якою було б можливо оцінити придатність контейнера.

Методи дослідження включали: вивчення наукого технічної літератури, вибір набору стеганографічних контейнерів-зображень з різними характеристиками, програмна реалізація метрик аналізу придатності, аналіз і порівняння результатів, заснованих на визначених метриках, для визначення найбільш ефективних контейнерів-зображень.

Результатами проведеної роботи є створення методики оцінки придатності контейнерів-зображень та застосунку з відкритим вихідним кодом, здатний визначити корисне навантаження контейнеру та непомітність вбудови даних. Також виявлено бажані властивості для контейнерів, що знижують помітність спотворення при вбудуванні даних. Застосунок може бути застосований для запобігання використання контейнерів несумісних з алгоритмами стеганографії, через замале корисне навантаження або завелику помітність.

Ключові слова: СТЕГАНОГРАФІЯ, ЦИФРОВА СТЕГАНОГРАФІЯ, ЕФЕКТИВНІСТЬ СТЕГО-КОНТЕЙНЕРУ, ВЛАСТИВОСТІ ЗОБРАЖЕННЯ.

ABSTRACT

The explanatory note consists of 86 pages, 33 figures, 5 chapters, 3 tables, and 30 references.

The aim of the thesis was to develop a methodology and software application that would allow to evaluate the suitability of a container according to its characteristics, and to analyze which containers would be most suitable for different steganography algorithms.

The subject matter of the study is steganographic image containers with different characteristics, such as file format, image size, image content, and other parameters that may affect the process of embedding and recovering hidden information.

The scope of study is a methodology for comparing the properties of steganographic image containers with different characteristics. This includes a software implementation that would make it possible to assess the suitability of the container.

The research methods included: studying scientific and technical literature, selecting a set of steganographic image containers with different characteristics, software implementation of suitability analysis metrics, analysis and comparison of results based on the defined metrics to determine the most effective image containers.

The results of this work are the creation of a methodology for assessing the suitability of image containers and an open-source application capable of determining the container payload and the invisibility of data embedding. Desirable properties for containers that reduce the visibility of data embedding distortion were identified. The application can be used to prevent the use of containers incompatible with steganography algorithms, due to too unsufficent payload or too much visibility.

Keywords: STEGANOGRAPHY, DIGITAL STEGANOGRAPHY, STEGO-CONTAINER EFFICIENCY, IMAGE PROPERTIES.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

АРК	–	абсолютна різниця контрасту;
ЗСЛ	–	зорова система людини;
НВК	–	нормована взаємо кореляція;
НСАР	–	нормальна середня абсолютна різниця;
СП	–	середньоквадратична похибка;
ЯК	–	якість кореляції;
ACD	–	Absolute Contrast Difference;
BMP	–	Bitmap;
CQ	–	Correlation Quality;
HSV	–	Hue Saturation Value;
HVS	–	Human Visual System;
JPEG	–	Joint Photographic Experts Group;
LSB	–	Least Significant Bit;
MSE	–	Mean Square Error;
NAAD	–	Normal Average Absolute Difference;
NCC	–	Normalized Cross Correlation;
PNG	–	Portable Network Graphics;
PSNR	–	Peak Signal-to-Noise Ratio;
RGB	–	Red Green Blue;

ВСТУП

У сучасному цифровому світі величезні обсяги чутливої та конфіденційної інформації зберігаються та передаються в електронному вигляді. Сюди входять персональні дані, фінансова інформація, інтелектуальна власність, комерційна та державна таємниця. Уся ця інформація потребує захисту. Заходи кібербезпеки мають вирішальне значення для захисту цієї інформації від несанкціонованого доступу, крадіжки або неправомірного використання. Кібербезпека має важливе значення в сучасному світі. [1]

Стеганографія відіграє важливу роль в інформаційній безпеці. Стеганографія дозволяє непомітно передавати інформацію, вбудовуючи її у файли-носії, які здаються невинними, наприклад, зображення, аудіо- чи відеофайли. Приховуючи існування комунікації, стеганографія допомагає зберегти конфіденційність і запобігає виявленню несанкціонованими особами або системами спостереження. [3]

Завдяки широкому використанню цифрових носіїв та широкого розповсюдження цифрового медіа, стеганографічні методи приховування інформації в тому числі стеганографія на базі графічних контейнерів набули актуальності у сучасному світі. Але при використанні стеганографії може виникнути ситуації при яких інформація буде виявлена через або передана не в повному обсязі, якщо обраний не належний контейнер. [3]

Сама на розробці методів запобігання виявлення інформації і сфокусована ця робота.

1 АКТУАЛЬНИЙ СТАН І ПЕРСПЕКТИВИ РОЗВИТКУ СИСТЕМ СТЕГАНОГРАФІЇ

Стеганографія - це практика приховування повідомлення або інформації всередині іншого повідомлення або носія, наприклад, зображення, аудіофайлу або відео. Мета стеганографії - приховати існування повідомлення таким чином, щоб його не міг легко виявити той, хто не має наміру отримувати це повідомлення. [1]

1.1 Історія стеганографії

Історія стеганографії налічує тисячі років, і ця практика використовувалася окремими особами та організаціями протягом всієї історії для приховування повідомлень та інформації.

Геродот, давньогрецький історик і письменник, у 474 році до н.е. описав, як грецький полководець Гістіей, який потрапив у полон до персів, використовував стеганографію, щоб надсилати повідомлення своїм союзникам у Греції. А, Йоганн Тритемій був німецьким абатом, криптографом та істориком, якого часто вважають одним з найперших відомих авторів стеганографії. Наприкінці 15 століття Тритемій написав "Steganographia" - трактат про криптографію і стеганографію, в якому досліджував використання методів прихованого письма для кодування повідомлень. [1, 2]

1.2 Актуальний стан стеганографії

Стеганографія набула популярності в останні десятиліття завдяки кільком факторам. [1, 3]

- Збільшення використання цифрових носіїв: З широким розповсюдженням цифрових медіа, таких як зображення, аудіофайли та відео, стеганографія стала більш практичним та ефективним методом приховування повідомлень. Цифровими медіа можна легко маніпулювати і змінювати, що робить їх ідеальним середовищем для приховування прихованої інформації. [3]

- Удосконалення стеганографічних методів: Стеганографія розвинулася і включає в себе більш досконалі методи, такі як алгоритми глибокого навчання, які полегшують і роблять більш ефективним приховування повідомлень в цифрових медіа. Як результат, стеганографія стала більш доступною і практичною для різноманітних застосувань.
- Нові технології та застосунки: Стеганографію використовують у нових додатках, що з'являються, таких як блокчейн та Інтернет речей (IoT). Оскільки ці технології продовжують розвиватися, стеганографія, ймовірно, відіграватиме дедалі важливішу роль у захисті конфіденційної інформації. [1].
- Підвищення обізнаності про кібербезпеку: Оскільки загрози кібербезпеці стають все більш поширеними і витонченими, окремі особи та організації все більше усвідомлюють необхідність захисту конфіденційної інформації. Стеганографія пропонує спосіб захистити дані, приховуючи їх у відкритому доступі, а не покладаючись на традиційні методи шифрування.

Слід відмітити, що стеганографія часто використовується в поєднанні з криптографією для забезпечення додаткового рівня безпеки повідомлення або інформації, що передається. Це пов'язано з тим, що навіть якщо перехоплене повідомлення розшифрує неавторизована сторона, стеганографічний метод, використаний для приховування повідомлення, може ускладнити визначення того, чи є приховане повідомлення. [3]

Стеганографія має різні сучасні застосування, особливо у сфері цифрового зв'язку та безпеки даних. [1]

- Приховування інформації: Стеганографія може використовуватися для приховування конфіденційної інформації, такої як паролі, ключі шифрування або цифрові підписи, в цифрових носіях, таких як зображення, відео або аудіофайли. Це може допомогти захистити конфіденційну інформацію від несанкціонованого доступу або перехоплення під час передачі.

- Цифрові водяні знаки: Стеганографія може використовуватися для вбудовування цифрових водяних знаків у зображення або відео для захисту авторських прав або ідентифікації власника. Цифровий водяний знак - це унікальний підпис або ідентифікатор, який вбудовується в цифровий медіафайл для ідентифікації його джерела або автора.
- Прихована комунікація: Стеганографія може використовуватися для прихованої передачі повідомлень через публічні канали зв'язку, такі як Інтернет або соціальні мережі. Вбудовуючи повідомлення у, здавалося б, нешкідливі цифрові носії, такі як зображення або відео, стеганографія дозволяє здійснювати приховану комунікацію, не викликаючи підозр.
- Стеганоаналіз: Стеганографія також може бути використана для виявлення та аналізу прихованих даних у цифрових носіях. Методи стеганоаналізу використовуються для виявлення та вилучення прихованих даних зі стего-зображень або інших медіафайлів, які можуть бути використані для судово-медичної експертизи, правоохоронних органів або збору розвідувальної інформації.

1.3 Мета та цілі дослідження

Предмет – Стеганографічні контейнери-зображення.

Мета – Розробка методики порівняння властивостей стеганографічних контейнерів-зображень з різними характеристиками.

Цілі:

- вивчення принципів побудови стеганографічних систем;
- виділення властивостей зображень;
- вивчення можливих способів оцінки ефективності стеганографічних контейнерів-зображень, виходячи з властивостей;
- розробка методики порівняння стеганографічних контейнерів-зображень;
- програмна реалізація методики та її тестування.

2 ПРИНЦИПИ ПОБУДОВИ СТЕГАНОГРАФІЧНИХ СИСТЕМ

Стеганографія на базі графічних контейнерів (або ж стеганографія зображень, стеганографія цифрових зображень) - це практика приховування інформації в цифрових зображеннях, не впливаючи на якість або візуальний вигляд зображення. Мета цифрової стеганографії зображень, як і будь-якої стеганографічної практики - зробити інформацію невидимою для неозброєного ока і непомітною для автоматизованих методів виявлення. [3, 4]

2.1 Основні компоненти систем на базі графічних контейнерів

Виділяють такі основні компоненти систем на базі графічних контейнерів [4, 5]:

Обкладинка (або ж контейнер) – це зображення, яке використовується як носій для приховування даних. Обкладинка може бути будь-якого формату, наприклад, JPEG, BMP, PNG і т.д., а її розмір і роздільна здатність можуть бути різними.

Проховувані дані – це дані, які повинні бути приховані на обкладинці. Ці дані можуть бути будь-якого формату, наприклад, текст, зображення, аудіо або відео, а їхній розмір може змінюватися залежно від ємності обкладинки.

Стего-ключ – це ключ або пароль використовується у деяких стеганографічних алгоритмах для шифрування даних перед їх вбудовуванням у зображення обкладинки. Це додає додатковий рівень безпеки і ускладнює доступ неавторизованих осіб до прихованих даних.

Стего-зображення – це зображення, модифіковане для приховування даних за допомогою методів стеганографії. По суті, це зображення-обкладинка, з яким маніпулювали, щоб вбудувати інформацію таким чином, щоб її не було видно для людського ока. Процес створення стего-зображення включає в себе вбудовування даних у зображення обкладинки, гарантуючи при цьому, що зображення обкладинки все ще виглядає так, як і раніше, для людського ока. Це досягається шляхом внесення невеликих, непомітних змін до пікселів обкладинки, таких як

модифікація їхніх найменш значущих бітів, або за допомогою більш складних методів, таких як перетворення в частотній області.

Стего-алгоритм - це алгоритм, що використовується для приховування даних на зображенні обкладинки. Вибір алгоритму залежить від таких факторів, як тип даних, які потрібно приховати, ємність обкладинки та необхідний рівень безпеки. Він використовується для вбудовування даних у зображення прикриття та вилучення їх зі стего-зображення відповідно.

Діаграма функціонування стеганографічної системи на базі графічних контейнерів приведено на рисунку 2.1 (слід відмітити що «Стего-ключ» не завжди є обов'язковою частиною системи).

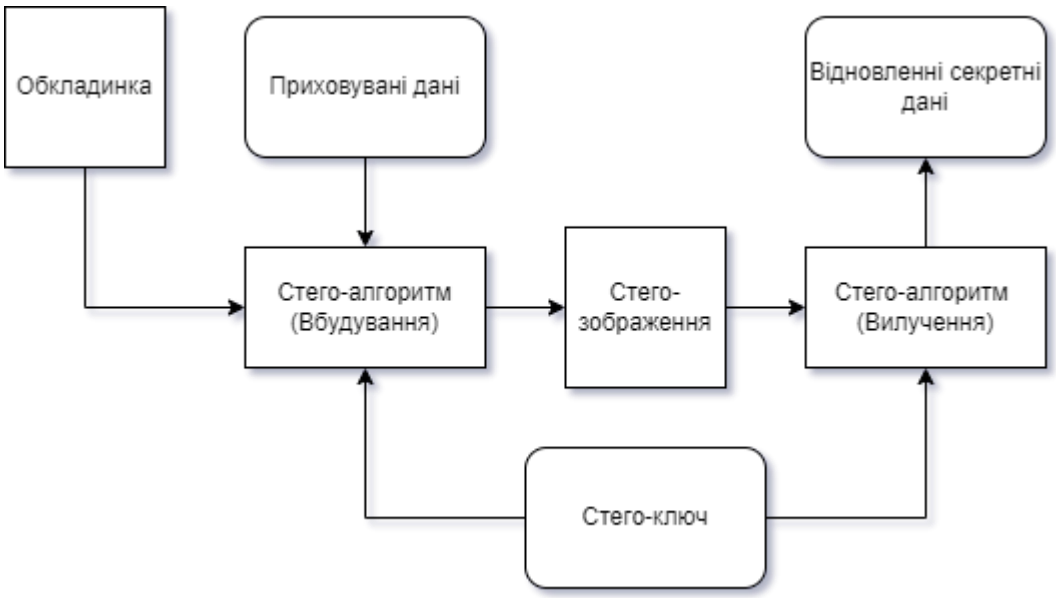

Рисунок 2.1 – Діаграма системи на базі графічних контейнерів.

2.2 Основні властивості стеганографії на базі графічних контейнерів

Стеганографія на базі графічних контейнерів базується на властивостях зорової системи людини (ЗСЛ, Human Visual System, HVS). У контексті стеганографії властивості ЗСЛ часто розглядаються як засіб підвищення ефективності стеганографічних методів. Зорова система людини має певні характеристики, які можуть бути використані для приховування інформації в цифрових носіях, мінімізуючи при цьому сприйнятливість прихованих даних для людини-спостерігача. Надалі наведено властивості зорової системи людини, які мають відношення до стеганографії. [6, 7, 8]

– *Низька чутливість до незначних змін яскравості зображення*: ЗСЛ менш чутливий до невеликих змін яскравості зображення. Ця властивість використовується в стеганографії для вбудовування прихованої інформації шляхом внесення незначних змін до значень яскравості пікселів. Чутливість ока до дії випромінювання визначається величиною, яка е зворотною до яскравості L, що викликає граничне роздратування (2.1):

$$v = \frac{1}{L_{\text{П}}} \qquad (2.1).$$

Чутливість може виражатися і в одиницях, зворотних до граничної освітленості спостережуваного зображення. Дослідження показали, що поріг чутливості ЗСЛ до зміни яскравості дорівнює 2-3 %. [3, 6], і таким чином якщо яскравість кодується L_m = 256 рівнями квантування зміна яскравості на ΔL = 8 рівнів, задовільнює цей параметр

$$\Delta = \frac{\Delta L}{L_m} 100\% = \frac{8}{256} 100\% \approx 8\% \qquad (2.2).$$

– *Низька чутливість до незначних змін яскравості синього каналу в кольорі зображення*: ЗСЛ має відносно нижчу чутливість до невеликих змін яскравості синього каналу в кольорі зображення порівняно з іншими колірними каналами. Ця властивість використовується в методах стеганографії, які маніпулюють синім каналом для вбудовування прихованої інформації. Різна чутливість ЗСЛ до кольорових складових растрових даних виражається в оцінці повнокольорової яскравості пікселя (2.2): [3, 7]

$$Y = 0{,}58662 \cdot G + 0{,}2989 \cdot R + 0{,}11448 \cdot B \qquad (2.3).$$

– *Низька чутливість до незначних змін контрастності зображення*: Подібно до яскравості, ЗСЛ також демонструє низьку чутливість до незначних змін контрасту зображення. Стеганографічні методи використовують цю властивість, регулюючи рівень контрастності зображення обкладинки для вбудовування прихованої інформації. [6, 8]

- *Частотна чутливість*: ЗСЛ має різні рівні чутливості до різних частот візуальної інформації. Він більш чутливий до змін низькочастотних компонентів і менш чутливий до змін високочастотних компонентів. Алгоритми стеганографії використовують цю властивість для вбудовування прихованої інформації в менш візуально помітні високочастотні компоненти зображення, такі як краї або текстуровані області. [7]
- *Ефект маскування*: ЗСЛ демонструє ефект маскування, коли сприйняття одного візуального стимулу може пригнічувати сприйняття сусідніх або одночасних стимулів. Ця властивість використовується в стеганографії для вбудовування прихованої інформації, використовуючи явище візуального маскування. Приховані дані вбудовуються в області зображення, де присутність іншого візуально домінуючого вмісту маскує їх присутність, що ускладнює їх виявлення. [3]

3 ОСОБЛИВОСТІ ВИКОРИСТАННЯ ГРАФІЧНИХ СТЕГОКОНТЕЙНЕРІВ

Використання цифрової стеганографії зображень пов'язане з деякими особливостями, які виникають через методи, що застосовуються, та властивості зображень обкладинок. Врахування цих особливостей має вирішальне значення для розробки ефективних і безпечних стеганографічних систем. Вибір стеганографічного методу, розуміння властивостей зображень обкладинок та їх форматів, забезпечення непомітності, а також протидія потенційним атакам і методам стеганоаналізу сприяють успішному впровадженню цифрової стеганографії зображень. [3]

3.1 Методи стеганографії

Як зазначено у розділі 2.1 стеганографічний метод (стего-алгоритм) - це метод, який використовується для приховування інформації в об'єкті прикриття, такому як зображення, аудіофайл або відео, у спосіб, непомітний для людських органів чуття або статистичного аналізу. Хоча конкретні стеганографічні методи можуть відрізнятися за своєю реалізацією, вони, як правило, слідують схожому процесу з певними спільними елементами. В даній роботі було реалізовано декілька стего-алгоритмів. [4, 5]

3.1.1 Метод заміни найменш значимих бітів

Метод заміни найменш значимих бітів (Least Significant Bit, LSB) є одним з найпоширеніших методів в стеганографії для приховування інформації в цифрових медіафайлах, таких як зображення або звукові файли. Його основна ідея полягає в тому, що найменш значущі біти пікселів замінюються бітами прихованої інформації. [3, 9]

Основні кроки методу заміни найменш значимих бітів в стеганографії наступні:

- Кодування інформації: Інформація, яку потрібно приховати, кодується у вигляді бітів. Ці біти вбудовуються в найменш значущі біти пікселів.

Тобто, найменш значущий біт пікселя може бути заміненний на один з бітів прихованої інформації. Алгоритм наведений на рисунку 3.1.

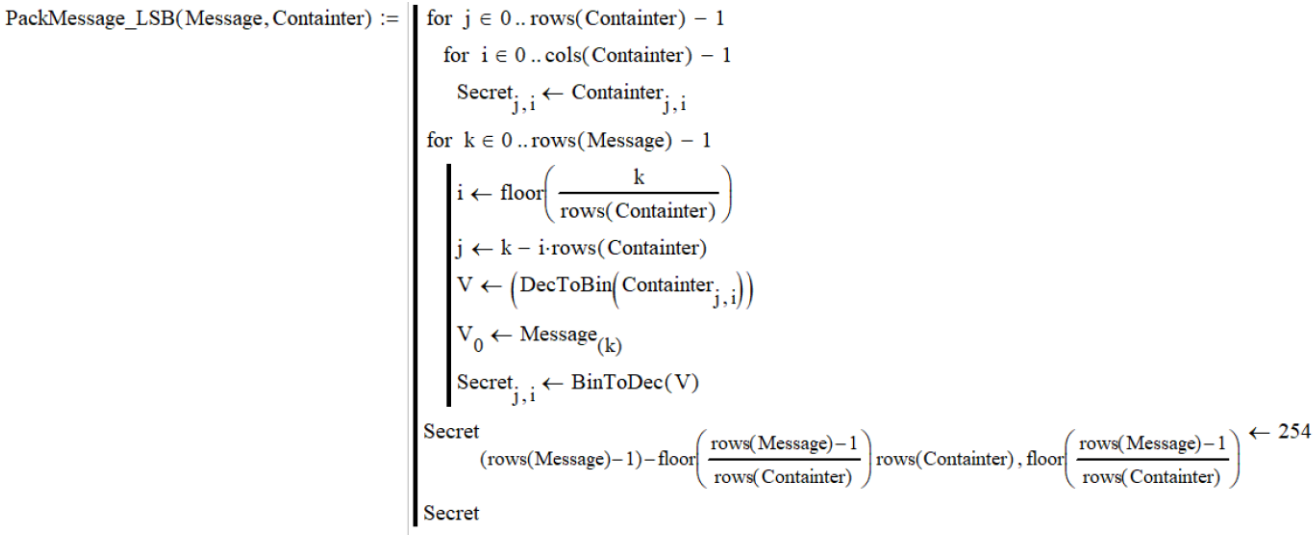

Рисунок 3.1 – Алгоритм вбудування інформації методу заміни найменш значимих бітів

– Відновлення інформації: Для отримання прихованої інформації процес виконується в зворотному порядку. Найменш значущі біти пікселів витягуються, і з них відновлюється прихована інформація. Алгоритм наведений на рисунку 3.2.

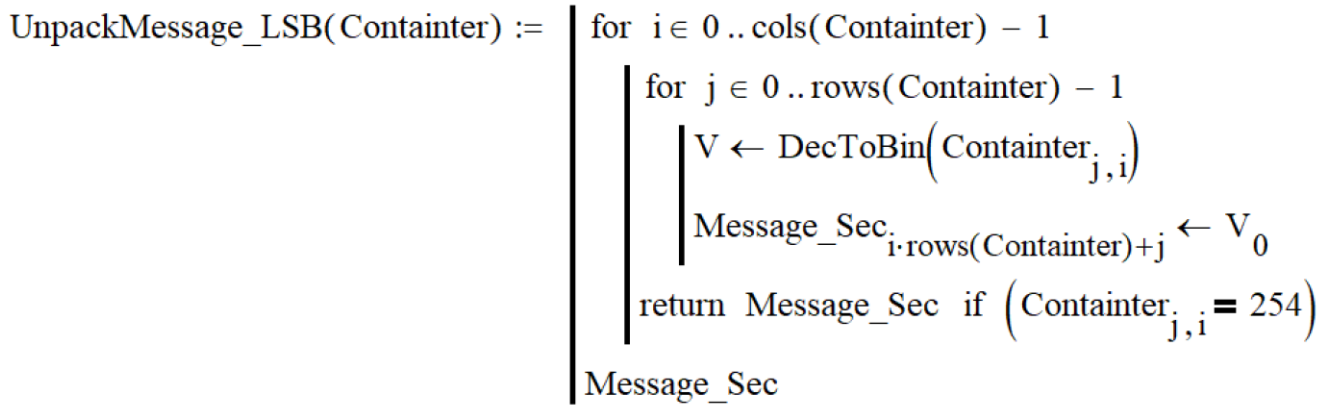

Рисунок 3.2 – Алгоритм відновлення інформації методу заміни найменш значимих бітів

Метод заміни найменш значимих бітів є відносно простим для реалізації, але він може бути вразливим до атак, які спрямовані на виявлення та вилучення прихованої інформації. Для підвищення стійкості і захищеності методу можуть

використовуватись додаткові техніки, такі як перестановка бітів або використання ключа шифрування. [3]

3.1.2 Метод блочного приховування

Метод блочного приховування розділяє основний зображення на блоки і встраює приховану інформацію в кожен блок незалежно, зазвичай використовуючи метод заміни найменш значимих бітів, тому іноді його називають модифікацією LSB. [11, 12]

Основні кроки методу блочного приховування в стеганографії наступні:

- Розбиття на блоки: Оригінальні дані (наприклад, зображення або відео) розбиваються на блоки однакового розміру. Розмір блоку може бути вибраний в залежності від конкретного алгоритму або вимог застосовуваної стеганографічної системи. Варто відмітити при цьому що способ розділення контейнеру може служити ключем системи. Запропонована реалізація методу передбачає адаптивне розбиття на блоки за колонками або стовпцями пікселів для підвищення корисного навантаження.
- Вбудування інформації: Інформація, яку потрібно приховати, кодується у вигляді бітів. Ці біти вбудовуються в блоки даних. Для кожного з даних блоків вираховується біт парності. В кожному такому блоці приховується один біт повідомлення. У випадку, коли біт парності не дорівнює біту повідомлення, виконується інвертування одного з найменш значимих бітів блока. Алгоритм наведений на рисунку 3.3.

$$
\begin{aligned}
&\text{PackMessage\_BH}(\text{Message},\text{Container}) := \\
&\quad \text{for } i \in 0..\text{cols}(\text{Container}) - 1 \\
&\quad\quad b \leftarrow \text{mod}\left(\sum_{j=0}^{\text{rows}(\text{Container})-1} \text{Container}_{j,i}, 2\right) \\
&\quad\quad \text{if } \text{Message} \neq b \\
&\quad\quad\quad P \leftarrow \text{DecToBin}\left(B_{0,i}\right) \\
&\quad\quad\quad P_0 \leftarrow P_0 \oplus 1 \\
&\quad\quad\quad \text{Secret}_{0,i} \leftarrow \text{BinToDec}(P) \\
&\quad\quad \text{Secret}_{0,i} \leftarrow B_{0,i} \text{ if } \text{Message}_i = b \\
&\quad\quad \text{for } j \in 1..\text{rows}(B) - 1 \\
&\quad\quad\quad \text{Secret}_{j,i} \leftarrow \text{Container}_{j,i} \\
&\quad \text{Secret}
\end{aligned}
$$

Рисунок 3.3 – Алгоритм вбудування інформації методу блочного приховування

– Відновлення інформації: Для отримання прихованої інформації процес виконується в зворотному порядку. Блоки даних витягуються, і з них відновлюється прихована інформація. Алгоритм наведений на рисунку 3.4.

$$
\begin{aligned}
&\text{UnpackMessage\_BH}(\text{Container}) := \\
&\quad \text{for } i \in 0..\text{cols}(\text{Container}) - 1 \\
&\quad\quad \text{Message}_i \leftarrow \text{mod}\left(\sum_{j=0}^{\text{rows}(\text{Container})-1} \text{Container}_{j,i}, 2\right) \\
&\quad \text{Message}
\end{aligned}
$$

Рисунок 3.4 – Алгоритм відновлення інформації методу блочного приховування

Метод блочного приховування може бути ефективним для приховування інформації в цифрових носіях. Однак, він може бути вразливим до атак, таких як статистичний аналіз, що спрямовані на виявлення змін у блоках даних. Для підвищення стійкості методу можуть використовуватись додаткові техніки, такі як шифрування інформації або застосування складних алгоритмів вбудовування. [3, 12]

3.1.3 Метод псевдовипадкової перестановки

Метод псевдовипадкової перестановки в стеганографії є одним зі способів приховання інформації в цифрових медіафайлах, зокрема зображеннях. Його основна ідея полягає в тому, щоб переставити пікселі або блоки пікселів у зображенні за допомогою псевдовипадкового алгоритму з метою приховати інформацію. [1, 2, 10]

Процес використання методу псевдовипадкової перестановки може бути наступним:

– Вибір ключа: Генерується псевдовипадковий ключ, значення якого не повторюються і є індексами пікселів в зображенні. Алгоритм наведений на рисунку 3.5.

```
GenerateKey_PRP(Container) := | for i ∈ 0 .. cols(Container) − 1
                              |   Key_i ← floor(rnd(rows(Container)))
                              | Key
```

Рисунок 3.5 – Алгоритм генерації ключа методу псевдовипадкової перестановки

– Приховання інформації: Інформація, яку потрібно приховати, кодується у вигляді бітів. Відповідні біти повідомлення записуються в пікселі яку були відповідно зазначені ключем. Алгоритм наведений на рисунку 3.6.

```
PackMessage_PRP(Message, Container, Key) :=
    for j ∈ 0 .. rows(Container) − 1
      for i ∈ 0 .. cols(Container) − 1
        Secret_{j,i} ← Container_{j,i}
    for i ∈ 0 .. cols(Container) − 1
      V ← DecToBin(Container_{Key_i, i})
      V_0 ← Message_(i)
      Secret_{Key_i, i} ← BinToDec(V)
    Secret
```

Рисунок 3.6 – Алгоритм вбудування інформації методу псевдовипадкової перестановки

– Відновлення інформації: Для отримання прихованої інформації процес виконується в зворотному порядку. З використанням ключа, витягують значення бітів відповідних пікселів. Алгоритм наведений на рисунку 3.7.

```
UnpackMessage_PRP(Message_Sec, Containter, Key) :=
    for i ∈ 0 .. cols(Containter) − 1
      V ← DecToBin(Containter_{Key_i, i})
      Message_Sec_i ← V_0
    Message_Sec
```

Рисунок 3.7 – Алгоритм відновлення інформації методу псевдовипадкової перестановки

3.1.4 Метод квантування зображення

Метод квантування зображення в стеганографії використовується для приховування інформації в квантованому зображенні. Квантування - це процес розділення діапазону значень пікселів на певну кількість рівнів або категорій. У контексті стеганографії, цей метод використовується для вбудовування бітів інформації в найменш значущі біти пікселів, що піддаються квантуванню. [10]

Суть методу заключається в тому, що між сусідніми пікселями c_i та c_{i+1} є різниця, наприклад ∂_i. Якщо цю різницю задати параметром для міжпіксельної функції Ω, то вийде Ω*:* $\Delta_i = \Omega(c_i - c_{i+1})$, де Δ_i - дискретна апроксимація різниці

сигналів c_i - c_{i+1}. Для приховування інформації даний метод вираховує різницю Δ_i для кожного і-го біта. При цьому, формується таблиця відповідності, в якій кожному можливому значенню Δ_i ставиться у відповідність деякий біт b_i. У випадку коли b_i не відповідає прихованому біту, який потрібно приховати, то значення Δ_i замінюється найближчим Δ_j, для якого виконується дана умова. При цьому відповідним чином змінюється інтенсивність пікселів, між якими вираховується різниця Δ_i. [13]

Основні кроки методу квантування зображення в стеганографії наступні:

- Таблиця відповідності: Сформована таблиця відповідності також може слугувати ключем так як часто формується псевдовипадково. Алгоритм наведений на рисунку 3.8.

```
GenerateKey_IQ(Key) := | for i ∈ 0..510
                       |     Key_{0,i} ← i − 255
                       |     Key_{1,i} ← ceil(rnd(2)) − 1
                       | Key
```

Рисунок 3.8 – Алгоритм генерації ключа методу квантування зображення

- Вбудовування інформації: Інформація, яку потрібно приховати, кодується у вигляді бітів. Ці біти вбудовуються в найменш значущі біти пікселів, які піддаються квантуванню. Алгоритм наведений на рисунку 3.9.

```
PackMessage_IQ(Message, Container) :=
  for i ∈ 0 .. rows(Container) − 1
    b ← Container_{i,0} − Container_{i,1}
    Secret_{i,0} ← Container_{i,0}  if  Message_i = Key_{1,b+255}
    if  Message_i ≠ Key_{1,b+255}
      j ← 1
      while  Message_i ≠ Key_{1,b+255+j} ∧ j < 509
        j ← j + 1
      Secret_{i,0} ← Container_{i,0} + Key_{0,b+255+j} − b
    for  j ∈ 1 .. cols(Container) − 1
      Secret_{i,j} ← Container_{i,j}
  Secret
```

Рисунок 3.9 – Алгоритм вбудовування інформації методу квантування зображення

- Відновлення інформації: Для отримання прихованої інформації процес виконується в зворотному порядку. Зображення піддається квантуванню, і змінені значення пікселів витягуються, щоб відновити приховану інформацію. Алгоритм наведений на рисунку 3.10.

```
UnpackMessage_IQ(Conteiner) :=
  for  i ∈ 0 .. rows(Conteiner) − 1
    b ← Conteiner_{i,0} − Conteiner_{i,1}
    Message_i ← Key_{1,b+255}
  Message
```

Рисунок 3.10 – Алгоритм відновлення інформації методу квантування зображення

Важливо зазначити, що квантування може вплинути на якість зображення, особливо при значній бітовій точності або великій кількості вбудованої інформації. Підбір оптимальних параметрів квантування і перевірка стійкості методу до різних атак є важливими аспектами при використанні цього методу стеганографії. [3]

3.2 Формати цифрових зображень

Формати зображень - це формати файлів, які використовуються для зберігання та кодування цифрових зображень. Існує безліч форматів зображень, кожен з яких має свої характеристики, особливості та призначення. Вибір формату зображення залежить від таких факторів, як тип зображення, призначення, бажана якість і вимоги до сумісності [3]. В даній роботі використовувалися формати зі стиснення без втрат, тобто такі при використанні яких закодована інформація може бути повністю відновлена зі стиснутих даних, а саме BMP та PNG.

3.2.1 Формат BMP

Формат BMP (Bitmap) - це широко використовуваний формат растрових зображень, розроблений компанією Microsoft. Він зберігає растрові зображення з деталізацією на рівні пікселів, підтримуючи як нестиснуті, так і стиснуті дані. Нижче наведено опис формату BMP. [14 - 16]

Заголовок файлу:

- Файл BMP починається з 14-байтового заголовка, який містить інформацію про формат і розмір файлу.
- Він включає підпис файлу, який зазвичай має вигляд "BM" (для Windows) або "BA" (для OS/2).
- Він визначає розмір BMP-файлу в байтах.
- Включає зарезервовані поля та зсув до піксельних даних.

Заголовок растрової інформації:

- Після заголовка файлу йде заголовок растрової інформації, який містить детальну інформацію про дані зображення.
- Він включає розмір самого заголовка, зазвичай 40 байт для Windows BMP.
- Він визначає ширину і висоту зображення в пікселях.
- Визначає кількість колірних площин (зазвичай 1) і кількість біт на піксель.
- Вказує на метод стиснення, що використовується, наприклад, без стиснення, RLE (Run-Length Encoding) або інші алгоритми стиснення.

- Вказує розмір даних зображення у байтах і надає інформацію про роздільну здатність.

Палітра кольорів (не обов'язково):

- Для індексованих кольорових зображень після заголовка растрової інформації може міститися палітра кольорів.
- Палітра кольорів містить список елементів кольорів, які визначають доступні кольори у зображенні.
- Кожен елемент кольору складається з червоного, зеленого, синього та зарезервованих компонентів, які визначають значення RGB певного колірного індексу.

Піксельні дані:

- Піксельні дані представляють фактичні пікселі зображення і відповідають колірній палітрі (якщо вона присутня).
- Піксельні дані зберігаються рядок за рядком, знизу вгору, і кожен рядок має пробіл, щоб забезпечити вирівнювання за межами байта.
- Формат пікселів залежить від кількості бітів на піксель, вказаної у заголовку растрової інформації.
- Наприклад, якщо кожен піксель представлений 24 бітами (по 8 бітів для кожного каналу RGB), піксельні дані зберігатимуть послідовно значення RGB кожного пікселя.

Формат BMP підтримує різні глибини кольору, зокрема 1-бітову, 4-бітову, 8-бітну, 16-бітну, 24-бітну та 32-бітну. Він дозволяє зберігати як нестиснуті, так і стиснуті дані зображення, забезпечуючи гнучкість щодо розміру файлу та якості зображення.

Важливо зазначити, що існують різні версії та варіації формату BMP, зокрема Windows BMP та OS/2 BMP. Кожна версія може мати специфічні особливості, розширення або обмеження. Наведений вище опис фокусується на найпоширенішому форматі Windows BMP.

3.2.2 Формат PNG

Формат PNG (Portable Network Graphics) - це широко використовуваний формат растрових зображень, який був розроблений як заміна формату GIF (Graphics Interchange Format) для подолання його обмежень. Файли PNG підтримують стиснення без втрат і пропонують різні функції, які роблять їх придатними для зберігання та передачі зображень. Нижче наведено опис формату PNG: [17, 18]

Заголовок файлу:

- PNG-файл починається з 8-байтової сигнатури, яка складається з певних значень байт, що ідентифікують файл як PNG-зображення.
- За підписом слідує послідовність фрагментів, які зберігають різні частини даних зображення.

Блок IHDR:

- Блок IHDR (заголовок зображення) є першим блоком після заголовка файлу і містить основну інформацію про зображення.
- Вона включає такі дані, як ширина і висота зображення в пікселях, розрядність на канал, тип кольору, метод стиснення, метод фільтрації і метод черезстрочної розгортки.

Фрагмент PLTE (для індексованих кольорових зображень):

- Якщо зображення використовує індексовану кольорову палітру, може бути присутнім фрагмент PLTE (палітра).
- У фрагменті PLTE зберігаються RGB-значення кольорів у палітрі.

Фрагменти IDAT:

- Фактичні дані зображення зберігаються в одному або декількох фрагментах IDAT (Image Data).
- Дані зображення стискаються за допомогою алгоритму стиснення DEFLATE, що забезпечує ефективне зберігання без втрати інформації.

Інші фрагменти:

- Файли PNG можуть містити додаткові необов'язкові фрагменти, які надають додаткову інформацію або функції.

- Деякі часто використовувані фрагменти включають tEXt (текстову інформацію), tIME (мітку часу), pHYs (фізичні розміри пікселів) тощо.
- Ці фрагменти можуть зберігати метадані, інформацію про прозорість та інші допоміжні дані.

Фрагмент IEND:

- Чанк IEND (кінець) позначає кінець PNG-файлу.
- Цей фрагмент фіксованої довжини, який не містить жодних даних.

Формат PNG має кілька переваг над іншими форматами зображень, зокрема стиснення без втрат, підтримка прозорості (як двійкового, так і альфа-каналу), а також можливість зберігати багатшу інформацію про колір. Файли PNG широко використовуються в Інтернеті, цифровому мистецтві та інших сферах, де важлива якість і гнучкість зображень.

Важливо зазначити, що специфікація формату PNG допускає різні варіації та розширення, але наведений вище опис охоплює основні елементи стандартного PNG-зображення.

3.3 Властивості цифрових зображень

Властивості зображення є важливими в стеганографії для визначення стеганографічної придатності, вибору відповідних методів вбудовування, протидії стегоаналізу, збереження якості зображення та забезпечення сумісності з різними форматами зображень. Використовуючи та розуміючи ці властивості, стеганографи можуть розробляти ефективні та непомітні методи приховування, зберігаючи цілісність та візуальну якість зображення обкладинки.

3.3.1 Вміст

Для цілей цього дослідження було виділено таку властивість зображення як вміст та наступні категорії для демонстрації слабких місць стеганографічної системи а також для тестування на поширених на практиці прикладах. Далі перелічені та описані ці категорії.

- Пейзаж: Пейзажні зображення зазвичай містять природні сцени, такі як гори, ліси або морські пейзажі. Ці зображення часто мають вищий рівень складності та візуальних деталей. У стеганографії ландшафтні

зображення можуть запропонувати більше можливостей для вбудовування даних завдяки своїй багатій текстурі та різноманітному розподілу кольорів. Варіації природних елементів у ландшафтних зображеннях можуть допомогти більш ефективно приховати вбудовані дані. Вони є часто майже ідеальним варіантом у прикладному контексті.

- Портрет: Портретні зображення зосереджені на захопленні людських облич та виразів. Вони, як правило, мають більш помітний об'єкт і простіше тло. З точки зору стеганографії, портретні зображення можуть становити певні труднощі, оскільки будь-які зміни або спотворення, спричинені вбудовуванням даних, можуть потенційно вплинути на розпізнавання та сприйняття людини на зображенні. Необхідно подбати про те, щоб вбудовані дані суттєво не впливали на риси обличчя або загальну якість портрета.
- Документ: Зображення документів включають відскановані або цифрові зображення текстових або графічних документів, фотокопії та скріншоти, наприклад листів, контрактів або схем. Ці зображення зазвичай мають високий рівень структури та містять певні візерунки, фігури та текст. У стеганографії зображення документа можуть потребувати різних стратегій вбудовування, щоб гарантувати, що приховані дані не порушують читабельність або цілісність вмісту документа. У процесі вбудовування слід враховувати збереження розбірливості тексту та уникати введення помітних артефактів, які можуть скомпрометувати інформацію документа.
- Монотонний колір: Монотонні зображення мають один домінуючий колір. У стеганографії монотонні зображення є дуже проблемними через обмежений колірний простір. Вбудовуючи дані в такі зображення, необхідно майже неможливо зберегти непомітними, це найгірший можливий випадок для стеганографічної системи.

– Шум: Шумові зображення генеруються або маніпулюються, щоб містити випадкові варіації або патерни. Вони, як правило, не мають чіткої структури або змістовного наповнення. Шумові зображення є важливими в стеганографії, оскільки вони надають можливість приховувати дані в областях з високим рівнем випадковості та низькою візуальною значущістю. Вбудовування даних у шумові зображення може використовувати властиві їм нерівності та варіації, щоб зробити приховану інформацію менш помітною. Загалом такі зображення можна назвати найкращим випадком для стеганографічної системи. Було застосовано два види шуму:

  - Гауссівський шум, також відомий як адитивний білий гауссівський шум - це тип випадкового шуму, який підпорядковується гауссівському розподілу. Це поширений тип шуму, який зустрічається в різних сферах, включаючи обробку зображень та аналіз сигналів. Гауссівський шум характеризується своїми статистичними властивостями, зокрема, середнім значенням і стандартним відхиленням. Математична репрезентація гауссівського шуму: [19]

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.1).$$

    де μ – математичне сподівання x,

    σ – середнє квадратичне відхилення.

  - Рожевий шум, також відомий як 1/f шум або фрактальний шум, - це тип випадкового сигналу, який має однакову енергію в кожній октаві або смузі частот. Він характеризується спектром потужності, який зменшується на 3 децибели (дБ) на октаву. Рожевий шум отримав свою назву через схожість зі звуком постійного дощу або бурхливого водоспаду, який має заспокійливу і збалансовану якість. Математична репрезентація фрактальний шуму: [20]

$$h(x) = \sigma\sqrt{\frac{N}{2}}\sum_u \frac{\chi_u}{u}\sin(\frac{2\pi ux}{N} + \phi_u) \quad (3.2).,$$

3.3.2 Корисне навантаження

Корисне навантаження обкладинки - це кількість прихованих даних, які можна вбудувати в зображення за допомогою стеганографічного методу. Це здатність зображення нести приховану інформацію, не спричиняючи помітних змін у його візуальному вигляді. [1, 3]

Корисне навантаження зазвичай вимірюється в бітах або байтах і являє собою максимальну кількість прихованих даних, які можна вбудувати в зображення обкладинки. Обсяг корисного навантаження залежить від різних факторів, включаючи формат зображення, розмір зображення, використовуваний стеганографічний алгоритм і бажаний рівень непомітності. [3]

При використанні стеганографії мета полягає в тому, щоб максимізувати корисне навантаження, гарантуючи при цьому, що приховані дані залишаться невидимими для спостерігача і стійкими до різних операцій обробки зображень. Для досягнення балансу між корисним навантаженням і непомітністю часто використовуються такі методи, як вбудовування LSB (Least Significant Bit), методи частотної області або адаптивні алгоритми вбудовування.

Важливо заначити, що корисне навантаження зображення для обкладинки слід вибирати ретельно, враховуючи обмеження стеганографічного алгоритму, характер прихованих даних і передбачуване застосування. Більш високе корисне навантаження може збільшити ризик виявлення або погіршення якості зображення, в той час як низьке корисне навантаження може обмежити кількість інформації, яку можна приховати. [3, 9]

3.3.3 Контраст

Контраст - це різниця в яскравості та кольорі між різними частинами зображення. Це ключова візуальна характеристика, яка визначає рівень розрізнення елементів на зображенні. Висококонтрастні зображення демонструють значну різницю між світлими і темними ділянками, в той час як низькоконтрастні мають менш виражені варіації яскравості.

У контексті обробки зображень і стеганографії контрастність відіграє важливу роль у визначенні видимості прихованої інформації. Стеганографічні методи спрямовані на вбудовування даних таким чином, щоб мінімально впливати на загальний візуальний вигляд зображення, в тому числі на його контрастність. [3]

При роботі зі стеганографією дуже важливо враховувати контрастність зображення обкладинки, щоб переконатися, що приховані дані залишаються непомітними. Вбудовування занадто великої кількості даних або використання невідповідних методів вбудовування може потенційно змінити контрастні характеристики зображення, що призведе до змін, які можна виявити. Контрастність можливо визначити наступним чином: [21]

$$C = \max(x) - \min(x) \quad (3.3),$$

де *x* значення пікселя картинки.

3.3.4 Відтінок, насиченість та яскравість

Модель HSV (Hue, Saturation, Value), також відома як HSB (Hue, Saturation, Brightness), - це колірна модель, що використовується для представлення кольорів з точки зору їх сприйняття. На відміну від колірної моделі RGB (Red, Green, Blue), яка базується на адитивному поєднанні основних кольорів, модель HSV розділяє інформацію про колір на три компоненти, які відповідають різним якостям сприйняття та таким чином краще відображає властивості зображення відповідно до зору людини, залишаючись математично вичерпною [22]. Візуальне предаствлення моделі HSV приведено на рисунку 3.11:

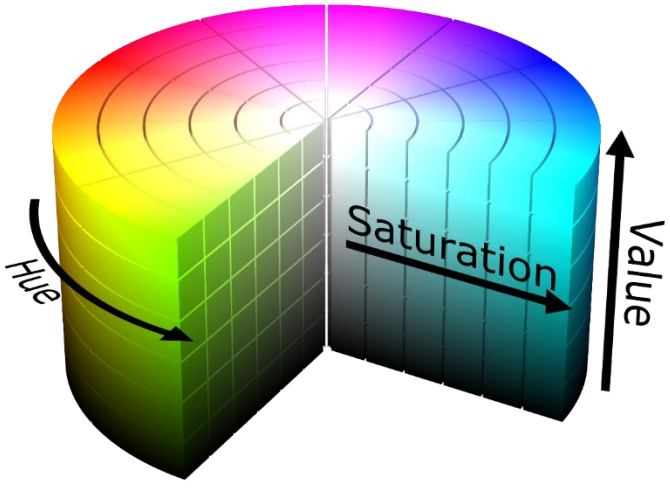

Рисунок 3.11 – Репрезентація моделі HSV

Нижче наведено опис кожного компонента в моделі HSV: [23]

– Відтінок: Компонент відтінку представляє домінуючий колір або позицію кольору на колірному колі. Він описує тип кольору, наприклад, червоний, зелений, синій, жовтий тощо. Значення відтінку коливається від 0 до 360 градусів, де 0 градусів відповідає червоному, 120 градусів - зеленому і 240 градусів - синьому. Відповідне математичне представлення:

$$H = \arccos\left(\frac{\frac{1}{2}(2R-G-B)}{\sqrt{(R-G)^2-(R-B)(G-B)}}\right) \quad (3.3),$$

де R, G та B відповідні значення за моделлю RGB.

– Насиченість: Компонент насиченості визначає інтенсивність або чистоту кольору. Він представляє кількість сірого в кольорі. Значення насиченості 0 вказує на зображення у відтінках сірого, тоді як значення насиченості 1 відображає максимальну інтенсивність кольору. Значення насиченості між 0 і 1 дають різний ступінь інтенсивності кольору. Відповідне математичне представлення:

$$S = \frac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)} \quad (3.4),$$

– Значення (або Яскравість): Компонент значення представляє яскравість або світлість кольору. Він визначає, наскільки яскравим або темним виглядає колір. Значення 0 відповідає чорному кольору, а значення 1 - максимальній яскравості кольору. Відповідне математичне представлення:

$$V = \max(R, G, B) \quad (3.5),$$

У контексті стеганографії за допомогою модель HSV можна краще відслідковувати небажані зміни в зображені які будуть помітними.

4 РОЗРОБКА МЕТОДУ ПОРІВНЯННЯ ЕФЕКТИВНОСТІ СТЕГОКОНТЕЙНЕРІВ ІЗ РІЗНИМИ ХАРАКТЕРИСТИКАМИ

Існує багато метрик що дозволяють оцінювати різні аспекти стегоконтейнерів. Вони є кількісними показниками, що використовуються для оцінки властивостей і характеристик зображення. Ці метрики надають об'єктивні показники, які можна використовувати для оцінки якості, змісту та інших атрибутів обкладинки. Запропонований метод порівняння ефективності використовує різні метрики, із різними сильними та слабкими сторонами. Усі метрики наведено в таблиці 4.1.

Таблиця 4.1 – Список метрик методу

Назва	Інтерпретування	Діапазон значень
Нормальна середня абсолютна різниця	Чим менше тим краще	0 – 1
Середньоквадратична похибка	Чим менше тим краще	Від 0 до безкінечності
Нормована взаємо кореляція	Чим більше тим краще	0 – 1
Якість кореляції	Чим більше тим краще	Від 0 до безкінечності
SSIM	Чим більше тим краще	0 – 1
UIQI	Чим більше тим краще	0 – 1
PSNR	Чим більше тим краще	Від 0 до безкінечності
Різниця за HSV моделлю	Чим менше тим краще	Від 0 до 360 для відтінку, від 0 до 100 для насиченості, від 0 до 100 для яскравості
Різниця за контрастом	Чим менше тим краще	0 – 255

Нижче детально описано усі метрики.

4.1 Різницеві метрики

Різницеві метріки використовуються для кількісної оцінки відмінностей між двома зображеннями. Ці метрики вимірюють рівень спотворення або відмінності між пікселями двох зображень, порівнюючи їхні значення кольору або інтенсивності. Вони широко використовуються в різних програмах обробки зображень, включаючи стеганографію, стиснення зображень і оцінку якості зображень. [3, 24]

Мета різницевих метрик - надати числове значення, яке відображає різницю в сприйнятті або спотворення між двома зображеннями. Ці метрики враховують різні фактори, такі як колірні відмінності, просторові варіації та властивості зорової системи людини, щоб оцінити якість сприйняття зображень. [24]

4.1.1 Нормальна середня абсолютна різниця

Нормальна середня абсолютна різниця (НСАР, Normal Average Absolute Difference, NAAD) – це різницева метрика, яка вимірює середню абсолютну різницю між відповідними значеннями пікселів двох зображень. Вона кількісно визначає загальну різницю в кольорі або інтенсивності між пікселями в нормалізованому вигляді. [24]

НСАР дає змогу виміряти середню абсолютну різницю між відповідними пікселями, вказуючи на рівень розбіжностей у кольорі або інтенсивності між зображеннями. Вище значення НСАР вказує на більшу різницю між зображеннями, тоді як нижче значення НСАР вказує на більшу схожість. [3]

НСАР є загальновживаною метрикою для оцінки якості зображень і може застосовуватися в різних задачах обробки зображень, включаючи стеганографію, стиснення зображень і покращення зображень. Він допомагає оцінити точність або схожість між зображеннями і надає кількісну міру різниці у сприйнятті.

Формула для обчислення нормальної середньої абсолютної різниці між двома зображеннями наступна: [3, 24]

$$NAAD = \frac{\sum_{m,n}|x_{m,n} - \tilde{x}_{m,n}|}{\sum_{m,n}|x_{m,n}|} \quad (4.1),$$

де $x_{m,n}$ – значення відповідного пікселю оригінальної картинки,

$\tilde{x}_{m,n}$ – значення відповідного пікселю стего-зображення картинки.

4.1.2 Середньоквадратична похибка

Середньоквадратична похибка (СП, Mean Square Error, MSE) – це різницева метрика, яка вимірює середню квадратичну різницю між відповідними значеннями пікселів двох зображень. Вона кількісно визначає загальну різницю між зображеннями, враховуючи як величину, так і напрямок різниці. [3]

СП вимірює середню квадратичну різницю між відповідними пікселями, надаючи більшу вагу більшим різницям. Він широко використовується в задачах обробки зображень, таких як стиснення та реконструкція зображень, оскільки допомагає оцінити точність або якість реконструкції зображення. Нижче значення СП вказує на більшу схожість або кращу якість реконструкції між зображеннями, тоді як вище значення СП вказує на більшу різницю. [24, 25]

Важливо зазначити, що СП не відображає безпосередньо людське сприйняття, оскільки враховує квадрати відмінностей, які можуть збільшити вплив викидів або великих помилок. Тому СП часто використовують у поєднанні з іншими метриками для комплексної оцінки якості зображення. [25]

Формула для обчислення середньо квадратичної похибки між двома зображеннями наступна: [3, 24]

$$MSE = \frac{1}{MN} \cdot \sum_{m,n} \left| x_{m,n} - \tilde{x}_{m,n} \right| \quad (4.2),$$

де $x_{m,n}$ – значення відповідного пікселю оригінальної картинки,

$\tilde{x}_{m,n}$ – значення відповідного пікселю стего-зображення картинки,

M – стовпці картинки,

N – рядки картинки.

4.2 Кореляційні метрики

Кореляційні метрики використовуються для вимірювання кореляції або подібності між двома зображеннями. Ці метрики зосереджені на кількісній оцінці зв'язку між значеннями пікселів зображень, часто з урахуванням статистичних властивостей або закономірностей.

Важливо зазначити, що кореляційні метрики спотворення зосереджені на статистичному взаємозв'язку між зображеннями і можуть не відображати зміни сприйняття або візуальної якості. Тому їх варто використовувати у поєднанні з іншими метриками для комплексної оцінки стеганографічних алгоритмів та їхнього впливу на кореляцію зображень.

4.2.1 Нормована взаємо кореляція

Нормована взаємо кореляція (НВК, Normalized Cross Correlation, NCC) - це метрика кореляційних спотворень, яка вимірює схожість між двома зображеннями шляхом обчислення нормованого коефіцієнта крос-кореляції. Він оцінює лінійний зв'язок і подібність між відповідними значеннями пікселів на зображеннях. [3]

У контексті стеганографії НВК можна використовувати для оцінки впливу вбудовування даних на кореляцію між зображеннями обкладинки і стего. Значне зменшення НВК вказує на те, що стеганографічний процес вніс спотворення і змінив схожість між зображеннями. [24, 26]

Формула для обчислення нормованої взаємо кореляції між двома зображеннями наступна: [3, 24]

$$NCC = \frac{\sum_{m,n} x_{m,n} \cdot \tilde{x}_{m,n}}{\sum_{m,n} {x_{m,n}}^2} \quad (4.3),$$

де $x_{m,n}$ – значення відповідного пікселю оригінальної картинки,

$\tilde{x}_{m,n}$ – значення відповідного пікселю стего-зображення картинки.

4.2.2 Якість кореляції

Якість кореляції (ЯК, Correlation Quality, CQ) - це метрика спотворення, яка вимірює якість кореляції між зображенням обкладинки і стего-зображенням у стеганографії. Вона кількісно оцінює зміни в кореляційних характеристиках, що виникають в результаті вбудовування даних. [3]

У стеганографії ЯК можна використовувати для оцінки ефективності стеганографічних алгоритмів у збереженні кореляційних властивостей зображення приховування інформації. Нижчі значення ЯК вказують на вищий рівень спотворення і зниження кореляції, що може свідчити про наявність вбудованих даних. [3]

Формула для обчислення якості кореляції між двома зображеннями наступна: [3, 24]

$$CQ = \frac{\sum_{m,n} x_{m,n} \cdot \tilde{x}_{m,n}}{\sum_{m,n} x_{m,n}} \quad (4.4),$$

де $x_{m,n}$ – значення відповідного пікселю оригінальної картинки,

$\tilde{x}_{m,n}$ – значення відповідного пікселю стего-зображення картинки.

4.3 Метрика PSNR

PSNR (Peak Signal-to-Noise Ratio - пікове співвідношення сигнал/шум) - це об'єктивна метрика вимірювання, яка широко використовується для оцінки якості спотвореного зображення порівняно з еталонним зображенням. Він вимірює співвідношення між максимально можливою потужністю сигналу (в даному випадку еталонного зображення) і потужністю різницевого сигналу (спотворення між еталонним і спотвореним зображеннями). PSNR виражається в децибелах (дБ).

PSNR забезпечує кількісне вимірювання погіршення якості зображення. Вище значення PSNR вказує на нижчий рівень спотворення або вищу якість зображення. І навпаки, нижче значення PSNR свідчить про вищий рівень спотворень або нижчу якість зображення.

Важливо зазначити, що PSNR - це піксельна метрика, яка не враховує важливість різних ділянок зображення для сприйняття. При обчисленні всі пікселі розглядаються однаково, незалежно від їхньої візуальної важливості. Тому PSNR не завжди відповідає людському сприйняттю якості зображення.

Формула для обчислення PSNR між двома зображеннями наступна: [27]

$$PSNR = 10 \log_{10}\left(\frac{L^2}{MSE}\right) \quad (4.6),$$

де MSE – середньоквадратична похибка,

L – рівні квантування за замовчуванням дорівнює 255.

4.4 Метрики SSIM та UIQI

UIQI (Universal Image Quality Index) та SSIM (Structural Similarity Index) - це метрики оцінки якості зображень, які мають на меті виміряти схожість між двома зображеннями. Вони мають певну схожість з точки зору їх призначення та підходу. Обидві метрики враховують особливості людського зорового сприйняття. Вони розроблені так, щоб корелювати з людською оцінкою якості зображення, враховуючи такі фактори, як яскравість, контрастність і структурна схожість. Крім того, обидві метрики враховують структурну інформацію зображень. Вони оцінюють, наскільки добре збережені просторові структури, такі як краї та текстури, між еталонним і спотвореним зображеннями.

4.4.2 Метрика UIQI

UIQI (Universal Image Quality Index) - це метрика оцінки якості зображення, яка вимірює схожість між еталонним і спотвореним зображенням. Він враховує різні властивості зображень, включаючи яскравість, контрастність і структурну схожість. Вище значення UIQI вказує на більшу схожість між зображеннями, тоді як нижче значення вказує на більш суттєві відмінності. [27]

UIQI широко використовується в дослідженнях і додатках для оцінки якості зображень. Однак варто зазначити, що в різних контекстах і дослідницьких роботах можуть бути варіації у застосуванні та конкретних розрахунках UIQI. [3, 27]

Формула для обчислення UIQI між двома зображеннями наступна: [27]

$$UIQI = \frac{4\sigma_{I\tilde{I}}\mu_I\mu_{\tilde{I}}}{(\mu_I{}^2+\mu_{\tilde{I}}{}^2)(\sigma_I{}^2+\sigma_{\tilde{I}}{}^2)} \quad (4.6),$$

де σ_I – стандартне відхилення оригінального зображення,

$\sigma_{\tilde{I}}$ – стандартне відхилення стего-зображення,

$\sigma_{I\tilde{I}}$ – взаємна коваріація,

μ_I – середнє значення оригінального зображення,

$\mu_{\tilde{I}}$ – середнє значення стего-зображення.

4.4.1 Метрика SSIM

SSIM (Structural Similarity Index) - це метрика оцінки якості зображення, яка вимірює структурну схожість між еталонним і спотвореним зображеннями. Він

ґрунтується на концепції UIQI (Універсальний індекс якості зображення) і вводить додаткові елементи для врахування структурної інформації. [26]

SSIM враховує яскравість, контрастність і структурну схожість між зображеннями. Він враховує як середнє, так і стандартне відхилення інтенсивностей пікселів, а також перехресну коваріацію. Включаючи ці елементи, SSIM забезпечує більш комплексну оцінку якості зображення, особливо з точки зору збереження структурної інформації. [26]

SSIM стала широко прийнятою метрикою якості зображень завдяки своїй здатності фіксувати як глобальну, так і локальну структурну подібність. Він широко використовується для стиснення зображень і відео, реставрації зображень та інших застосувань, де збереження структурних деталей є важливим. [3, 26]

Формула для обчислення SSIM між двома зображеннями наступна: [26]

$$SSIM = \frac{(2\mu_I\mu_{\tilde{I}}+c_1)(2\sigma_{I\tilde{I}}+c_2)}{(\mu_I{}^2+\mu_{\tilde{I}}{}^2+c_1)(\sigma_I{}^2+\sigma_{\tilde{I}}{}^2+c_2)} \quad (4.7),$$

де σ_I – стандартне відхилення оригінального зображення,

$\sigma_{\tilde{I}}$ – стандартне відхилення стего-зображення,

$\sigma_{I\tilde{I}}$ – взаємна коваріація,

μ_I – середнє значення оригінального зображення,

$\mu_{\tilde{I}}$ – середнє значення стего-зображення,

c_1– константа один, дорівнює $c_1 = (K_1 \cdot L)^2$, де за замовчуванням $K_1 = 0{,}01$,

c_2– константа два, дорівнює $c_2 = (K_2 \cdot L)^2$, де за замовчуванням $K_2 = 0{,}03$,

L – рівні квантування за замовчуванням дорівнює 255.

4.5 Різниця за HSV моделю

На основі проведеного можливо використати порівняння властивостей зображення за HSV моделю. Вимірюючи абсолючну різницю в кожному компоненті моделі HVS, можливо отримати уявлення про те, як відтінок, насиченість і значення відрізняються між двома зображеннями. Цей підхід

дозволяє більш детально проаналізувати варіації кольорів і може бути корисним у сценаріях, де важливим є розуміння конкретного внеску кожного компонента.

Так наприклад чорно-білі зображення завжди мають нульове значення насиченості, таким чином навіть дуже незначні зміни насиченості буде дуже легко аналітично виявити у чорно-білому зображені.

4.6 Різниця за контрастом

На основі проведеного аналізу пропонується застосувати метрику яка має на меті зафіксувати абсолютну різницю контрасту між двома зображеннями, тобто абсолютну різниця контрасту (АРК, Absolute Contrast Difference, ACD). ACD вимірює розбіжність у контрасті між еталонним і спотвореним зображеннями.

АРК вимірює абсолютну різницю в значеннях контрасту між еталонним і спотвореним зображеннями. Вище значення АРК вказує на більшу розбіжність у контрастності, що свідчить про більший рівень спотворення з точки зору контрастності. І навпаки, нижче значення АРК вказує на меншу різницю контрасту і потенційно менш помітне спотворення.

АРК особливо корисна у випадках, коли контраст відіграє важливу роль у візуальному сприйнятті або оцінці якості зображення, тобто в зображеннях з низькою контрасністю. Він може допомогти оцінити точність методів покращення зображення або маніпуляцій, які впливають на характеристики контрасту.

5 РОЗРОБКА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДЛЯ ЗАПРОПОНОВАНОГО МЕТОДА

Застосовуючи метод порівняння за допомогою метрик що було розглядено у розділи чотири було розроблено програмний застосунок на мові програмування Java. Мову Java було обрано з декількох причин. [28]

- *Продуктивність та ефективність:* Хоча Java часто критикують за її низьку продуктивність порівняно з мовами низького рівня, вона все ще пропонує достатню продуктивність для багатьох додатків, включаючи аналіз та обробку зображень. Завдяки відповідним алгоритмам та оптимізаціям, Java може ефективно впоратися з обчислювальними вимогами стеганографічного додатку для оцінки зображень обкладинок.
- *Об'єктно-орієнтований підхід:* Парадигма об'єктно-орієнтованого програмування (ООП) Java добре підходить для створення складних і модульних додатків. Вона сприяє організації коду, повторному використанню та підтримці. Використовуючи принципи ООП, додаток можна структурувати на модульні компоненти, що полегшує його розуміння, підтримку та вдосконалення в майбутньому.
- *Велика спільнота розробників:* Java має велике та активне співтовариство розробників, що означає, що доступна велика кількість документації, ресурсів і підтримки спільноти. Розробники можуть скористатися знаннями та досвідом, якими ділиться спільнота, ефективніше вирішувати проблеми та мати доступ до безлічі навчальних матеріалів і посібників.

До того ж були застосовані бібліотеки з відритим вихідним кодом.

- Apache Commons – це бібліотека з колекцією багаторазових компонентів Java, які забезпечують реалізацію загальних утилітарних функцій і структур даних. Він пропонує широкий спектр утилітарних

класів та допоміжних методів, які спрощують типові завдання програмування та підвищують продуктивність розробників. Деякі з ключових можливостей Apache Commons включають маніпуляції з рядками, операції вводу/виводу, математичні функції, утиліти для роботи з датою і часом та багато іншого. [29]

- У данній програмі використовувалися класи і функцій утиліт Apache Commons, які спрощують такі завдання, як маніпуляції з файлами, обробка рядків і утилітні операції загального призначення.

- Apache POI (Poor Obfuscation Implementation) - це бібліотека Java для читання і запису файлів форматів Microsoft Office, включаючи таблиці Excel (.xls, .xlsx) і документи Word (.doc, .docx). Вона надає набір інструментів, які дозволяють розробникам програмно маніпулювати та витягувати дані з цих форматів файлів. Apache POI підтримує різні функції, такі як створення нових документів, модифікація існуючих, читання значень комірок, форматування комірок і виконання інших операцій, характерних для файлів Microsoft Office. [30]
  - У данній програмі використовувалися класи і функцій утиліт Apache POI, для роботи з файлами XLS та зручного відображення результатів оцінювання в табличному форматі, що полегшує аналіз і порівняння результатів кількох оцінюваних зображень.

Програма має на меті допомогти користувачам у виборі відповідних зображень для приховування інформації за допомогою стеганографічних методів та виявленню непридатних для цих цілей зображень. Вона забезпечує аналіз зображень-контейнерів для оцінки їхнього потенціалу для непомітного приховування інформації.

5.1 Опис програми

Програмний застосунок складається з п’яти компонентів, які відіграють певну роль у функціональності програми. Вихідний код основний компоненитів наведений у додатку Б. Усі компоненти наведені в діаграмі компонентів приведеній на рисунку 5.1.

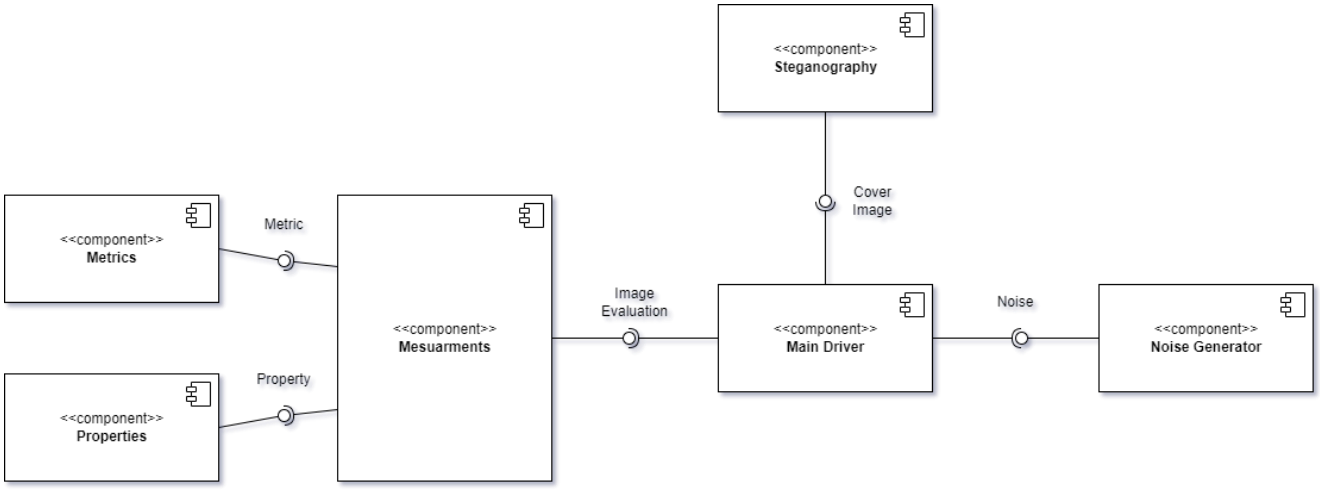

Рисунок 5.1 – Діаграма компонентів програми.

Цими компонентами є:

- *«Main Driver»*: слугує точкою входу в програму. Він керує загальним потоком виконання і координує взаємодію між різними компонентами. Він ініціалізує необхідні об'єкти, викликати відповідні методи та керувати загальною логікою програми.
- *«Measurements»*: відповідає за виконання різноманітних вимірювань та аналізу зображень обкладинок та представлення результатів. Взаємодії з компонентами «Metrics» та «Properties» для отримання розрахунків.
- *«Metrics»*: включає різні метрики якості, що використовуються для оцінки впливу стеганографії на зображення контейнеру. Він включає реалізацію таких метрик, як SSIM, PSNR, UIQI або інших відповідних метрик.
- *«Properties»*: відповідає за розрахунок та вилучення властивостей контейнеру, таких як контраст та значення за моделлю HSV.

- *Steganography*: реалізує стеганографічні методи, що використовуються для приховування інформації у контейнерах. Він включає алгоритми для вбудовування даних, вилучення прихованої інформації.
- *Noise Generator*: Компонент Генератор шуму генерує різні типи шуму, такі як гаусівський шум або фрактальний шум. Він надає методи створення шумових патернів, які використовуються для генерації зображень. Цей компонент включає алгоритми для генерування шумових патернів з потрібними характеристиками.

Повна структура програми наведена у вигляді діаграми класів на рисунку 5.2. Далі описано кожний компонент та його складові окремо у відповідному розділі.

5.1.1 Компонент «Main Driver»

Структура компонент «Main Driver» наведена у вигляді діаграми класів на рисунку 5.3.

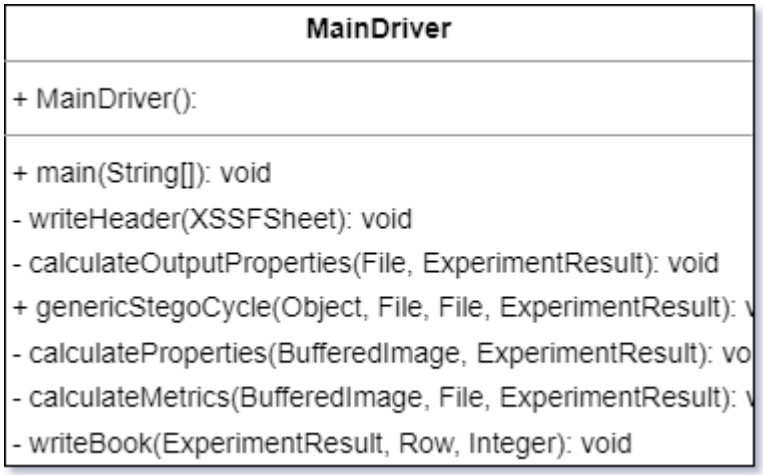

Рисунок 5.3 – Діаграма класів компоненту «Main Driver»

Компонент «Main Driver» у містить клас MainDriver. Цей клас містить наступні методи:

- main(String[]): void : слугує точкою входу в програму.
- genericStegoCycle(Object, File, File, ExperimentResult): void : представляє собою загальний стеганографічний цикл. Він приймає вхідні параметри, такі як стеганографічний алгоритм, вхідний файл зображення обкладинки, вихідний файл стего-зображення та об'єкт

ExperimentResult. Він виконує процес стеганографічного вбудовування, оновлюючи об'єкт ExperimentResult результатом процесу (Успіх, Частковий успіх, Помилка).

- calculateProperties(BufferedImage, ExperimentResult): void : обчислює властивості зображення що знаходиться у BufferedImage (в данному випадку це контейнера) і оновлює об'єкт ExperimentResult обчисленими значеннями.
- calculateOutputProperties(File, ExperimentResult): void : обчислює властивості зображення що знаходиться у File (в данному випадку це стего-зображення) і оновлює об'єкт ExperimentResult обчисленими значеннями.
- calculateMetrics(BufferedImage, File, ExperimentResult): void : обчислює метрики між стего-зображенням (File) та оригінальним зображеннмя обкладинки (BufferedImage). Обчислені метрики зберігаються в об'єкті ExperimentResult.
- writeHeader(XSSFSheet): void : записує назви стовпців у вказаний XSSFSheet в електронній таблиці Excel.
- writeBook(ExperimentResult, Row, Integer): void : записує результати експерименту, що зберігаються в об'єкті ExperimentResult, до вказаного рядка Row у таблицю Excel.

5.1.2 Компонент «Steganography»

Структура компонент «Steganography» наведена у вигляді діаграми класів на рисунку 5.4.

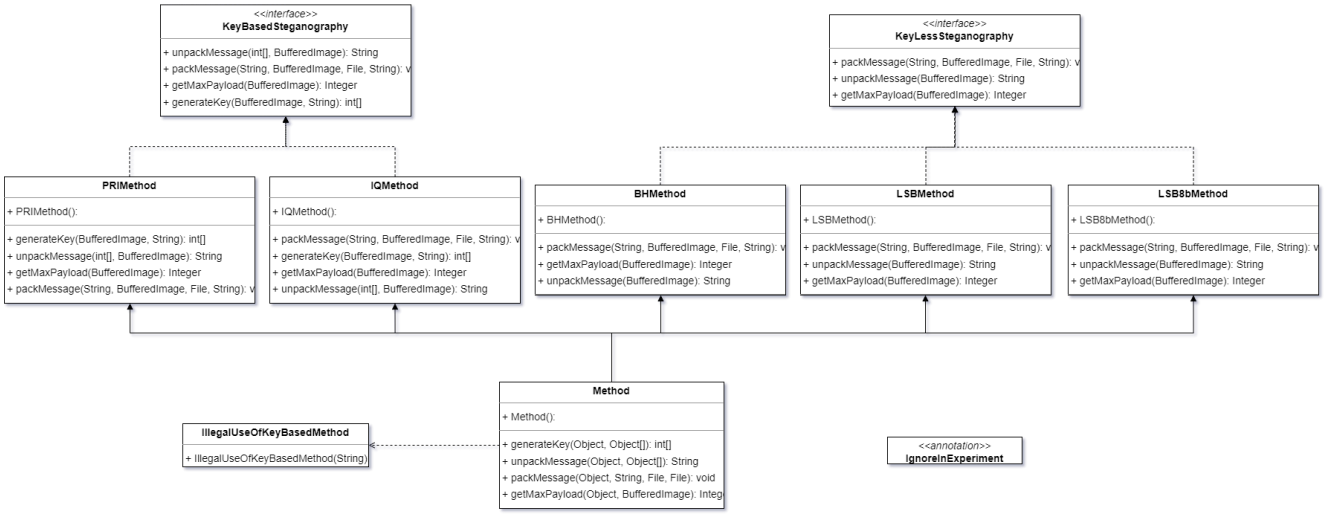

Рисунок 5.4 – Діаграма класів компоненту «Steganography»

Компонент «Steganography» складається з декількох інтерфейсів, абстрактних класів, класів, виключень та анотацій. Нижче наведено огляд кожного елемента компонента «Steganography»:

Інтерфейси:

- SteganographyMethod: визначає базову поведінку для методу стеганографії. Він визначає методи, які повинен реалізовувати метод стеганографії. Цей інтерфейс містить наступні методи:
    - getMaxPayload(Object, BufferedImage): Integer : обчислює і повертає максимальний розмір корисного навантаження, яке можна вбудувати в задане зображення обкладинки, використовуючи вказаний метод стеганографії.
    - packMessage(Object, String, File, File): void : вбудовує задане повідомлення у файл зображення обкладинки і створює файл стего-зображення, використовуючи вказаний метод стеганографії. Повідомлення надається у вигляді рядка.
    - unpackMessage(Object, Object[]): String : витягує і повертає приховане повідомлення зі стего-зображення, використовуючи надані параметри.
    - generateKey(Object, Object[]): int[] : генерує ключ на основі наданих об'єктів і стего-алгоритму. Ключ повертається як масив цілих чисел.

- KeyBasedSteganography: визначає поведінку для методів стеганографії, які вимагають ключ для вбудовування та вилучення прихованої інформації.
- KeyLessSteganography: визначає поведінку для методів стеганографії, які не потребують ключа для вбудовування та вилучення прихованої інформації.

Абстрактні класи:

- Method: реалізує інтерфейс SteganographyMethod. Він забезпечує базову реалізацію загальної функціональності, необхідної для методів стеганографії. Конкретні методи стеганографії розширюють цей клас для реалізації специфічних алгоритмів вбудовування та вилучення.

Класи:

- LSBMethod: розширює абстрактний клас Method і реалізує інтерфейс KeyLessSteganography. Він представляє метод стеганографії, заснований на методі найменшого значущого біта (LSB) для вбудовування та вилучення прихованої інформації.
- LSB8bMethod: розширює абстрактний клас Method і реалізує інтерфейс KeyLessSteganography. Він представляє метод стеганографії, заснований на техніці LSB, але використовує найстарший біт кожного пікселя для вбудовування та вилучення прихованої інформації.
- BHMethod: розширює абстрактний клас Method і реалізує інтерфейс KeyLessSteganography. Він представляє метод стеганографії, заснований на техніці блокового приховування для вбудовування та вилучення прихованої інформації.
- PRIMethod: розширює абстрактний клас Method і реалізує інтерфейс KeyBasedSteganography. Він представляє метод стеганографії, заснований на техніці псевдо-випадкового інтервалу, яка вимагає ключа для вбудовування та вилучення прихованої інформації.
- IQMethod: розширює абстрактний клас Method і реалізує інтерфейс KeyBasedSteganography. Він представляє метод стеганографії,

заснований на методі квантування зображення, який потребує ключа для вбудовування та вилучення прихованої інформації.

Виключення:

- IllegalUseOfKeyBasedMethod: генерується, коли метод стеганографії на основі ключів використовується неналежним чином, наприклад, при спробі вбудувати або витягти приховану інформацію з застосуванням ключа неналежного формату.

Анотації:

- IgnoreInExperiment: використовується для позначення певних класів, які слід ігнорувати або виключити з процесу експериментального оцінювання.

5.1.3 Компонент «Measurements»

Структура компонент «Measurements» наведена у вигляді діаграми класів на рисунку 5.5.

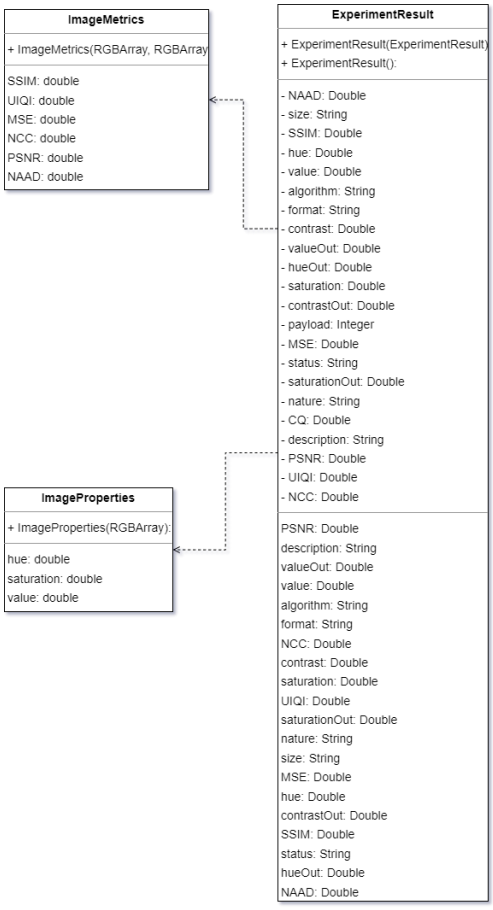

Рисунок 5.5 – Діаграма класів компоненту «Measurements»

Компонент «Measurements» складається з декількох класів. Нижче наведено огляд кожного класа компонента «Measurements»:

- ExperimentResult: представляє результат експерименту оцінки для однієї пари обкладинки та стего-зображення. Він зберігає обчислені властивості зображення, метрики та іншу релевантну інформацію, як то опис.
- ImageMetrics: інкапсулює обчислення та вимірювання різних метрик зображення. Він звертається до статичних методів відповідних класів для обчислення метрик.
- ImageProperties: інкапсулює обчислення та вимірювання різних властивостей зображення. Він звертається до статичних методів відповідних класів для обчислення властивостей.

5.1.4 Компонент «Metrics»

Структура компонент «Metrics» наведена у вигляді діаграми класів на рисунку 5.6.

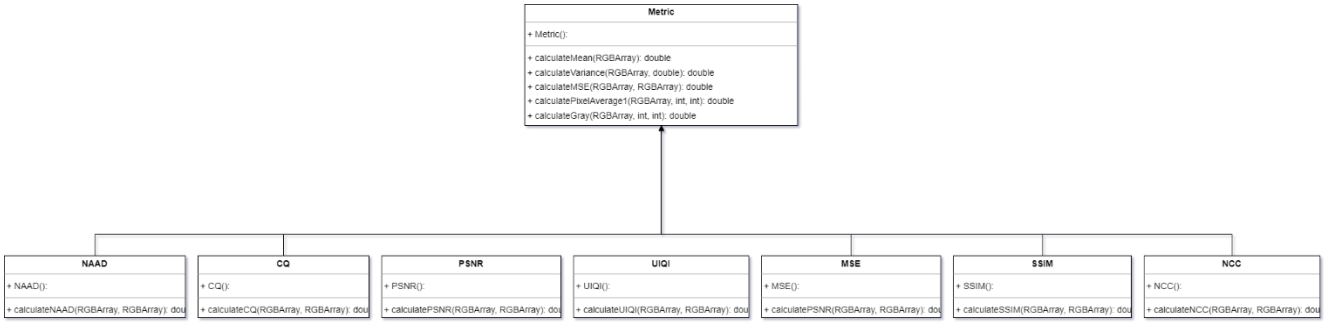

Рисунок 5.6 – Діаграма класів компоненту «Metrics»

Компонент «Metrics» складається з декількох абстрактних класі та класів. Нижче наведено огляд кожного елемента компонента «Metrics»:

Абстрактні класи:

- Metric: надає базову реалізацію для метрик оцінювання зображень, включаючи допоміжні методи для виконання необхідних обчислень. Конкретні класи метрик розширюватимуть цей абстрактний клас і надаватимуть конкретні реалізації обчислень метрик.

Класи:

- NAAD: розширює абстрактний клас Metric і вимірює нормальну середню абсолютну різницю між відповідними пікселями на обкладинці та стего-зображенні для оцінки рівня спотворення.
- MSE: розширює абстрактний клас Metric і вимірює середньоквадратичну похибку.
- NCC: розширює абстрактний клас Metric і вимірює нормовану взаємо кореляцію.
- CQ: розширює абстрактний клас Metric і вимірює якість кореляції.
- PSNR: розширює абстрактний клас Metric і вимірює пікове співвідношення сигнал/шум.
- SSIM: розширює абстрактний клас Metric і вимірює індекс структурної подібності.
- UIQI: розширює абстрактний клас Metric і вимірює універсальний індекс якості зображення.

5.1.5 Компонент «Properties»

Структура компонент «Properties» наведена у вигляді діаграми класів на рисунку 5.7.

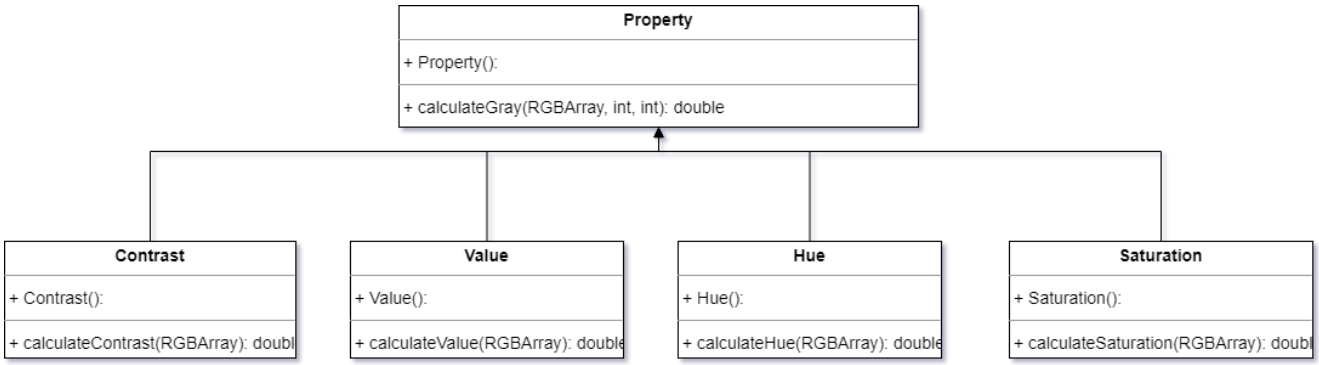

Рисунок 5.7 – Діаграма класів компоненту «Properties»

Компонент «Properties» складається з декількох абстрактних класі та класів. Нижче наведено огляд кожного елемента компонента «Properties»:

Абстрактні класи:

- Property: надає базову реалізацію для оцінювання властивосцей зображень зображень, включаючи допоміжні методи для виконання необхідних обчислень. Конкретні класи властивостей

розширюватимуть цей абстрактний клас і надаватимуть конкретні реалізації обчислень властивостей.

Класи:

- Hue: розширює абстрактний клас Property і вимірює відтінок зображення.
- Saturation: розширює абстрактний клас Property і вимірює насиченість зображення.
- Value: розширює абстрактний клас Property і вимірює яскравість зображення.
- Contrast: розширює абстрактний клас Property і вимірює контрастність зображення.

5.1.6 Компонент «Noise Generator»

Структура компонент «Main Driver» наведена у вигляді діаграми класів на рисунку 5.8.

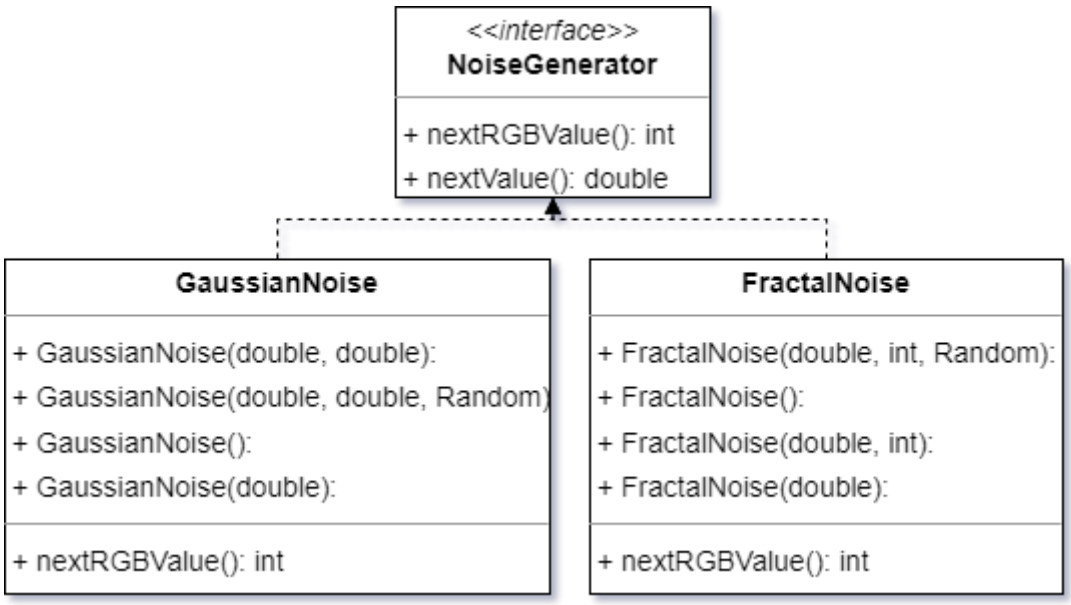

Рисунок 5.8 – Діаграма класів компоненту «Noise Generator»

Компонент «Noise Generator» складається з декількох інтерфейсів та класів, виключень. Нижче наведено огляд кожного елемента компонента «Noise Generator»:

Інтерфейси:

- NoiseGenerator: визначає базову поведінку для генератора шуму. Він визначає методи, які повинен реалізовувати генератор шуму. Цей інтерфейс містить наступні методи:
    - nextValue(): double: генерує наступне значення шуму як число з плаваючою комою.
    - nextRGBValue(): int: метод генерує наступне значення шуму як ціле число у колірному просторі RGB.

Класи:

- GaussianNoise: реалізує інтерфейс NoiseGenerator і представляє генерацію гаусівського шуму.
- FractalNoise: реалізує інтерфейс NoiseGenerator і представляє генерацію фрактального шуму.

5.1.7 Утилітні класи

Усі утилітні класи наведеніна рисунку 5.9.

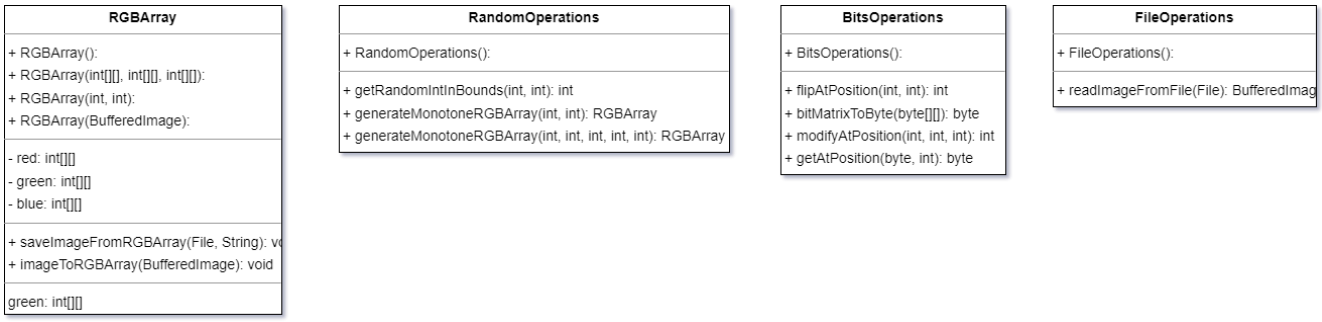

Рисунок 5.9 – Утилітні класи

Програма включає декілька утилітних класів, які надають різні допоміжні функції та операції.

5.2 Результати роботи програми

Практично метод було протестовано на 60 зображеннях (по 12 за кожним типом вмісту), список наведено у таблиці у додатку В, із застосуванням кожного з чотирьох застосованих стеганографічних методів. Ця комплексна оцінка налічує таким чином 300 експериментів, список наведено у таблиці у додатку Г, що

дозволило оцінити продуктивність та ефективність контейнери-зображень із різними характеристиками. Проведено ретельний аналіз і порівняння результатів, розглянутих в контексті різних властивостей зображень і характеристик. Надалі наведено результати.

5.2.1 Різницеві метрики

Згідно результатам приведеним на рисунку 5.10:

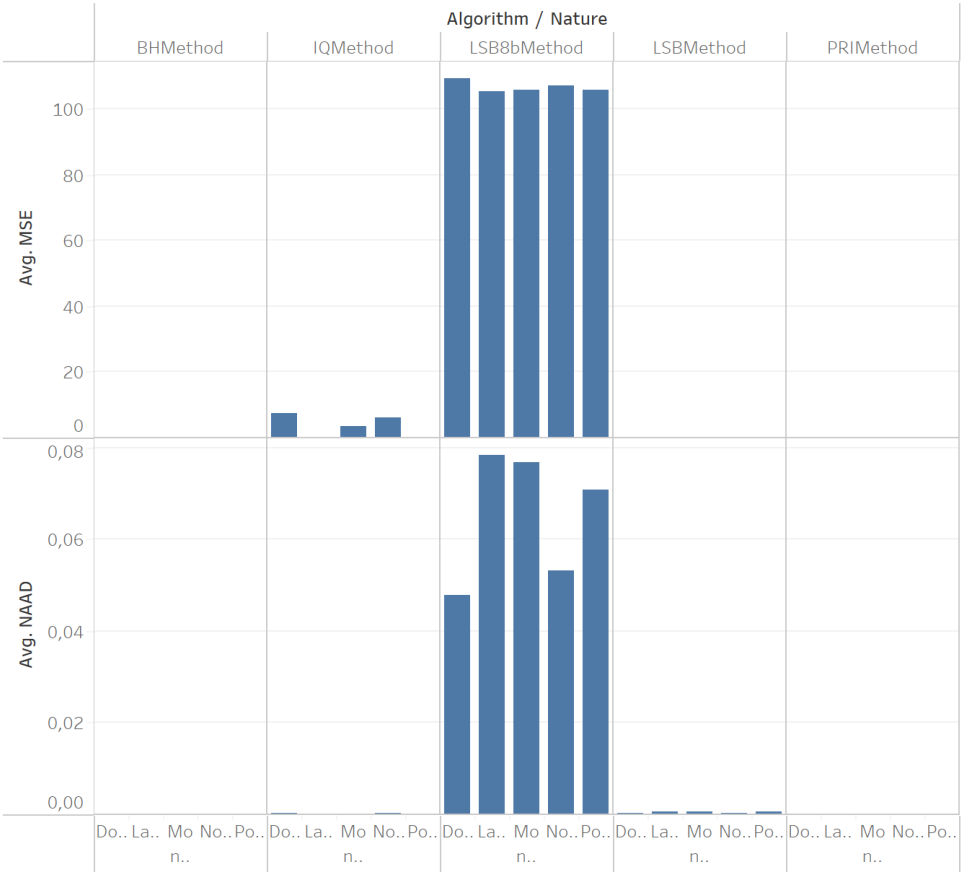

Рисунок 5.10 – Загальна гістограма результатів різницевих метрик

– MSE достатньо чутлива щоб виявити спотворення створене навмисно помітним методом (LSB8b – запис у 8й біт), та також здатна виявляти аномальні значення через особливості роботи метода. Так в силу реалізації метод не пристосований для роботи з занадто низькою або високою контрастністю (тобто програють якщо вмістом є документ або шум, тому що вони часто мають сусідні пікселі з дуже великою різницею в значеннях).
– NAAD здатна виявляти помітні спотворення, але доволі обмежено.

Загалом різницеві метрики придатні для автоматичного виявлення стего-зображень з високим рівнем спотворення. Детально результати для кожного зображення наведені на рисунку 5.11.

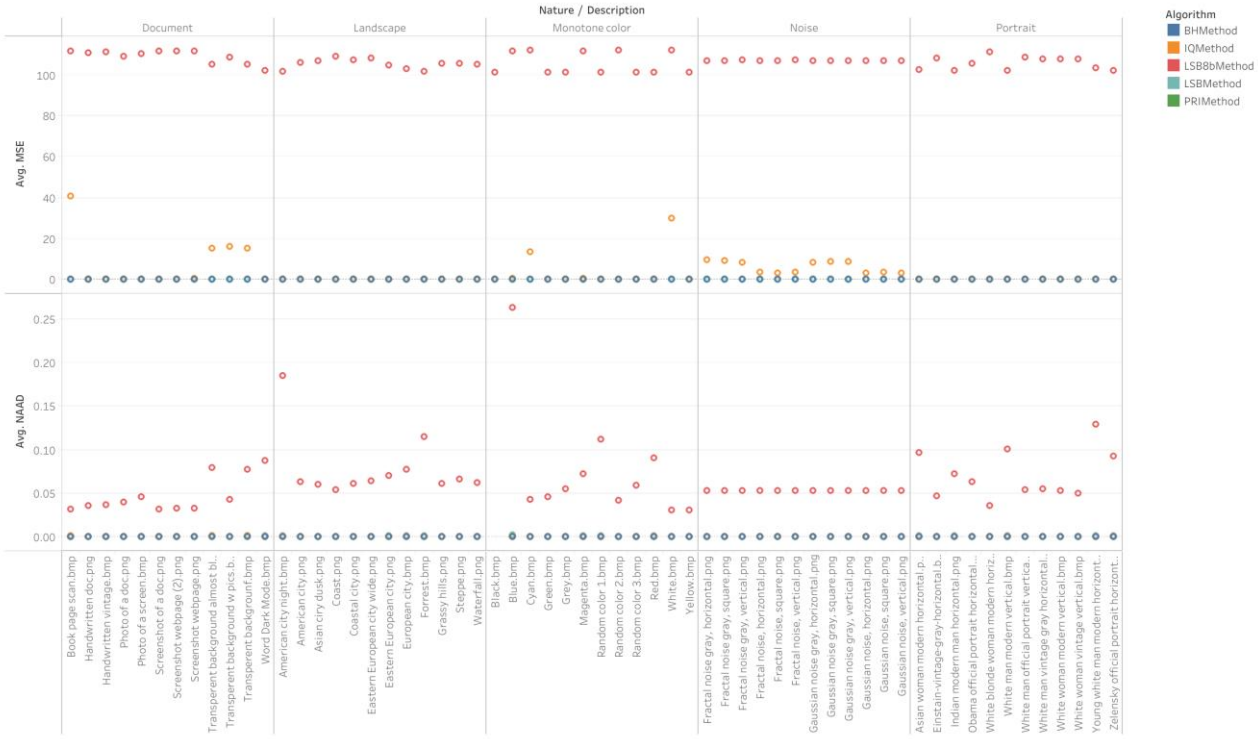

Рисунок 5.11 – Різницеві метрики кожного зображення

5.2.2 Кореляційні метрики

Згідно результатам приведеним на рисунку 5.12:

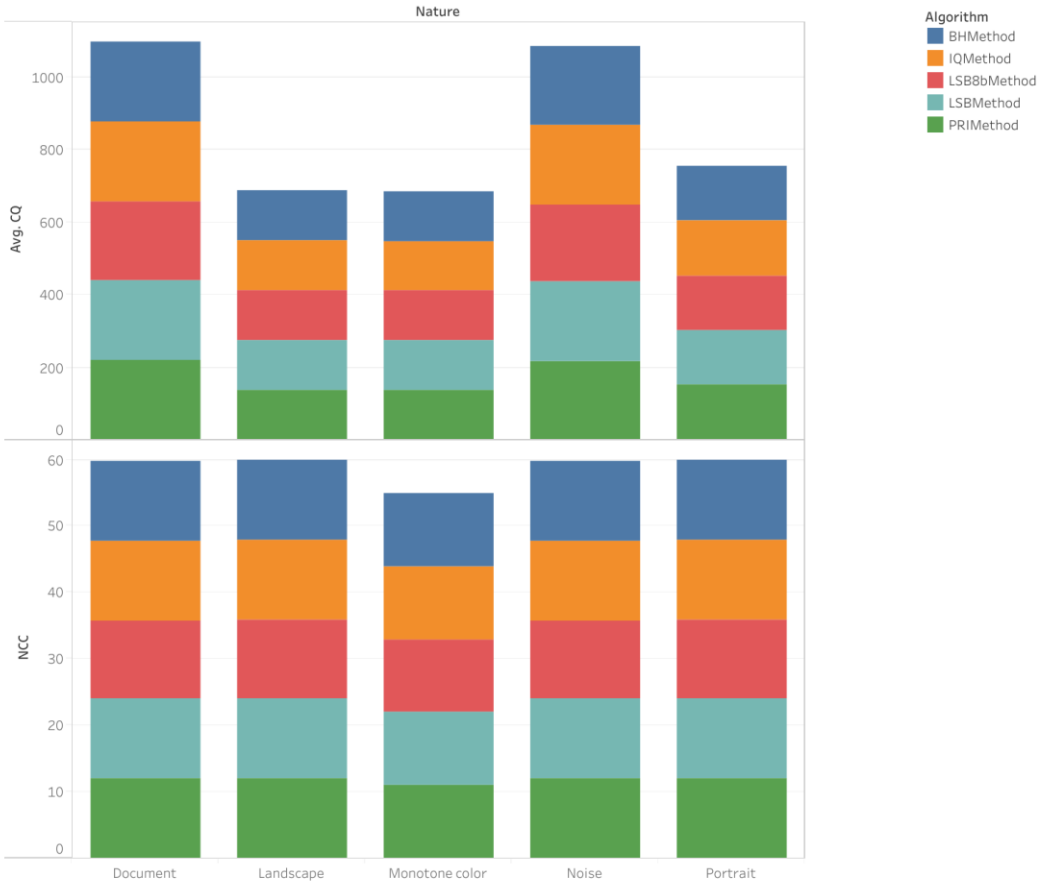

Рисунок 5.12 – Загальна гістограма результатів кореляційних метрик

– Застосування NCC нажаль не виявилося ефективним. Метрика не виявила зображення що мали навіть дуже помітні спотворення, за виключенням суто чорного зображення.
– CQ також на дала переконливих результатів, значення метрики коливається без відповідно до ступіню спотворення.

Таким чином кореляційні метрики не придатні для застосування у методі оцінки ефективності стеганографічних контейнерів-зображень. Детально результати для кожного зображення наведені на рисунку 5.13.

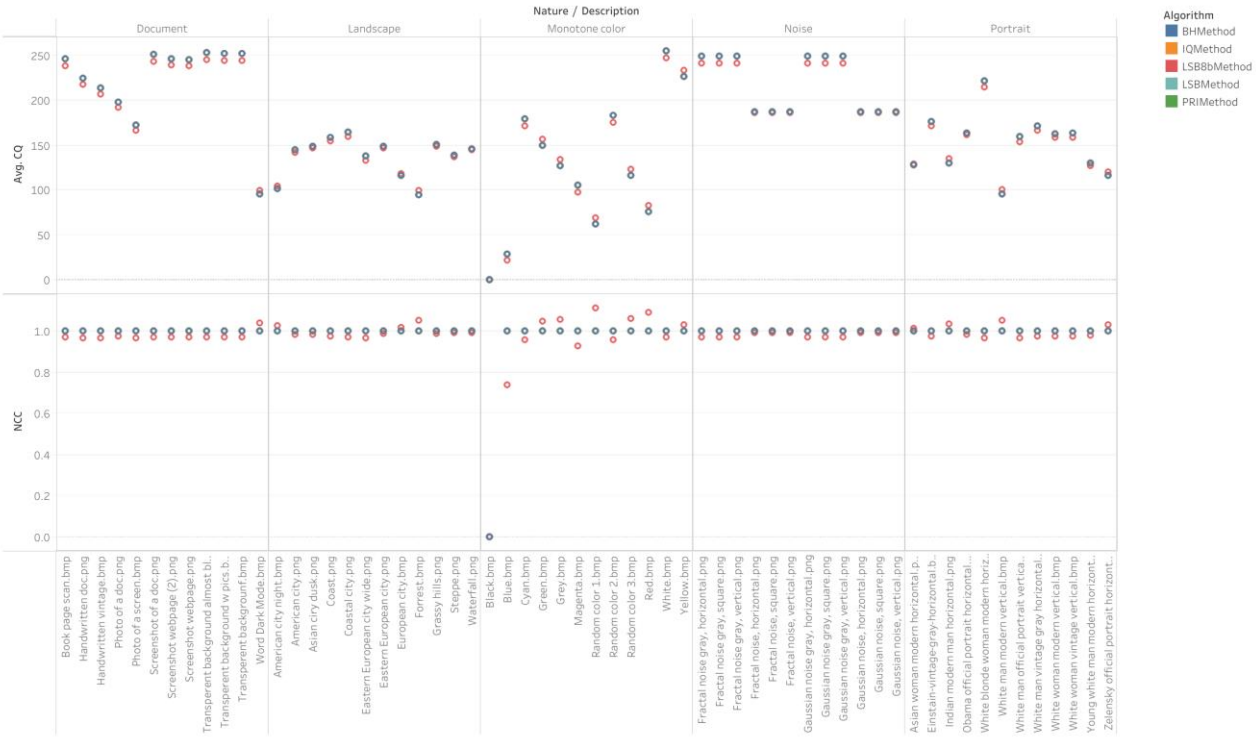

Рисунок 5.13 – Кореляііційні метрики кожного зображення

5.2.3 Метрики SSIM та UIQI

Згідно результатам приведеним на рисунку 5.14:

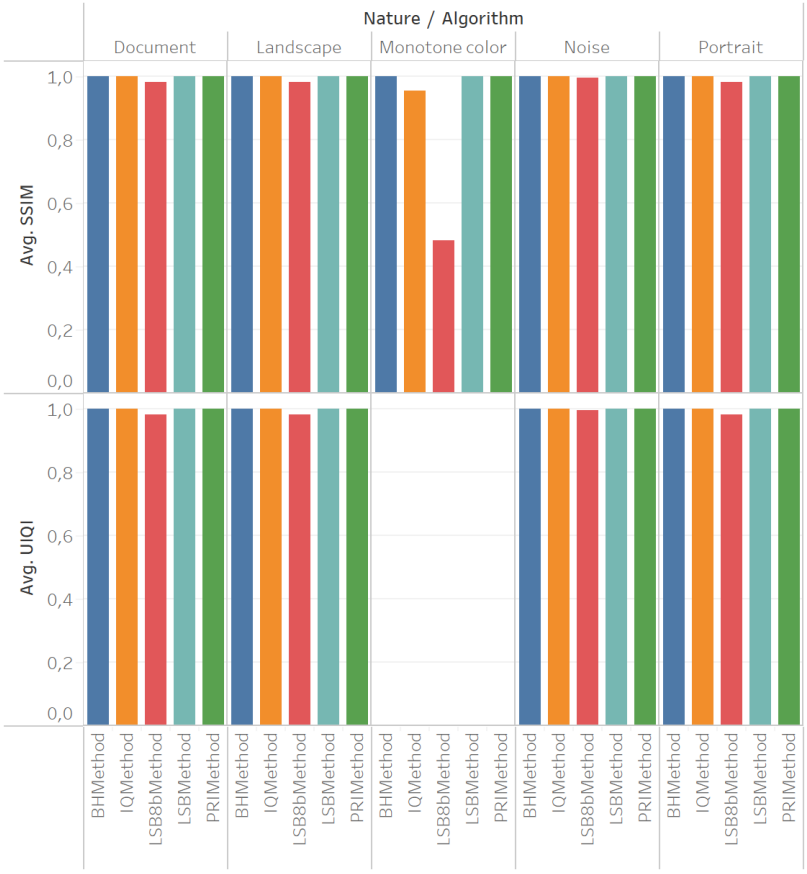

Рисунок 5.14 – Загальна гістограма результатів SSIM та UIQI

– SSIM мала достатньо виражену реакцію на зміну до структури зображення, хоча вона і має проблеми з виявленням спотворень особливо у шумі так в нього відсутня чітка структура.
– UIQI у свою чергу майже повторює результати SSIM але більш чутлива навіть до незначних змін до зображення з чіткою стуктурою якто монотонні зображення.

Загалом ці метрики дозволяють автоматично виявляти спотворення до картинок з виразною структурою і добре доповнюють минулі метрики. Детально результати для кожного зображення наведені на рисунку 5.15.

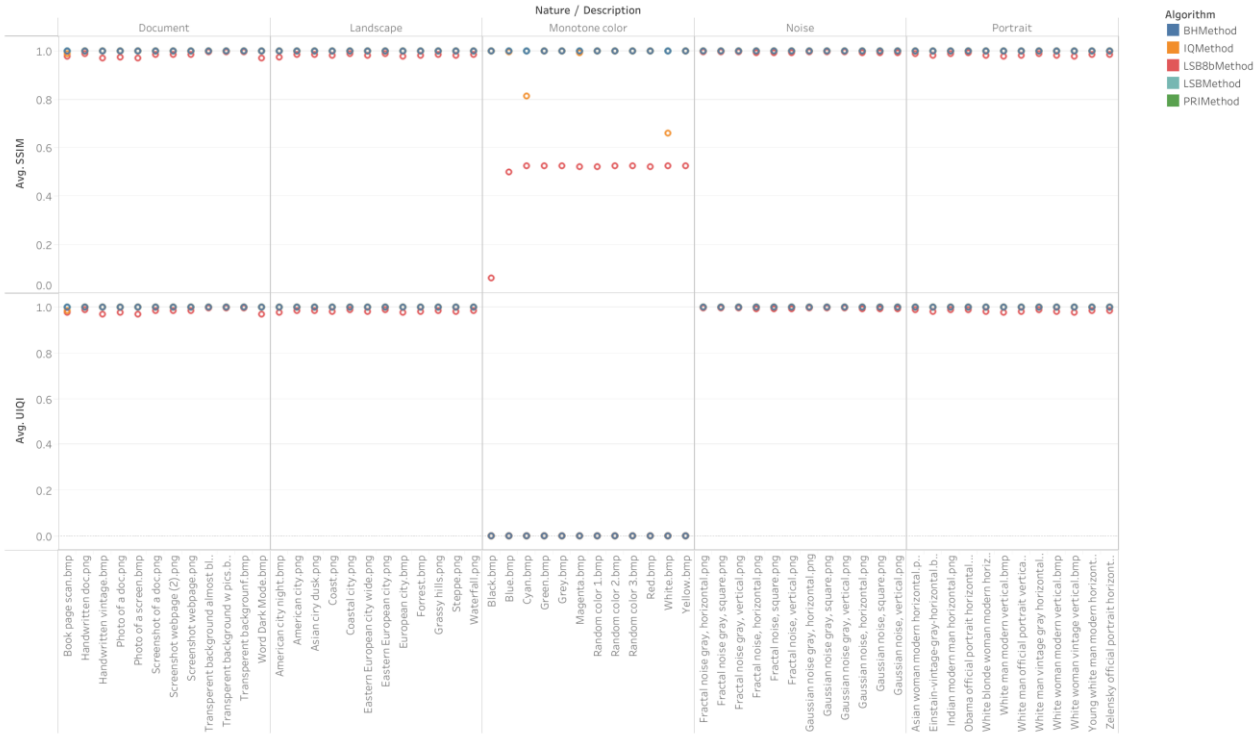

Рисунок 5.15 – SSIM та UIQI кожного зображення

5.2.4 Метрика PSNR

Згідно результатам приведеним на рисунку 5.16:

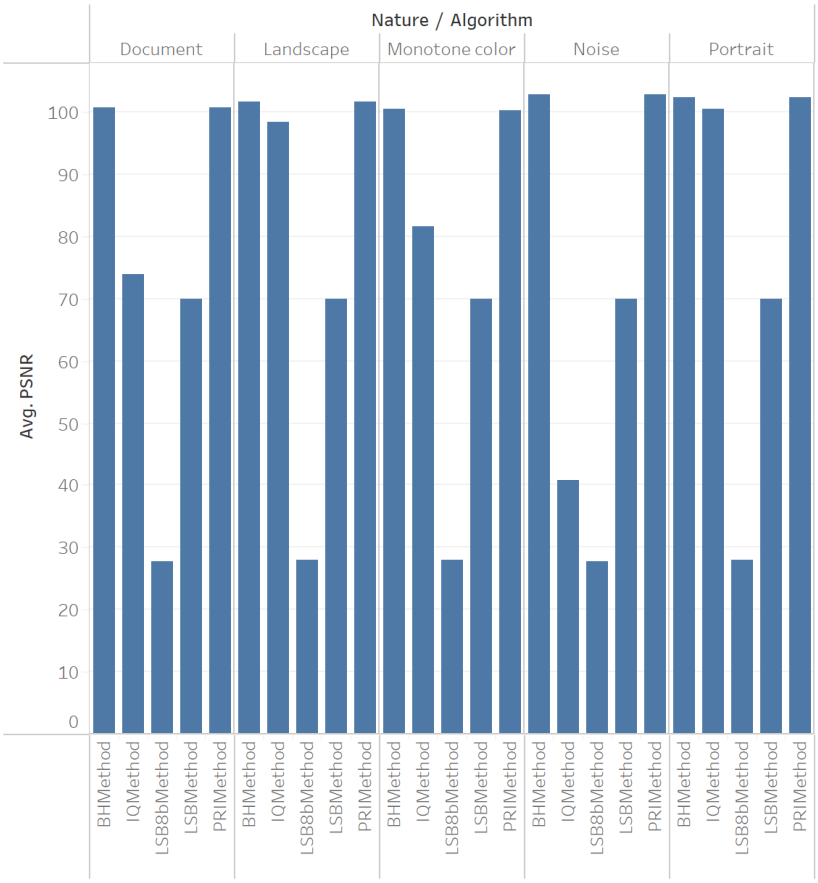

Рисунок 5.16 – Загальна гістограма результатів PSNR

PRSN дозволяє впевнено оцінити рівень спотворення внесений в зображення як і різницеві метрики, але є більш наочний завдяки представленні результата в дБ. Ця метрика доповнює метрики SSIM та UIQI. Детально результати для кожного зображення наведені на рисунку 5.17.

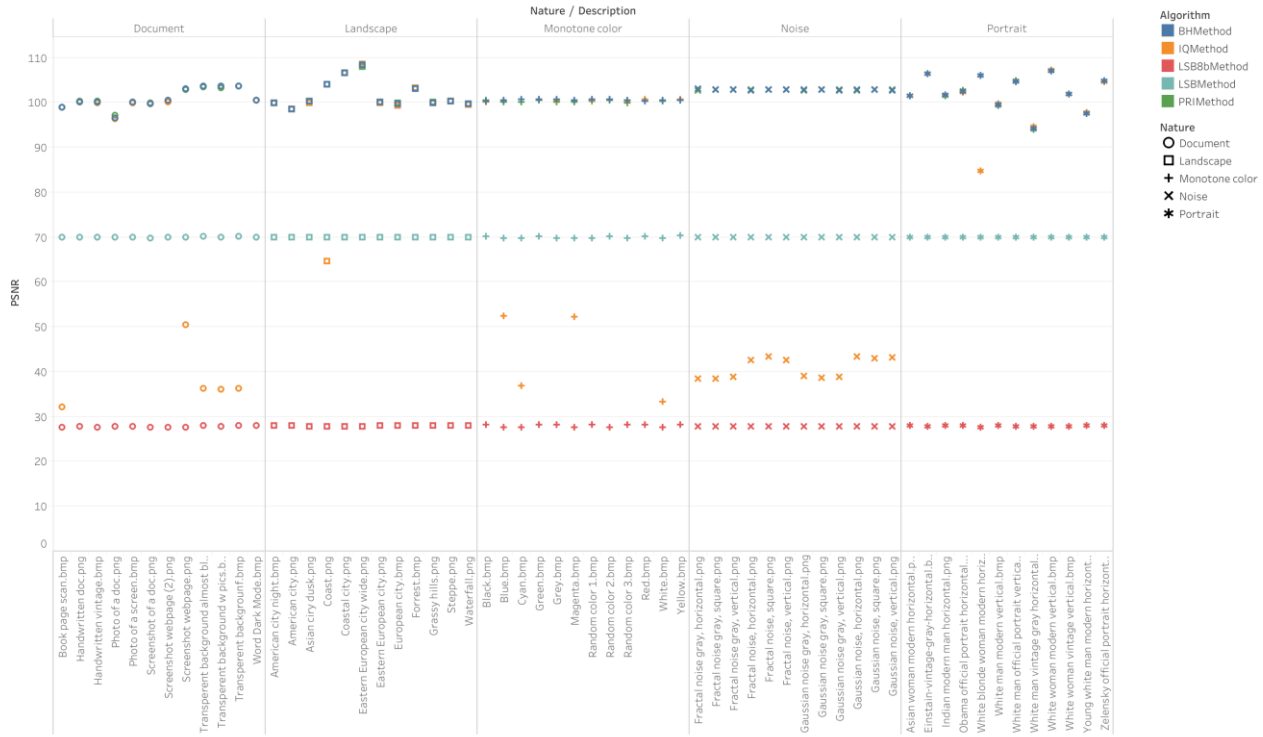

Рисунок 5.17 – PSNR кожного зображення

5.2.5 Абсолютна різниця за HSV моделлю

Згідно результатам приведеним на рисунку 5.18:

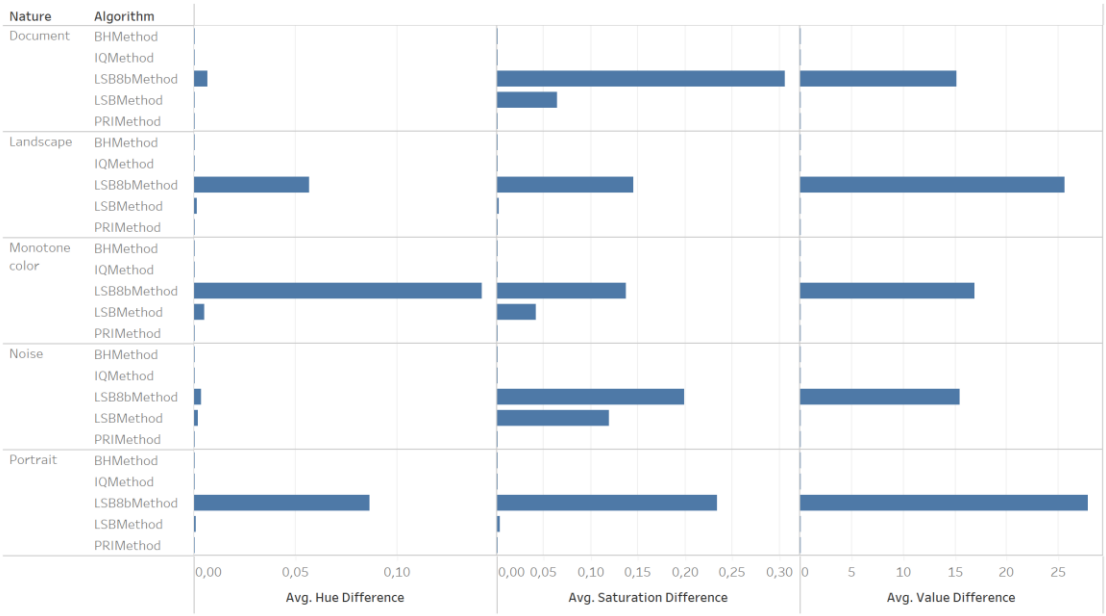

Рисунок 5.18 – Загальна гістограма результатів різницевих метрик

Ця метрика може доповнювати різницеві метрики надаючи більш комплексну оцінку. Головною її перевагою є здатність виявляти спотворення чорно-білих зображень, що є достатньо помітними навіть при невеликих значеннях. Детально результати для кожного зображення наведені на рисунку 5.19.

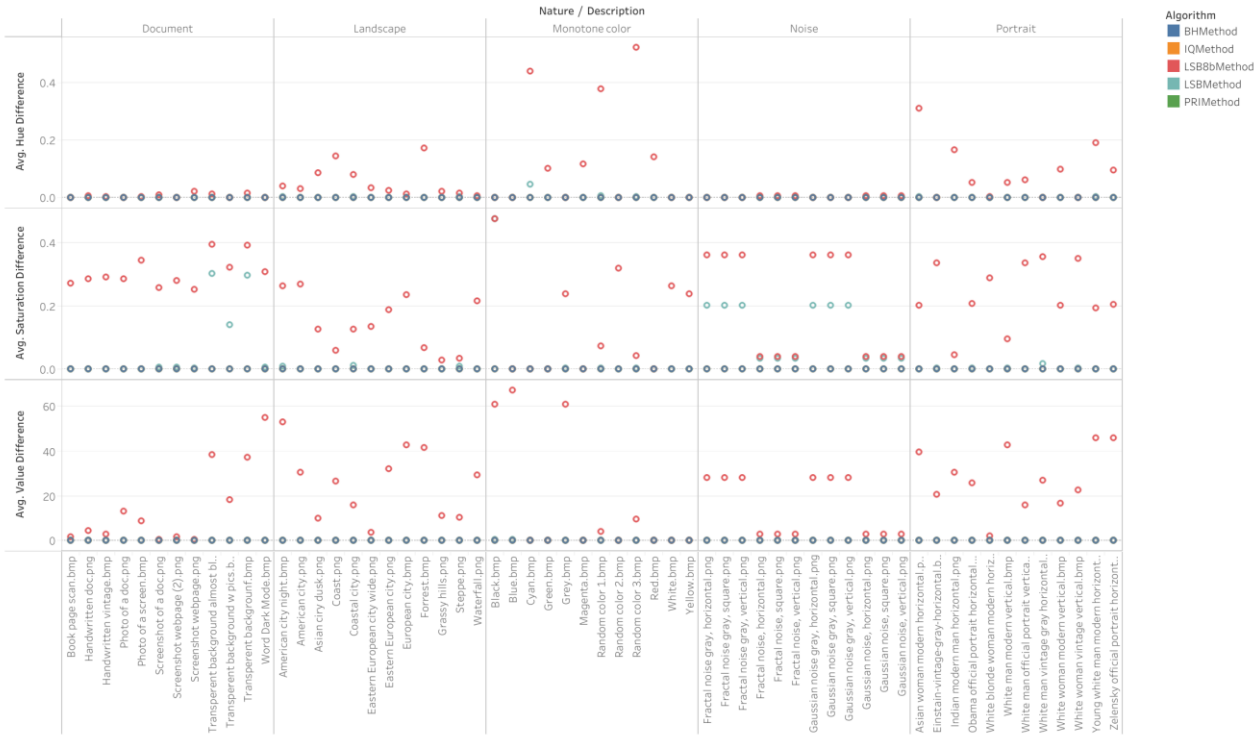

Рисунок 5.19 – Різницеві метрики кожного зображення

5.2.6 Абсолютна різниця за контрастом

Згідно результатам приведеним на рисунку 5.16:

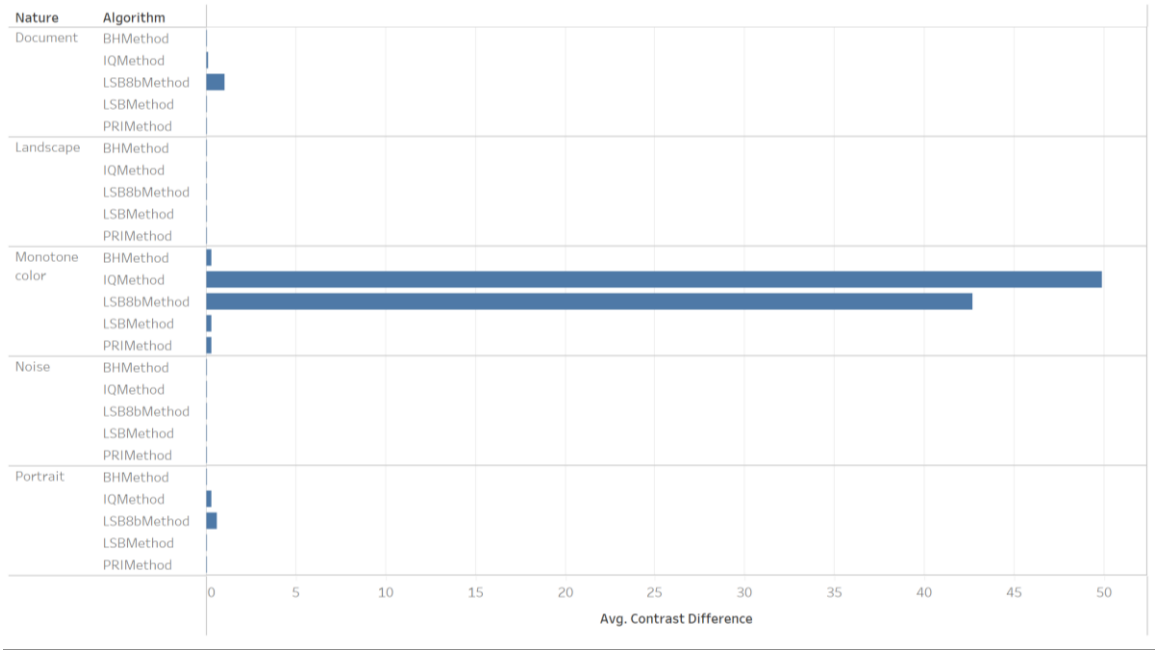

Рисунок 5.20 – Загальна гістограма результатів різницевих метрик

Ця має можливості виявлення значних спотворень в зображеннях низької контрастності якщо ті не будуть виявленні SSIM або UIQI. Детально результати для кожного зображення наведені на рисунку 5.21.

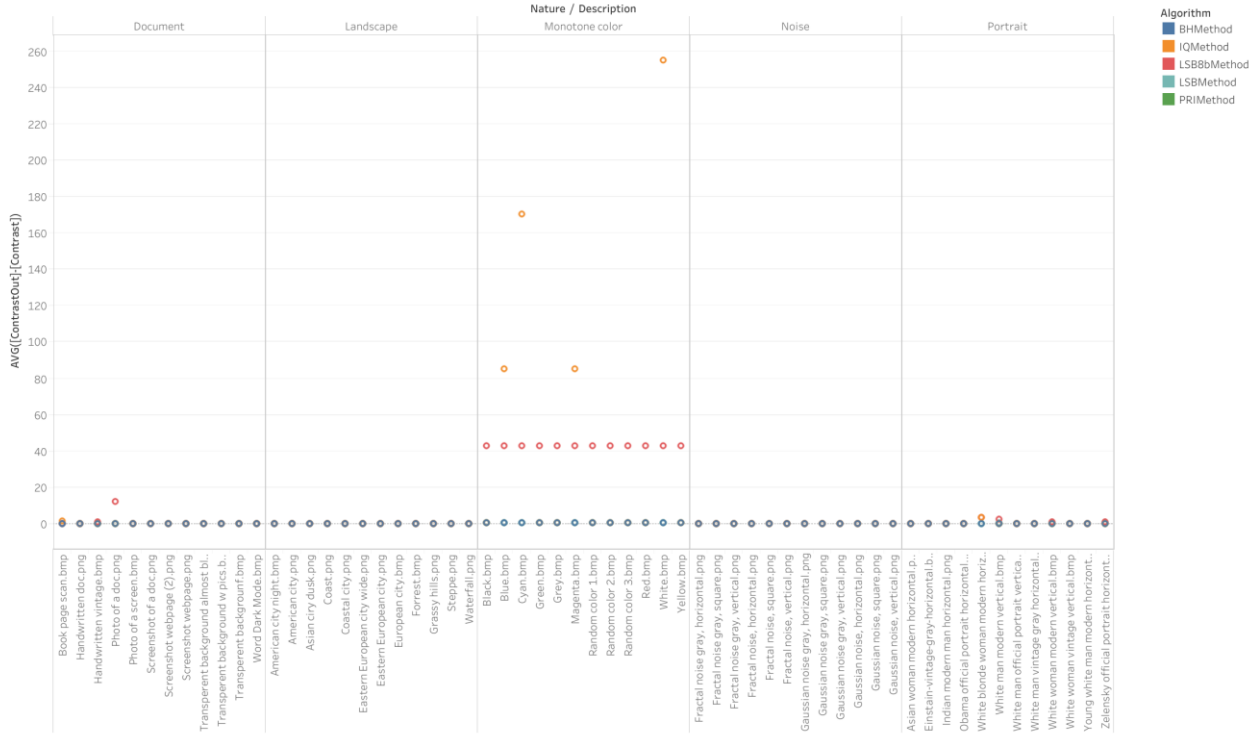

Рисунок 5.21 – Різницеві метрики кожного зображення

ВИСНОВКИ

Таким чином стеганографія залишається актуальною та активною галуззю досліджень, в якій постійно розвиваються як методи, так і застосування. Оскільки цифрові комунікації та технології продовжують розвиватися, цілком ймовірно, що стеганографія і надалі відіграватиме важливу роль у забезпеченні безпеки та захисті конфіденційної інформації.

Як слідує з проведеного аналізу стеганографія на базі графічних контейнерів базується на властивостях зорової системи людини, які можна використовувати для приховування інформації в цифрових носіях, мінімізуючи при цьому помітність прихованих даних для сторонніх спостерігачів.

У цілому розроблений метод має можливості автоматизовано виявлення при вбудовуванні, щоб знизити ризики зловмисного виявлення під час аналізу зображення у майбутньому.

Сукупність метрик дозволяє виявити вбудування у зображення неналежним алгоритмом, як-то вбудування у високо контрастне зображення методом квантування зображення, вбудування у монотонне, чорно-біле зображення, або вбудування що порушує структуру зображення.

До того ж проведений аналіз показав що зображення що містять пейзаж мають найбільші максимальні показники непомітності, однак при цьому зображення що містять шум мають теж дуже високу непомітність, при цьому усі метрики дуже стабільні в незалежності від розміру зображення. Також варто відмітити, що непомітність вбудування в зображення, що містить портрет була доволі нестабільною.

З реалізованих методів: метод заміни найменших значимих бітів, та його навмисно помітна версія що записує у найстарший біт, метод блочного приховування, метод псевдовипадкової перестановки, метод квантування

зображення, найкращі показники непомітності у меодів псевдовипадкової перестановки та блочного приховування. Реалізація методу квантування зображення, в свою чергу має вади що роблять вбудування дуже помітним на високо контрасних зображеннях.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Digital Watermarking and Steganography / J. Bloom et al. Elsevier Science & Technology Books, 2007.- 591 p.
2. Digital Steganography–An Introduction to Techniques and Tools / M. B. Pope та ін. Communications of the Association for Information Systems. 2012. Т. 30.
3. Комп'ютерна стеганографія: Теорія та практика : навч. посіб. Київ : МК-Пресс, 2006. 280 с.
4. Hussain M. A Survey of Image Steganography Techniques. International Journal of Advanced Science and Technology. (IJAST), 2013. 54. pp. 113-125.
5. Navjot K., Ashima B. A review on Digital Image Steganography. International Journal of Computer Science and Information Technologies. (IJCSIT), vol. 5, 2014, 8135-8137
6. Chakraborty S., Bikash Chowdhury A. Steganography Based on Human Perception. Oriental journal of computer science and technology. 2017. Т. 10, № 04. pp. 817–823. URL: https://doi.org/10.13005/ojcst/10.04.17 (дата звернення: 18.03.2023).
7. Fyffe B., Wang Y., Duncan I. Human visual based perc eption of steganographic images. Journal of Cyber Security Technology. 2019. Т. 3, № 2. pp. 61–107. URL: https://doi.org/10.1080/23742917.2019.1609393 (дата звернення: 22.03.2023).
8. Qi H., Zheng D., Zhao J. Human visual system based adaptive digital image watermarking. Signal Processing. 2008. Т. 88, № 1. pp. 174–188. URL: https://doi.org/10.1016/j.sigpro.2007.07.020 (дата звернення: 02.04.2023).
9. Techniques for data hiding / W. Bender та ін. IBM Systems Journal. 1996. Т. 35, № 3.4. С. 313–336. URL: https://doi.org/10.1147/sj.353.0313 (дата звернення: 31.03.2023).

10. Чунарьова А.В Потапенко Є.О. Аналіз сучасних методів стеганографічного захисту інформації. Education and Science. URL: http://www.rusnauka.com/29_DWS_2012/Informatica/4_120511.doc.htm (дата звернення: 24.03.2023).
11. Kasana G., Singh K., Bhatia S. S. Block-Based High Capacity Multilevel Image Steganography. Journal of Circuits, Systems and Computers. 2016. Т. 25, № 08. С. 1650091. URL: https://doi.org/10.1142/s0218126616500912 (дата звернення: 19.03.2023).
12. Hamdy M., Fathy Z., Ahmed H. Block Based Steganography. — American University in Cairo. 2017, — 12 p.
13. Хорошко В.О.. Азаров О.Д., Шелест М.Є.. Ярсмчук Ю.Є. Основи комп'ютерної стеганографії : Навчальний посібник для студентів і аспірантів. — Вінниця: ВДТУ, 2003, — 143 с.
14. BMP Format Overview - Win32 apps. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows/win32/wic/bmp-format-overview (дата звернення: 04.04.2023).
15. Bitmap Compression - Win32 apps. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows/win32/gdi/bitmap-compression (дата звернення: 04.04.2023).
16. Bitmap Storage - Win32 apps. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/en-us/windows/win32/gdi/bitmap-storage (дата звернення: 04.04.2023).
17. ISO/IEC 15948:2004 Information technology — Computer graphics and image processing — Portable Network Graphics (PNG): Functional specification, 2004. URL: https://www.iso.org/standard/29581.html (дата звернення: 06.04.2023).
18. Boutell T. PNG (Portable Network Graphics) Specification Version 1.0. RFC Editor, 1997. URL: https://doi.org/10.17487/rfc2083 (дата звернення: 06.04.2023).

19. Cattin. Ph. Image Restoration. Introduction to Signal and Image Processing. — MIAC, University of Basel, 2016, — 70 p.
20. Das A., Geisler W. Camouflage detection: experiments and a principled theory. Journal of Vision. 2022. T. 22, № 14. C. 4069. URL: https://doi.org/10.1167/jov.22.14.4069 (дата звернення: 10.04.2023).
21. Peli E. Contrast in complex images. Journal of the Optical Society of America A. 1990. T. 7, № 10. C. 2032. URL: https://doi.org/10.1364/josaa.7.002032 (дата звернення: 24.03.2023).
22. Color Spaces: S4 Classes and Utilities. A Toolbox for Manipulating and Assessing Colors and Palettes and colorspace. URL: https://colorspace.r-forge.r-project.org/articles/color_spaces.html (дата звернення: 14.04.2023).
23. Hema D., Kannan D. S. Interactive Color Image Segmentation using HSV Color Space. Science & Technology Journal. 2019. T. 7, № 1. C. 37–41. URL: https://doi.org/10.22232/stj.2019.07.01.05 (дата звернення: 14.04.2023).
24. Kutter M., Petitcolas F. A. P. Fair benchmark for image watermarking systems. Electronic Imaging '99, м. San Jose, CA / ред.: P. W. Wong, E. J. Delp III. 1999. URL: https://doi.org/10.1117/12.344672 (дата звернення: 22.04.2023).
25. Image Quality Assessment: From Error Visibility to Structural Similarity / Z. Wang та ін. IEEE Transactions on Image Processing. 2004. T. 13, № 4. pp. 600–612. URL: https://doi.org/10.1109/tip.2003.819861 (дата звернення: 21.04.2023).
26. Bovik A. C., Wang Z. Modern Image Quality Assessment. Morgan & Claypool Publishers, 2010.
27. Zhou Wang, Bovik A. C. A universal image quality index. IEEE Signal Processing Letters. 2002. T. 9, № 3. pp. 81–84. URL: https://doi.org/10.1109/97.995823 (дата звернення: 22.04.2023).

28. What is Java technology and why do I need it? Java. URL: https://www.java.com/en/download/help/whatis_java.html (дата звернення: 22.05.2023).
29. Apache Commons – Apache Commons. URL: https://commons.apache.org/ (дата звернення: 10.05.2023).
30. Apache POI - the Java API for Microsoft Documents. URL: https://poi.apache.org/ (дата звернення: 10.05.2023).

ДОДАТОК А.
ДІАГРАМА КЛАСІВ ПРОГРАМИ

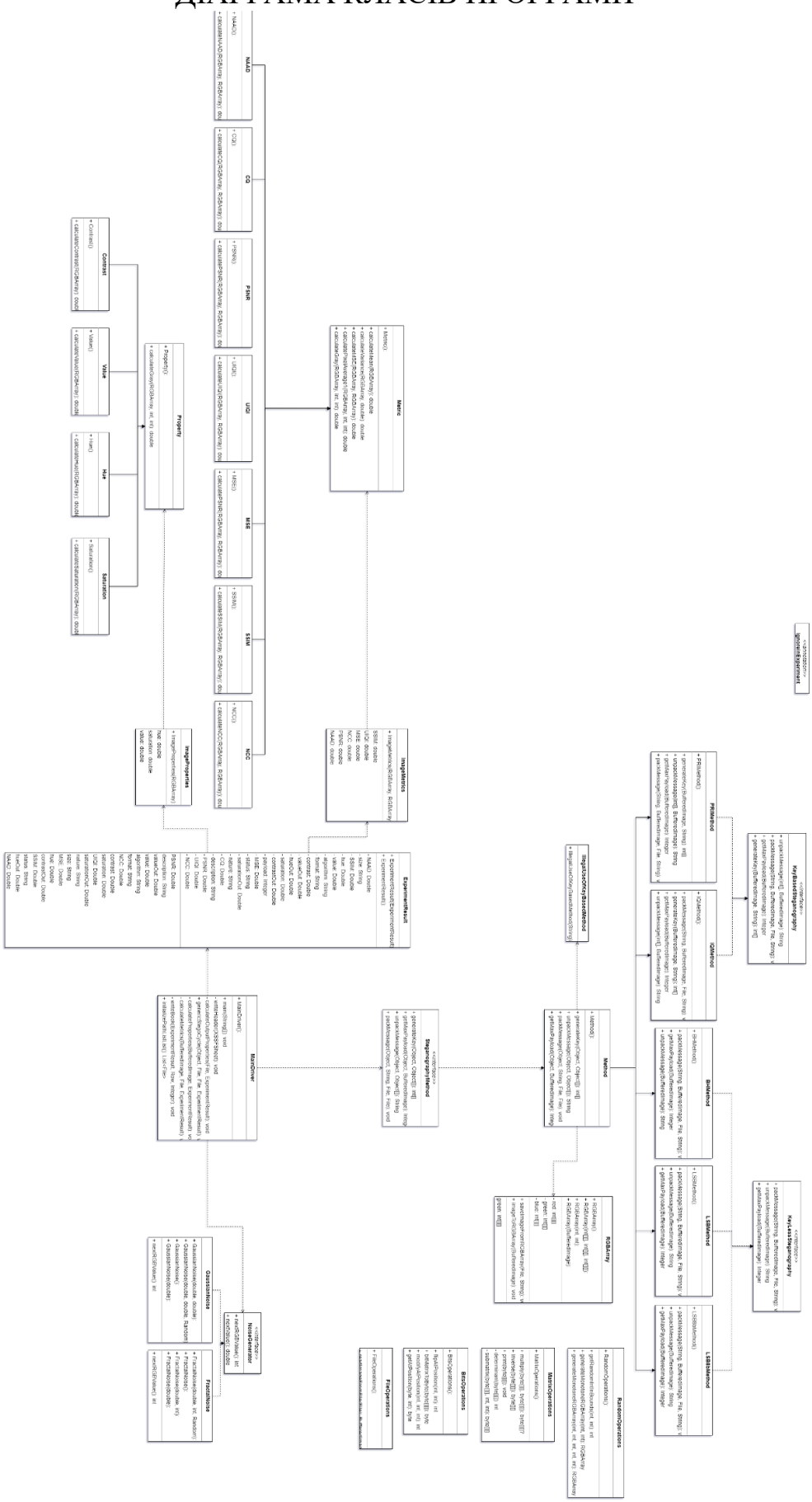

Рисунок А.1 – Діаграма класів програми

ДОДАТОК Б.
ВИХІДНИЙ КОД ОСНОВНИХ КОМПОНЕНТІВ ПРОГРАМИ

Лістинг Б.1 – Компонент «Main Driver»

```
package drivers;


import org.apache.commons.io.FileUtils;
import org.apache.commons.io.FilenameUtils;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import steganography.Method;
import steganography.interfaces.KeyBasedSteganography;
import steganography.keybased.IQMethod;
import steganography.keybased.PRIMethod;
import steganography.keyless.BHMethod;
import steganography.keyless.LSB8bMethod;
import steganography.keyless.LSBMethod;
import untility.ExperimentResult;
import untility.metrics.ImageMetrics;
import untility.operations.FileOperations;
import untility.RGBArray;


import java.awt.image.BufferedImage;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;


import org.apache.commons.lang.RandomStringUtils;
import untility.properties.ImageProperties;


public class MainDriver {


    public static void main(String[] args) throws IOException {
        File experimentDirectory = new File("src\\main\\resources");
        File outputFile = new File("src\\main\\resources\\Output");
```

```
        String[] extensions = new String[] { "bmp", "png" };
        List<File> pictures = (List<File>)
FileUtils.listFiles(experimentDirectory, extensions, true);
        List<ExperimentResult> experiments = new ArrayList<>();

        for (File picture : pictures) {
            System.out.println("picture: " + picture.getCanonicalPath() + "is
loaded");
        }

        for (File picture : pictures) {
            ExperimentResult experimentResult = new ExperimentResult();
            experimentResult.setNature(picture.getName().split("_")[0]);
            experimentResult.setDescription(picture.getName().split("_")[1]);

experimentResult.setFormat(FilenameUtils.getExtension(picture.getPath()));

            BufferedImage img = FileOperations.readImageFromFile(picture);
            calculateProperties(img, experimentResult);
            for(Object method : getAllMethods()){
                ExperimentResult experimentResultAlgo = new
ExperimentResult(experimentResult);
                genericStegoCycle(method, picture, outputFile,
experimentResultAlgo);
                calculateMetrics(img, outputFile, experimentResultAlgo);
                calculateOutputProperties(outputFile, experimentResultAlgo);
                experiments.add(experimentResultAlgo);
            }
        }

            XSSFWorkbook workbook = new XSSFWorkbook();
            XSSFSheet sheet = workbook.createSheet("Experiments");
            writeHeader(sheet);

            int rowCount = 1;
            for (ExperimentResult experiment : experiments) {
                Row row = sheet.createRow(++rowCount);
                writeBook(experiment, row, rowCount - 1);
            }
```

```
        try (FileOutputStream outputStream = new
FileOutputStream("src\\main\\resources\\ExperimentResults.xlsx")) {
                workbook.write(outputStream);
        }
    }

    private static void calculateMetrics(BufferedImage img, File outputFile,
ExperimentResult experimentResult) {
        BufferedImage stegoImage = FileOperations.readImageFromFile(outputFile);
        RGBArray originArray = new RGBArray();
        originArray.imageToRGBArray(img);
        RGBArray stegoArray = new RGBArray();
        stegoArray.imageToRGBArray(stegoImage);
        ImageMetrics metrics = new ImageMetrics(originArray, stegoArray);



        experimentResult.setNAAD(metrics.getNAAD());
        System.out.println("NAAD = " + experimentResult.getNAAD());
        experimentResult.setMSE(metrics.getMSE());
        System.out.println("MSE = " + experimentResult.getMSE());
        experimentResult.setNCC(metrics.getNCC());
        System.out.println("NCC = " + experimentResult.getNCC());
        experimentResult.setCQ(metrics.getCQ());
        System.out.println("CQ = " + experimentResult.getCQ());
        experimentResult.setPSNR(metrics.getPSNR());
        System.out.println("PSNR = " + experimentResult.getPSNR());
        experimentResult.setSSIM(metrics.getSSIM());
        System.out.println("SSIM = " + experimentResult.getSSIM());
        experimentResult.setUIQI(metrics.getUIQI());
        System.out.println("UIQI = " + experimentResult.getUIQI());


    }

    private static void calculateProperties(BufferedImage img, ExperimentResult
experimentResult) {
        experimentResult.setSize((img.getWidth() + "x" + img.getHeight()));
        RGBArray originArray = new RGBArray();
        originArray.imageToRGBArray(img);
```

```
        ImageProperties properties = new ImageProperties(originArray);


        experimentResult.setContrast(properties.getContrast());
        System.out.println("Contrast = " + experimentResult.getContrast());
        experimentResult.setHue(properties.getHue());
        System.out.println("Hue = " + experimentResult.getHue());
        experimentResult.setSaturation(properties.getSaturation());
        System.out.println("Saturation = " + experimentResult.getSaturation());
        experimentResult.setValue(properties.getValue());
        System.out.println("Value = " + experimentResult.getValue());
    }


    private static void calculateOutputProperties(File outputFile,
ExperimentResult experimentResult) {
        BufferedImage stegoImage = FileOperations.readImageFromFile(outputFile);
        RGBArray stegoArray = new RGBArray();
        stegoArray.imageToRGBArray(stegoImage);


        ImageProperties properties = new ImageProperties(stegoArray);


        experimentResult.setContrastOut(properties.getContrast());
        System.out.println("Contrast Out = " + experimentResult.getContrastOut());
        experimentResult.setHueOut(properties.getHue());
        System.out.println("Hue Out = " + experimentResult.getHueOut());
        experimentResult.setSaturationOut(properties.getSaturation());
        System.out.println("Saturation Out = " +
experimentResult.getSaturationOut());
        experimentResult.setValueOut(properties.getValue());
        System.out.println("Value = Out " + experimentResult.getValueOut());
    }


    public static void genericStegoCycle(Object method, File inputFile, File
outputFile, ExperimentResult experimentResult) throws IOException {
        System.out.println("Using method " + method.getClass().toString() + " on
file " + inputFile.getPath());

experimentResult.setAlgorithm(method.getClass().toString().split("\\.")[method.get
Class().toString().split("\\.").length - 1]);
        BufferedImage image = FileOperations.readImageFromFile(inputFile);
        Method genericMethod = new Method();
```

```
        int maxPayload = genericMethod.getMaxPayload(method, image);
        String randomStr = RandomStringUtils.randomAscii(maxPayload / 8);
        experimentResult.setPayload(maxPayload);
        int[] key = null;
        if(method instanceof KeyBasedSteganography){
            key = genericMethod.generateKey(method,randomStr, image);
        }
        genericMethod.packMessage(method,randomStr, inputFile, outputFile);
        BufferedImage imgContainer = FileOperations.readImageFromFile(outputFile);
        if(randomStr.equals(genericMethod.unpackMessage(method, key,
imgContainer))){
            experimentResult.setStatus("Success");
            System.out.println("Success");
        }
        else {
            experimentResult.setStatus("Warning");
            System.out.println("Warning");
        }
    }

    public static List<Object> getAllMethods(){
        List<Object> methods = new ArrayList<Object>();
        methods.add(new LSBMethod());
        //methods.add(new PRPMethod());
        methods.add(new PRIMethod());
        methods.add(new BHMethod());
        methods.add(new IQMethod());
        //methods.add(new KJBMethod());
        methods.add(new LSB8bMethod());
        return methods;
    }

    public static List<File> initializePathListList(){
        List<File> stegoPathes = new ArrayList<File>();
        stegoPathes.add(new File("src\\main\\resources\\StegoLSB.bmp"));
        stegoPathes.add(new File("src\\main\\resources\\StegoPRI.bmp"));
        //stegoPathes.add(new File("src\\main\\resources\\StegoPRP.bmp"));
        stegoPathes.add(new File("src\\main\\resources\\StegoBH.bmp"));
```

```
        stegoPathes.add(new File("src\\main\\resources\\StegoIQ.bmp"));
        //stegoPathes.add(new File("src\\main\\resources\\StegoKJB.bmp"));
        return stegoPathes;
    }
}
```

Лістинг Б.2 – Компонент «Steganography»

```
package steganography;

import exeptions.IllegalUseOfKeyBasedMethod;
import steganography.interfaces.KeyBasedSteganography;
import steganography.interfaces.KeyLessSteganography;
import steganography.interfaces.SteganographyMethod;
import untility.operations.FileOperations;

import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import org.apache.commons.io.FilenameUtils;


public class Method implements SteganographyMethod {

    @Override
    public String getName() {
        return null;
    }

    @Override
    public int[] generateKey(Object method, Object... args) {
        BufferedImage image = null;
        String message = null;
        int[] key;
        if (method instanceof KeyBasedSteganography) {
            for (Object obj : args) {
                if (obj instanceof BufferedImage) {
                    image = (BufferedImage) obj;
```

```
                } else if (obj instanceof String) {
                    message = (String) obj;
                } else {
                    if(obj == null) {
                        System.out.println("WARNING: Found VarArgDemo of value - 
null");
                    }else{
                        System.out.println("WARNING: Found VarArgDemo of value - " 
+ obj.toString() + " - might be unregnized ");
                    }
                }
            }
            if (message == null || image == null) {
                throw new IllegalArgumentException("Not enough arguments 
provided");
            }
            key = ((KeyBasedSteganography) method).generateKey(image, message);
        } else if (method instanceof KeyLessSteganography) {
            try {
                throw new IllegalUseOfKeyBasedMethod("Cannot generateKey for 
KeyLessSteganography method - " + method.getClass().toString());
            } catch (IllegalUseOfKeyBasedMethod e) {
                throw new RuntimeException(e);
            }
        } else {
            throw new IllegalArgumentException("Method " + 
method.getClass().toString() + " is not recognized");
        }
        return key;
    }


    @Override
    public void packMessage(Object method, String message, File inputFile, File 
outputFile) throws IOException {
        if(method instanceof KeyLessSteganography){
            ((KeyLessSteganography) method)
                    .packMessage(message,
                            FileOperations.readImageFromFile(inputFile),
                            outputFile,
                            FilenameUtils.getExtension(inputFile.getPath()));
```

```
        } else if(method instanceof KeyBasedSteganography){
            ((KeyBasedSteganography) method)
                    .packMessage(message,
                            FileOperations.readImageFromFile(inputFile),
                            outputFile,
                            FilenameUtils.getExtension(inputFile.getPath()));
        } else {
            throw new IllegalArgumentException("Method " +
method.getClass().toString() + " is not recognized");
        }
    }

    @Override
    public String unpackMessage(Object method, Object... args) throws IOException
{
        int[] key = null;
        BufferedImage image = null;
        String message = null;

        for (Object obj: args) {
            if(obj instanceof int[]){
                key = (int[]) obj;
            } else if(obj instanceof BufferedImage){
                image = (BufferedImage) obj;
            } else{
                if(obj == null) {
                    System.out.println("WARNING: Found VarArgDemo of value -
null");
                }else{
                    System.out.println("WARNING: Found VarArgDemo of value - " +
obj.toString() + " - might be unrecognized ");
                }
            }
        }

        if(method instanceof KeyLessSteganography){
            message = ((KeyLessSteganography) method).unpackMessage(image);
        } else if(method instanceof KeyBasedSteganography){
            message = ((KeyBasedSteganography) method).unpackMessage(key, image);
        } else {
```

```
            throw new IllegalArgumentException("Method " +
method.getClass().toString() + " is not unrecognized");
        }
        return message;
    }

    @Override
    public Integer getMaxPayload(Object method, BufferedImage container) {
        Integer result;
        if(method instanceof KeyLessSteganography){
            result = ((KeyLessSteganography) method).getMaxPayload(container);
        } else if(method instanceof KeyBasedSteganography){
            result = ((KeyBasedSteganography) method).getMaxPayload(container);
        } else {
            throw new IllegalArgumentException("Method " +
method.getClass().toString() + " is not recognized");
        }
        return result;
    }
}
```

Лістинг Б.3 – Компонент Metrics»

```
package untility.metrics;

import untility.RGBArray;

abstract public class Metric {
    public static double calculateMean(RGBArray rgbArray) {
        double mean = 0;
        for (int y = 0; y < rgbArray.getBlue().length; y++) {
            for (int x = 0; x < rgbArray.getBlue()[0].length; x++) {
                mean += calculateGray(rgbArray, x, y);
            }
        }
        mean = mean / (rgbArray.getBlue().length * rgbArray.getBlue()[0].length);
        return mean;
    }

    public static double calculateVariance(RGBArray rgbArray, double mean){
```

```
        double std = 0;
        for (int y = 0; y < rgbArray.getBlue().length; y++) {
            for (int x = 0; x < rgbArray.getBlue()[0].length; x++) {
                std += Math.pow(calculateGray(rgbArray, x, y) - mean, 2);
            }
        }
        std = std / (double) (rgbArray.getBlue().length *
rgbArray.getBlue()[0].length - 1);
        return std;
    }

    public static double calculateCrossCovariance(RGBArray rgbArray1, RGBArray
rgbArray2, double mean1, double mean2){
        double crossCovariance = 0;
        for (int y = 0; y < rgbArray1.getBlue().length; y++) {
            for (int x = 0; x < rgbArray1.getBlue()[0].length; x++) {
                crossCovariance +=  (calculateGray(rgbArray1, x, y) - mean1) *
(calculateGray(rgbArray2, x, y) - mean2);;
            }
        }
        crossCovariance = crossCovariance / (double) (rgbArray1.getBlue().length *
rgbArray1.getBlue()[0].length - 1);
        return crossCovariance;
    }

    public static double calculateMSE(RGBArray rgbArray1, RGBArray rgbArray2){
        double mse = 0;
        for (int y = 0; y < rgbArray1.getBlue().length; y++) {
            for (int x = 0; x < rgbArray1.getBlue()[0].length; x++) {
                mse +=  Math.pow((calculateGray(rgbArray1, x, y) -
calculateGray(rgbArray2, x, y)), 2);
            }
        }
        mse = mse / (rgbArray1.getBlue().length * rgbArray1.getBlue()[0].length);
        return mse;
    }

    public static double calculatePixelAverage1(RGBArray rgbArray, int x, int y){
        return (rgbArray.getRed()[y][x] + rgbArray.getGreen()[y][x] +
rgbArray.getBlue()[y][x]) / 3.0;
```

```
    }


    public static double calculateGray(RGBArray rgbArray, int x, int y){

        return (0.2989 * rgbArray.getRed()[y][x] +0.5870  *
rgbArray.getGreen()[y][x] + 0.1140  * rgbArray.getBlue()[y][x]);

    }


}
```

ДОДАТОК В.
СПИСОК ПРОТЕСТОВАНИХ ЗОБРАЖЕНЬ

Таблиця В.1 – Список протестованих зображень

Опис	Вміст	Формат	Розмір
Black	Monotone color	bmp	1080x1080
Blue	Monotone color	bmp	1080x1080
Cyan	Monotone color	bmp	1080x1080
Green	Monotone color	bmp	1080x1080
Grey	Monotone color	bmp	1080x1080
Magenta	Monotone color	bmp	1080x1080
Random color 1	Monotone color	bmp	1080x1080
Random color 2	Monotone color	bmp	1080x1080
Random color 3	Monotone color	bmp	1080x1080
Red	Monotone color	bmp	1080x1080
White	Monotone color	bmp	1080x1080
Yellow	Monotone color	bmp	1080x1080
Fractal noise gray, horizontal	Noise	png	1920x1080
Fractal noise gray, square	Noise	png	1920x1920
Fractal noise gray, vertical	Noise	png	1080x1920
Fractal noise, horizontal	Noise	png	1920x1080
Fractal noise, square	Noise	png	1920x1920
Fractal noise, vertical	Noise	png	1080x1920
Gaussian noise gray, horizontal	Noise	png	1920x1080
Gaussian noise gray, square	Noise	png	1920x1920
Gaussian noise gray, vertical	Noise	png	1080x1920
Gaussian noise, horizontal	Noise	png	1920x1080
Gaussian noise, square	Noise	png	1920x1920
Gaussian noise, vertical	Noise	png	1080x1920
Book page scan	Document	bmp	541x825
Handwritten doc	Document	png	787x1019
Handwritten vintage	Document	bmp	710x1024

Продовження таблиці В.1

Photo of a doc	Document	png	270x456
Photo of a screen	Document	bmp	768x1024
Screenshot of a doc	Document	png	756x972
Screenshot webpage (2)	Document	png	1083x882
Screenshot webpage	Document	png	1224x1955
Transperent background almost blank	Document	bmp	1700x2200
Transperent background w pics	Document	bmp	1700x2200
Transperent backgrounf	Document	bmp	1700x2200
Word Dark Mode	Document	bmp	1125x743
American city night	Landscape	bmp	970x646
American city	Landscape	png	709x492
Asian ciry dusk	Landscape	png	1024x683
Coast	Landscape	png	2560x1985
Coastal city	Landscape	png	4560x3040
Eastern European city wide	Landscape	png	6675x2736
Eastern European city	Landscape	png	1024x683
European city	Landscape	bmp	910x607
Forrest	Landscape	bmp	2048x1363
Grassy hills	Landscape	png	1024x683
Steppe	Landscape	png	1024x637
Waterfall	Landscape	png	910x683
Asian woman modern horizonta	Portrait	png	910x1365
Einstain-vintage-gray-horizontal	Portrait	bmp	3250x4333
Indian modern man horizontal	Portrait	png	910x1365
Obama official portrait horizontal	Portrait	bmp	1314x1752
White blonde woman modern horizontal	Portrait	png	2891x3840
White man modern vertical	Portrait	bmp	910x607

Продовження таблиці В.1

White man official portrait vertical	Portrait	bmp	3000x2000
White man vintage gray horizontal	Portrait	png	179x281
White woman modern vertical	Portrait	bmp	5184x3456
White woman vintage vertical	Portrait	bmp	1600x1067
Young white man modern horizontal	Portrait	png	408x615
Zelensky official portrait horizontal	Portrait	png	2238x2985

ДОДАТОК Г.
СПИСОК РЕЗУЛЬТАТІВ ЕКСПЕРИМЕНТІВ

Таблиця Г.1 – Список результатів експериментів

ID	Algorithm	Payload	NAAD	MSE	NCC	CQ	PSNR	SSIM	UIQI	DeltaContrast	DeltaHue	DeltaSaturation	DeltaValue
1	LSBMethod	1E+06	########	0,006	########	########	70,22	0,999	0	0,333	0	0,476	0,476
2	PRIMethod	1080	########	6E-06	########	########	100,4	1	0	0,333	0	5E-04	5E-04
3	BHMethod	1080	########	6E-06	########	########	100,4	1	0	0,333	0	5E-04	5E-04
4	IQMethod	1080	########	7E-06	########	########	100	1	0	0,333	0	5E-04	5E-04
5	LSB8bMethod	1E+06	########	101,2	########	########	28,08	0,062	0	42,67	0	0,475	60,84
6	LSBMethod	1E+06	0,002	0,007	0,998	29,01	69,79	1	3E-17	0,333	0	0	0,525
7	PRIMethod	1080	2E-06	6E-06	1	29,07	100,1	1	2E-14	0,333	0	0	5E-04
8	BHMethod	1080	2E-06	6E-06	1	29,07	100,5	1	2E-14	0,333	0	0	4E-04
9	IQMethod	1080	4E-04	0,366	1	29,06	52,5	0,994	3E-19	85,33	9E-04	0	0,115
10	LSB8bMethod	1E+06	0,263	111,7	0,737	21,41	27,65	0,501	2E-21	42,67	0	0	67,15
11	LSBMethod	1E+06	3E-04	0,007	1	178,7	69,8	1	4E-16	0,333	0,046	0	0
12	PRIMethod	1080	3E-07	6E-06	1	178,8	100,1	1	3E-12	0,333	4E-05	0	0
13	BHMethod	1080	3E-07	6E-06	1	178,8	100,6	1	3E-12	0,333	4E-05	0	0
14	IQMethod	1080	4E-04	13,57	1	178,7	36,81	0,812	1E-18	170,3	9E-04	0	0,108
15	LSB8bMethod	1E+06	0,043	111,8	0,957	171,1	27,64	0,524	3E-19	42,67	0,439	0	0
16	LSBMethod	1E+06	4E-04	0,006	1	149,7	70,22	1	-4E-16	0,333	5E-04	0	0
17	PRIMethod	1080	3E-07	6E-06	1	149,7	100,4	1	1E-12	0,333	5E-07	0	0
18	BHMethod	1080	3E-07	6E-06	1	149,7	100,6	1	1E-12	0,333	5E-07	0	0
19	IQMethod	1080	3E-07	6E-06	1	149,7	100,6	1	1E-12	0,333	5E-07	0	0
20	LSB8bMethod	1E+06	0,046	101,1	1,046	156,6	28,08	0,524	1E-19	42,67	0,101	0	0
21	LSBMethod	1E+06	5E-04	0,007	1	126,9	69,8	1	1E-15	0,333	0	0,004	0
22	PRIMethod	1080	4E-07	6E-06	1	127	100,1	1	1E-12	0,333	0	4E-06	0
23	BHMethod	1080	4E-07	6E-06	1	127	100,6	1	1E-12	0,333	0	3E-06	0

Продовження таблиці Г.1

24	IQMethod	1080	4E-07	6E-06	1	127	100,4	1	1E-12	0,333	0	4E-06	5E-04
25	LSB8bMethod	1E+06	0,054	101	1,054	133,9	28,09	0,524	7E-20	42,67	0	0,238	60,7
26	LSBMethod	1E+06	6E-04	0,007	0,999	105,2	69,79	1	-4E-16	0,333	0,001	0	0
27	PRIMethod	1080	5E-07	6E-06	1	105,3	100,1	1	4E-13	0,333	9E-07	0	0
28	BHMethod	1080	5E-07	6E-06	1	105,3	100,4	1	4E-13	0,333	9E-07	0	0
29	IQMethod	1080	1E-04	0,392	1	105,3	52,19	0,993	7E-18	85,33	3E-04	0	0
30	LSB8bMethod	1E+06	0,073	111,7	0,927	97,64	27,65	0,523	-2E-20	42,67	0,119	0	0
31	LSBMethod	1E+06	1E-03	0,007	0,999	62,01	69,79	1	4E-16	0,333	0,007	0,004	0
32	PRIMethod	1080	9E-07	6E-06	1	62,07	100,2	1	3E-13	0,333	7E-06	4E-06	0
33	BHMethod	1080	8E-07	6E-06	1	62,07	100,7	1	3E-13	0,333	6E-06	3E-06	0
34	IQMethod	1080	8E-07	6E-06	1	62,07	100,5	1	3E-13	0,333	6E-06	3E-06	0
35	LSB8bMethod	1E+06	0,112	101,1	1,112	69	28,08	0,521	2E-20	42,67	0,376	0,074	4,275
36	LSBMethod	1E+06	3E-04	0,006	1	182,8	70,22	1	1E-15	0,333	0	0,002	0
37	PRIMethod	1080	3E-07	6E-06	1	182,7	100,4	1	7E-13	0,333	0	2E-06	0
38	BHMethod	1080	3E-07	5E-06	1	182,7	100,7	1	7E-13	0,333	0	2E-06	0
39	IQMethod	1080	3E-07	6E-06	1	182,7	100,6	1	7E-13	0,333	0	2E-06	0
40	LSB8bMethod	1E+06	0,042	112	0,958	175,1	27,64	0,524	8E-20	42,67	0	0,319	0
41	LSBMethod	1E+06	5E-04	0,007	0,999	116,5	69,79	1	1E-15	0,333	0,005	0,003	0
42	PRIMethod	1080	5E-07	7E-06	1	116,5	99,96	1	2E-12	0,333	5E-06	3E-06	0
43	BHMethod	1080	4E-07	6E-06	1	116,5	100,4	1	2E-12	0,333	5E-06	3E-06	0
44	IQMethod	1080	5E-07	6E-06	1	116,5	100,2	1	2E-12	0,333	5E-06	3E-06	0
45	LSB8bMethod	1E+06	0,059	101	1,059	123,5	28,09	0,523	1E-19	42,67	0,523	0,044	9,488
46	LSBMethod	1E+06	7E-04	0,006	1,001	76,27	70,23	1	-6E-17	0,333	0,002	0	0
47	PRIMethod	1080	7E-07	6E-06	1	76,22	100,5	1	2E-13	0,333	1E-06	0	0
48	BHMethod	1080	7E-07	6E-06	1	76,22	100,3	1	2E-13	0,333	2E-06	0	0

Продовження таблиці Г.1

49	IQMethod	1080	6E-07	6E-06	1	76,22	100,7	1	2E-13	0,333	1E-06	0	0
50	LSB8bMethod	1E+06	0,091	101,1	1,091	83,15	28,08	0,522	2E-21	42,67	0,142	0	0
51	LSBMethod	1E+06	2E-04	0,007	1	254,9	69,79	1	5E-17	0,333	0	0,002	0
52	PRIMethod	1080	2E-07	6E-06	1	255	100,3	1	7E-14	0,333	0	2E-06	0
53	BHMethod	1080	2E-07	6E-06	1	255	100,5	1	7E-14	0,333	0	2E-06	0
54	IQMethod	1080	5E-04	30,04	1	254,9	33,35	0,661	1E-20	255	0	0	0,118
55	LSB8bMethod	1E+06	0,03	111,8	0,97	247,3	27,65	0,524	-5E-20	42,67	0	0,264	0
56	LSBMethod	1E+06	2E-04	0,006	1	226	70,24	1	1E-15	0,333	0	0,002	0
57	PRIMethod	1080	2E-07	6E-06	1	225,9	100,5	1	1E-12	0,333	0	2E-06	0
58	BHMethod	1080	2E-07	6E-06	1	225,9	100,5	1	1E-12	0,333	0	2E-06	0
59	IQMethod	1080	2E-07	6E-06	1	225,9	100,7	1	1E-12	0,333	0	2E-06	0
60	LSB8bMethod	1E+06	0,031	101,3	1,031	232,8	28,08	0,524	-2E-19	42,67	0	0,239	0
61	LSBMethod	2E+06	4E-04	0,007	1	248,7	69,98	1	1	0	0	0,202	0,219
62	PRIMethod	1920	2E-07	3E-06	1	248,8	102,7	1	1	0	0	1E-04	1E-04
63	BHMethod	1920	2E-07	3E-06	1	248,8	103	1	1	0	0	1E-04	1E-04
64	IQMethod	1920	3E-04	9,374	1	248,7	38,41	1	1	0	0	1E-04	0,037
65	LSB8bMethod	2E+06	0,053	106,9	0,97	241,4	27,84	0,996	0,996	0	0	0,359	27,99
66	LSBMethod	4E+06	4E-04	0,007	1	248,7	69,98	1	1	0	0	0,202	0,219
67	PRIMethod	1920	2E-07	3E-06	1	248,8	102,8	1	1	0	0	1E-04	1E-04
68	BHMethod	1920	2E-07	3E-06	1	248,8	102,9	1	1	0	0	1E-04	1E-04
69	IQMethod	1920	3E-04	9,206	1	248,7	38,49	1	1	0	0	1E-04	0,036
70	LSB8bMethod	4E+06	0,053	106,9	0,97	241,4	27,84	0,996	0,996	0	0	0,359	28
71	LSBMethod	2E+06	4E-04	0,007	1	248,7	69,99	1	1	0	0	0,202	0,219
72	PRIMethod	1920	2E-07	3E-06	1	248,8	102,8	1	1	0	0	1E-04	1E-04
73	BHMethod	1920	2E-07	3E-06	1	248,8	102,8	1	1	0	0	1E-04	1E-04

Продовження таблиці Г.1

74	IQMethod	1920	2E-04	8,434	1	248,7	38,87	1	1	0	0	1E-04	0,033
75	LSB8bMethod	2E+06	0,053	107	0,97	241,4	27,84	0,996	0,996	0	0	0,36	28,02
76	LSBMethod	2E+06	4E-04	0,007	1	187,2	69,99	1	1	0	0,004	0,036	0,03
77	PRIMethod	1920	2E-07	3E-06	1	187,3	102,9	1	1	0	2E-06	2E-05	1E-05
78	BHMethod	1920	2E-07	4E-06	1	187,3	102,6	1	1	0	2E-06	2E-05	2E-05
79	IQMethod	1920	1E-04	3,578	1	187,2	42,59	1	1	0	2E-05	2E-05	0,024
80	LSB8bMethod	2E+06	0,053	106,8	0,991	185,6	27,84	0,992	0,992	0	0,007	0,04	2,865
81	LSBMethod	4E+06	4E-04	0,007	1	187,3	69,98	1	1	0	0,004	0,035	0,031
82	PRIMethod	1920	2E-07	3E-06	1	187,3	102,9	1	1	0	2E-06	2E-05	2E-05
83	BHMethod	1920	2E-07	3E-06	1	187,3	102,8	1	1	0	3E-06	2E-05	1E-05
84	IQMethod	1920	1E-04	3,047	1	187,3	43,29	1	1	0	1E-05	2E-05	0,022
85	LSB8bMethod	4E+06	0,053	106,9	0,991	185,7	27,84	0,992	0,992	0	0,007	0,039	2,844
86	LSBMethod	2E+06	4E-04	0,007	1	187,3	69,98	1	1	0	0,004	0,035	0,03
87	PRIMethod	1920	2E-07	3E-06	1	187,3	102,9	1	1	0	3E-06	2E-05	1E-05
88	BHMethod	1920	2E-07	3E-06	1	187,3	102,8	1	1	0	3E-06	2E-05	2E-05
89	IQMethod	1920	1E-04	3,693	1	187,3	42,46	1	1	0	2E-05	3E-05	0,027
90	LSB8bMethod	2E+06	0,053	107,1	0,991	185,6	27,83	0,992	0,992	0	0,007	0,04	2,815
91	LSBMethod	2E+06	4E-04	0,007	1	248,7	69,98	1	1	0	0	0,202	0,219
92	PRIMethod	1920	2E-07	3E-06	1	248,8	102,7	1	1	0	0	1E-04	1E-04
93	BHMethod	1920	2E-07	3E-06	1	248,8	102,7	1	1	0	0	1E-04	1E-04
94	IQMethod	1920	2E-04	8,183	1	248,7	39	1	1	0	0	1E-04	0,032
95	LSB8bMethod	2E+06	0,053	106,9	0,97	241,4	27,84	0,996	0,996	0	0	0,359	28,02
96	LSBMethod	4E+06	4E-04	0,007	1	248,7	69,99	1	1	0	0	0,202	0,219
97	PRIMethod	1920	2E-07	3E-06	1	248,7	102,8	1	1	0	0	1E-04	1E-04
98	BHMethod	1920	2E-07	3E-06	1	248,7	102,9	1	1	0	0	1E-04	1E-04

Продовження таблиці Г.1

99	IQMethod	1920	3E-04	8,818	1	248,7	38,68	1	1	0	0	1E-04	0,034
100	LSB8bMethod	4E+06	0,053	106,8	0,97	241,3	27,84	0,996	0,996	0	0	0,359	27,96
101	LSBMethod	2E+06	4E-04	0,007	1	248,7	69,99	1	1	0	0	0,202	0,218
102	PRIMethod	1920	2E-07	3E-06	1	248,8	102,8	1	1	0	0	1E-04	1E-04
103	BHMethod	1920	2E-07	4E-06	1	248,8	102,7	1	1	0	0	1E-04	1E-04
104	IQMethod	1920	2E-04	8,716	1	248,7	38,73	1	1	0	0	1E-04	0,034
105	LSB8bMethod	2E+06	0,053	107	0,97	241,4	27,84	0,996	0,996	0	0	0,359	27,99
106	LSBMethod	2E+06	4E-04	0,007	1	187,3	69,98	1	1	0	0,004	0,035	0,031
107	PRIMethod	1920	2E-07	3E-06	1	187,3	102,7	1	1	0	3E-06	2E-05	2E-05
108	BHMethod	1920	2E-07	3E-06	1	187,3	102,8	1	1	0	3E-06	2E-05	7E-06
109	IQMethod	1920	1E-04	3,06	1	187,3	43,27	1	1	0	1E-06	3E-05	0,022
110	LSB8bMethod	2E+06	0,053	106,8	0,991	185,7	27,84	0,992	0,992	0	0,007	0,04	2,855
111	LSBMethod	4E+06	4E-04	0,007	1	187,3	69,98	1	1	0	0,004	0,035	0,031
112	PRIMethod	1920	2E-07	3E-06	1	187,3	102,8	1	1	0	3E-06	2E-05	2E-05
113	BHMethod	1920	2E-07	3E-06	1	187,3	102,8	1	1	0	2E-06	2E-05	2E-05
114	IQMethod	1920	1E-04	3,3	1	187,3	42,95	1	1	0	2E-05	2E-05	0,022
115	LSB8bMethod	4E+06	0,053	106,8	0,991	185,6	27,84	0,992	0,992	0	0,008	0,04	2,813
116	LSBMethod	2E+06	4E-04	0,007	1	187,3	69,98	1	1	0	0,004	0,036	0,03
117	PRIMethod	1920	2E-07	3E-06	1	187,3	102,9	1	1	0	3E-06	2E-05	7E-06
118	BHMethod	1920	2E-07	3E-06	1	187,3	102,7	1	1	0	3E-06	2E-05	8E-06
119	IQMethod	1920	1E-04	3,186	1	187,3	43,1	1	1	0	5E-06	3E-05	0,023
120	LSB8bMethod	2E+06	0,053	107	0,991	185,6	27,84	0,992	0,992	0	0,007	0,04	2,83
121	LSBMethod	4E+05	2E-04	0,007	1	246,1	69,88	1	1	0	2E-05	0,001	0,027
122	PRIMethod	825	3E-07	8E-06	1	246,1	98,9	1	1	0	0	2E-06	6E-05
123	BHMethod	825	3E-07	9E-06	1	246,1	98,82	1	1	0	0	3E-06	0

Продовження таблиці Г.1

124	IQMethod	825	7E-04	40,72	0,999	245,9	32,03	0,987	0,986	1,333	2E-06	1E-05	0,159
125	LSB8bMethod	4E+05	0,032	111,4	0,97	238,6	27,66	0,979	0,978	0	2E-04	0,27	1,582
126	LSBMethod	8E+05	3E-04	0,006	1	224,5	70	1	1	0	1E-03	3E-04	0,006
127	PRIMethod	1019	3E-07	6E-06	1	224,5	100,3	1	1	0	7E-07	9E-08	5E-06
128	BHMethod	1019	3E-07	6E-06	1	224,5	100,1	1	1	0	0	1E-07	1E-05
129	IQMethod	1019	3E-07	6E-06	1	224,5	100,1	1	1	0	0	3E-08	1E-04
130	LSB8bMethod	8E+05	0,036	110,8	0,967	217,1	27,69	0,987	0,987	0	0,007	0,286	4,548
131	LSBMethod	7E+05	3E-04	0,007	1	213,6	69,99	1	1	0	9E-06	3E-04	9E-05
132	PRIMethod	1024	3E-07	6E-06	1	213,6	100,3	1	1	0	1E-07	3E-07	0
133	BHMethod	1024	3E-07	6E-06	1	213,6	100,1	1	1	0	0	2E-07	0
134	IQMethod	1024	3E-07	7E-06	1	213,6	99,95	1	1	0	0	2E-06	0
135	LSB8bMethod	7E+05	0,037	111	0,966	206,4	27,68	0,97	0,97	0,667	0,005	0,292	2,895
136	LSBMethod	1E+05	3E-04	0,006	1	197,6	70,02	1	1	0	0	4E-04	0,024
137	PRIMethod	456	6E-07	1E-05	1	197,6	97,03	1	1	0	0	9E-07	6E-05
138	BHMethod	456	7E-07	1E-05	1	197,6	96,56	1	1	0	0	1E-07	0
139	IQMethod	456	7E-07	2E-05	1	197,6	96,31	1	1	0	0	1E-05	0
140	LSB8bMethod	1E+05	0,04	109,1	0,973	192,2	27,75	0,975	0,975	12	0	0,284	13,03
141	LSBMethod	8E+05	3E-04	0,006	1	172,7	70	1	1	0	3E-04	8E-04	0,004
142	PRIMethod	1024	4E-07	7E-06	1	172,7	99,95	1	1	0	0	2E-07	0
143	BHMethod	1024	3E-07	6E-06	1	172,7	100,1	1	1	0	0	2E-08	3E-06
144	IQMethod	1024	4E-07	7E-06	1	172,7	99,9	1	1	0	0	2E-06	1E-05
145	LSB8bMethod	8E+05	0,046	110	0,964	166,6	27,72	0,969	0,969	0	0,004	0,345	8,782
146	LSBMethod	7E+05	2E-04	0,007	1	250,5	69,81	1	1	0	2E-04	0,006	0,002
147	PRIMethod	972	2E-07	7E-06	1	250,6	99,81	1	1	0	1E-07	5E-06	4E-06
148	BHMethod	972	3E-07	7E-06	1	250,6	99,66	1	1	0	0	4E-07	1E-04

Продовження таблиці Г.1

149	IQMethod	972	3E-07	7E-06	1	250,6	99,69	1	1	0	0	3E-06	5E-04
150	LSB8bMethod	7E+05	0,031	111,6	0,97	243	27,65	0,983	0,983	0	0,011	0,256	0,47
151	LSBMethod	1E+06	2E-04	0,007	1	246,5	69,86	1	1	0	0	0,008	0,075
152	PRIMethod	1083	2E-07	6E-06	1	246,6	100,2	1	1	0	0	9E-06	7E-05
153	BHMethod	1083	2E-07	6E-06	1	246,6	100,5	1	1	0	0	2E-06	0
154	IQMethod	1083	2E-07	6E-06	1	246,6	100,1	1	1	0	0	2E-06	5E-04
155	LSB8bMethod	1E+06	0,032	111,6	0,969	239	27,65	0,986	0,986	0	4E-04	0,278	1,806
156	LSBMethod	2E+06	2E-04	0,007	1	245,3	69,9	1	1	0	7E-05	0,004	0,085
157	PRIMethod	1955	1E-07	3E-06	1	245,4	102,9	1	1	0	5E-08	2E-06	9E-06
158	BHMethod	1955	1E-07	3E-06	1	245,4	103	1	1	0	0	1E-06	2E-04
159	IQMethod	1955	1E-05	0,598	1	245,4	50,37	1	1	0	0	9E-07	0,002
160	LSB8bMethod	2E+06	0,032	111,6	0,969	237,8	27,65	0,983	0,982	0	0,023	0,251	0,561
161	LSBMethod	4E+06	6E-04	0,006	1	252,6	70,07	1	1	0	3E-07	0,302	0,299
162	PRIMethod	2200	3E-07	3E-06	1	252,7	103,5	1	1	0	0	2E-04	1E-04
163	BHMethod	2200	3E-07	3E-06	1	252,7	103,6	1	1	0	0	8E-07	0
164	IQMethod	2200	7E-04	15,09	0,999	252,5	36,34	0,999	0,999	0	0	0	0,059
165	LSB8bMethod	4E+06	0,08	105	0,97	245,1	27,92	0,996	0,996	0	0,014	0,394	38,13
166	LSBMethod	4E+06	3E-04	0,007	1	251,9	69,92	1	1	0	3E-05	0,14	0,141
167	PRIMethod	2200	2E-07	3E-06	1	252	103,3	1	1	0	1E-07	2E-05	2E-05
168	BHMethod	2200	1E-07	3E-06	1	252	103,6	1	1	0	0	9E-07	0
169	IQMethod	2200	4E-04	15,94	1	251,9	36,11	0,999	0,999	0	0	0	0,063
170	LSB8bMethod	4E+06	0,043	108,4	0,97	244,5	27,78	0,996	0,996	0	2E-05	0,322	18,43
171	LSBMethod	4E+06	6E-04	0,006	1	252,1	70,06	1	1	0	3E-07	0,296	0,298
172	PRIMethod	2200	3E-07	3E-06	1	252,2	103,6	1	1	0	0	1E-04	1E-04
173	BHMethod	2200	3E-07	3E-06	1	252,2	103,6	1	1	0	0	9E-07	0

Продовження таблиці Г.1

174	IQMethod	2200	6E-04	15,19	0,999	252	36,31	0,999	0,999	0	0	0	0,06
175	LSB8bMethod	4E+06	0,077	105,2	0,97	244,6	27,91	0,996	0,996	0	0,016	0,391	37,28
176	LSBMethod	8E+05	7E-04	0,007	1	95,89	69,89	1	1	0	3E-06	0,007	0,092
177	PRIMethod	1125	6E-07	6E-06	1	95,92	100,4	1	1	0	2E-09	5E-06	1E-04
178	BHMethod	1125	6E-07	6E-06	1	95,92	100,6	1	1	0	0	6E-06	2E-04
179	IQMethod	1125	6E-07	6E-06	1	95,92	100,5	1	1	0	0	7E-06	5E-04
180	LSB8bMethod	8E+05	0,087	102,1	1,041	99,82	28,04	0,968	0,967	0	3E-04	0,307	54,7
181	LSBMethod	6E+05	0,002	0,007	1	101,6	70	1	1	0	0,004	0,01	0,045
182	PRIMethod	970	2E-06	7E-06	1	101,6	99,8	1	1	0	5E-06	3E-05	8E-05
183	BHMethod	970	2E-06	7E-06	1	101,6	99,82	1	1	0	1E-05	9E-06	7E-05
184	IQMethod	970	2E-06	7E-06	1	101,6	99,9	1	1	0	1E-05	6E-06	1E-04
185	LSB8bMethod	6E+05	0,186	101,8	1,027	104,3	28,06	0,975	0,975	0	0,041	0,262	52,65
186	LSBMethod	3E+05	5E-04	0,007	1	144,5	70	1	1	0	6E-04	4E-06	0,008
187	PRIMethod	709	7E-07	9E-06	1	144,5	98,56	1	1	0	3E-06	1E-07	5E-05
188	BHMethod	709	7E-07	9E-06	1	144,5	98,47	1	1	0	1E-05	4E-07	1E-05
189	IQMethod	709	7E-07	9E-06	1	144,5	98,46	1	1	0	8E-06	2E-06	5E-04
190	LSB8bMethod	3E+05	0,063	105,8	0,984	142,2	27,89	0,983	0,983	0	0,032	0,268	30,56
191	LSBMethod	7E+05	5E-04	0,007	1	149	70	1	1	0	7E-05	3E-05	0,013
192	PRIMethod	1024	4E-07	6E-06	1	149	100,2	1	1	0	9E-09	9E-08	2E-05
193	BHMethod	1024	4E-07	6E-06	1	149	100,3	1	1	0	7E-08	2E-07	2E-05
194	IQMethod	1024	5E-07	7E-06	1	149	99,89	1	1	0	3E-09	1E-06	3E-04
195	LSB8bMethod	7E+05	0,06	106,9	0,983	146,5	27,84	0,983	0,983	0	0,087	0,126	10,15
196	LSBMethod	5E+06	4E-04	0,007	1	158,7	69,99	1	1	0	1E-03	4E-05	0,054
197	PRIMethod	2560	2E-07	3E-06	1	158,7	104	1	1	0	3E-08	3E-08	2E-05
198	BHMethod	2560	2E-07	3E-06	1	158,7	104	1	1	0	1E-06	3E-07	2E-05

Продовження таблиці Г.1

199	IQMethod	2560	6E-06	0,023	1	158,7	64,58	1	1	0	4E-07	1E-05	0,002
200	LSB8bMethod	5E+06	0,054	109	0,976	154,9	27,76	0,982	0,982	0	0,147	0,059	26,68
201	LSBMethod	1E+07	5E-04	0,006	1	164,2	70,02	1	1	0	0,004	0,011	0,001
202	PRIMethod	4560	1E-07	1E-06	1	164,2	106,7	1	1	0	2E-06	3E-06	6E-07
203	BHMethod	4560	1E-07	1E-06	1	164,2	106,6	1	1	0	3E-07	8E-08	9E-07
204	IQMethod	4560	1E-07	1E-06	1	164,2	106,5	1	1	0	2E-07	2E-07	5E-05
205	LSB8bMethod	1E+07	0,061	107	0,97	159,3	27,84	0,989	0,989	0	0,08	0,126	15,99
206	LSBMethod	2E+07	5E-04	0,006	1	138	70	1	1	0	2E-05	2E-04	0,01
207	PRIMethod	6675	8E-08	1E-06	1	138	108	1	1	0	4E-08	5E-08	2E-06
208	BHMethod	6675	7E-08	1E-06	1	138	108,3	1	1	0	2E-07	3E-09	3E-06
209	IQMethod	6675	7E-08	9E-07	1	138	108,6	1	1	0	2E-07	9E-08	5E-05
210	LSB8bMethod	2E+07	0,065	108,1	0,966	133,2	27,79	0,979	0,979	0	0,034	0,136	3,594
211	LSBMethod	7E+05	6E-04	0,007	1	148,6	70	1	1	0	6E-04	6E-05	0,007
212	PRIMethod	1024	6E-07	6E-06	1	148,6	100,1	1	1	0	0	2E-08	6E-06
213	BHMethod	1024	5E-07	6E-06	1	148,6	100,1	1	1	0	7E-08	3E-07	1E-05
214	IQMethod	1024	6E-07	7E-06	1	148,6	99,81	1	1	0	4E-06	3E-06	1E-04
215	LSB8bMethod	7E+05	0,07	104,6	0,988	146,8	27,94	0,988	0,988	0	0,026	0,189	31,94
216	LSBMethod	6E+05	6E-04	0,006	1	116,4	70	1	1	0	1E-04	3E-04	8E-05
217	PRIMethod	910	6E-07	7E-06	1	116,4	99,84	1	1	0	3E-07	6E-07	1E-05
218	BHMethod	910	7E-07	7E-06	1	116,4	99,61	1	1	0	5E-06	2E-07	4E-06
219	IQMethod	910	7E-07	8E-06	1	116,4	99,38	1	1	0	1E-05	8E-06	4E-05
220	LSB8bMethod	6E+05	0,077	103,1	1,015	118,2	28	0,979	0,978	0	0,014	0,236	42,7
221	LSBMethod	3E+06	9E-04	0,006	1	94,64	70,02	1	1	0	0,002	7E-04	0,001
222	PRIMethod	2048	5E-07	3E-06	1	94,64	103,2	1	1	0	4E-06	1E-07	4E-07
223	BHMethod	2048	5E-07	3E-06	1	94,64	103,1	1	1	0	1E-06	8E-07	4E-06

Продовження таблиці Г.1

224	IQMethod	2048	5E-07	3E-06	1	94,64	103,2	1	1	0	3E-06	5E-06	2E-05
225	LSB8bMethod	3E+06	0,115	101,5	1,052	99,57	28,06	0,98	0,98	0	0,173	0,067	41,46
226	LSBMethod	7E+05	5E-04	0,006	1	151,1	70,02	1	1	0	7E-04	5E-04	0,007
227	PRIMethod	1024	5E-07	6E-06	1	151,1	100,1	1	1	0	3E-07	7E-08	1E-05
228	BHMethod	1024	5E-07	7E-06	1	151,1	99,96	1	1	0	5E-06	3E-07	7E-06
229	IQMethod	1024	5E-07	7E-06	1	151,1	99,83	1	1	0	2E-06	5E-06	5E-05
230	LSB8bMethod	7E+05	0,061	105,6	0,987	149,1	27,9	0,985	0,985	0	0,023	0,028	11,31
231	LSBMethod	7E+05	5E-04	0,007	1	138,3	69,99	1	1	0	0,003	0,009	0,012
232	PRIMethod	1024	5E-07	6E-06	1	138,3	100,2	1	1	0	6E-08	2E-05	3E-06
233	BHMethod	1024	5E-07	6E-06	1	138,3	100,3	1	1	0	1E-05	2E-05	2E-05
234	IQMethod	1024	5E-07	6E-06	1	138,3	100,3	1	1	0	8E-06	7E-06	1E-04
235	LSB8bMethod	7E+05	0,066	105,6	0,991	137,1	27,9	0,982	0,982	0	0,015	0,035	10,38
236	LSBMethod	6E+05	5E-04	0,007	1	145,7	69,99	1	1	0	4E-04	5E-05	0,006
237	PRIMethod	910	5E-07	7E-06	1	145,8	99,68	1	1	0	0	2E-07	4E-05
238	BHMethod	910	5E-07	7E-06	1	145,8	99,66	1	1	0	9E-06	1E-06	2E-05
239	IQMethod	910	6E-07	7E-06	1	145,8	99,5	1	1	0	4E-07	9E-06	1E-04
240	LSB8bMethod	6E+05	0,062	105	0,991	144,5	27,92	0,984	0,984	0	0,008	0,214	29,45
241	LSBMethod	1E+06	8E-04	0,006	1	127,6	70,01	1	1	0	0,004	4E-04	0,012
242	PRIMethod	1365	6E-07	5E-06	1	127,6	101,4	1	1	0	5E-06	5E-07	2E-05
243	BHMethod	1365	6E-07	5E-06	1	127,6	101,5	1	1	0	6E-05	1E-07	0
244	IQMethod	1365	6E-07	5E-06	1	127,6	101,4	1	1	0	6E-06	1E-06	8E-06
245	LSB8bMethod	1E+06	0,097	102,6	1,011	129	28,02	0,988	0,988	0	0,309	0,202	39,66
246	LSBMethod	1E+07	4E-04	0,006	1	175,8	70,01	1	1	0	0	0,004	0,244
247	PRIMethod	4333	8E-08	1E-06	1	175,8	106,4	1	1	0	0	7E-07	6E-05
248	BHMethod	4333	8E-08	1E-06	1	175,8	106,5	1	1	0	0	6E-07	6E-05

Продовження таблиці Г.1

249	IQMethod	4333	8E-08	2E-06	1	175,8	106,4	1	1	0	0	6E-07	1E-04
250	LSB8bMethod	1E+07	0,047	108,2	0,973	171,1	27,79	0,981	0,981	0	0	0,336	20,78
251	LSBMethod	1E+06	6E-04	0,006	1	130,2	70,01	1	1	0	3E-04	2E-04	7E-04
252	PRIMethod	1365	4E-07	5E-06	1	130,2	101,5	1	1	0	8E-06	5E-07	8E-07
253	BHMethod	1365	4E-07	5E-06	1	130,2	101,6	1	1	0	0	4E-07	0
254	IQMethod	1365	4E-07	5E-06	1	130,2	101,6	1	1	0	0	2E-06	0
255	LSB8bMethod	1E+06	0,072	101,8	1,035	134,8	28,05	0,988	0,988	0	0,165	0,046	30,24
256	LSBMethod	2E+06	5E-04	0,007	1	163,8	70	1	1	0	5E-05	0,003	0,016
257	PRIMethod	1752	3E-07	4E-06	1	163,8	102,6	1	1	0	2E-06	2E-07	2E-05
258	BHMethod	1752	3E-07	4E-06	1	163,8	102,4	1	1	0	1E-06	3E-08	7E-06
259	IQMethod	1752	3E-07	4E-06	1	163,8	102,3	1	1	0	4E-07	8E-07	2E-04
260	LSB8bMethod	2E+06	0,063	105,4	0,984	161,2	27,9	0,99	0,99	0	0,052	0,207	25,6
261	LSBMethod	1E+07	3E-04	0,006	1	221,9	70,02	1	1	0	8E-06	2E-04	0,03
262	PRIMethod	3840	7E-08	2E-06	1	221,9	105,9	1	1	0	0	1E-08	9E-07
263	BHMethod	3840	7E-08	2E-06	1	221,9	105,9	1	1	0	0	9E-09	2E-06
264	IQMethod	3840	1E-07	2E-04	1	221,9	84,69	1	1	3,333	0	8E-07	1E-04
265	LSB8bMethod	1E+07	0,036	111,1	0,968	214,7	27,67	0,982	0,982	3,333	0,006	0,287	2,196
266	LSBMethod	6E+05	8E-04	0,006	1	95,81	70,01	1	1	0,333	0,002	0,003	0,004
267	PRIMethod	910	1E-06	8E-06	1	95,81	99,28	1	1	0	7E-06	2E-08	0
268	BHMethod	910	9E-07	7E-06	1	95,81	99,56	1	1	0	0	6E-06	2E-05
269	IQMethod	910	9E-07	7E-06	1	95,81	99,67	1	1	0	0	2E-05	7E-05
270	LSB8bMethod	6E+05	0,101	101,9	1,053	100,9	28,05	0,979	0,978	2,333	0,052	0,095	42,48
271	LSBMethod	6E+06	4E-04	0,006	1	159,3	70	1	1	0	6E-04	4E-04	0,023
272	PRIMethod	3000	1E-07	2E-06	1	159,3	104,9	1	1	0	1E-07	8E-08	4E-06
273	BHMethod	3000	1E-07	2E-06	1	159,3	104,6	1	1	0	0	3E-07	0

Продовження таблиці Г.1

274	IQMethod	3000	1E-07	2E-06	1	159,3	104,6	1	1	0	0	1E-06	0
275	LSB8bMethod	6E+06	0,054	108,6	0,965	153,8	27,77	0,981	0,981	0	0,062	0,336	15,83
276	LSBMethod	50299	4E-04	0,007	1	171,1	69,96	1	1	0	0	0,018	0,249
277	PRIMethod	281	2E-06	3E-05	1	171,1	94,01	1	1	0	0	4E-05	0,001
278	BHMethod	281	2E-06	2E-05	1	171,1	94,19	1	1	0	0	9E-06	9E-04
279	IQMethod	281	1E-06	2E-05	1	171,1	94,56	1	1	0	0	8E-06	0,002
280	LSB8bMethod	50299	0,055	107,7	0,973	166,5	27,81	0,989	0,989	0	0	0,354	26,73
281	LSBMethod	2E+07	4E-04	0,007	1	162,5	70	1	1	0,333	4E-04	5E-05	0,026
282	PRIMethod	5184	8E-08	1E-06	1	162,5	107,1	1	1	0	1E-06	8E-08	2E-06
283	BHMethod	5184	8E-08	1E-06	1	162,5	107,1	1	1	0	0	7E-10	1E-06
284	IQMethod	5184	8E-08	1E-06	1	162,5	107,1	1	1	0	0	3E-07	9E-05
285	LSB8bMethod	2E+07	0,053	107,5	0,975	158,5	27,82	0,982	0,982	1	0,099	0,202	16,81
286	LSBMethod	2E+06	4E-04	0,007	1	163,1	70	1	1	0	0	0,004	0,228
287	PRIMethod	1600	3E-07	4E-06	1	163,1	101,8	1	1	0	0	3E-06	2E-04
288	BHMethod	1600	3E-07	4E-06	1	163,1	101,8	1	1	0	0	3E-06	2E-04
289	IQMethod	1600	3E-07	4E-06	1	163,1	101,9	1	1	0	0	2E-06	3E-04
290	LSB8bMethod	2E+06	0,05	107,9	0,975	159	27,8	0,975	0,975	0	0	0,348	22,4
291	LSBMethod	3E+05	0,001	0,006	1	129,8	70	1	1	0	0,005	9E-05	0,009
292	PRIMethod	615	2E-06	1E-05	1	129,8	97,47	1	1	0	2E-05	3E-07	2E-05
293	BHMethod	615	2E-06	1E-05	1	129,8	97,6	1	1	0	1E-04	3E-08	2E-05
294	IQMethod	615	2E-06	1E-05	1	129,8	97,68	1	1	0	1E-05	1E-05	4E-04
295	LSB8bMethod	3E+05	0,129	103,2	0,978	126,9	28	0,986	0,986	0	0,192	0,194	45,88
296	LSBMethod	7E+06	8E-04	0,006	1	116,3	70	1	1	0,333	6E-04	6E-04	0,004
297	PRIMethod	2985	3E-07	2E-06	1	116,3	104,6	1	1	0	3E-07	2E-07	1E-07
298	BHMethod	2985	3E-07	2E-06	1	116,3	104,7	1	1	0	0	2E-08	6E-07

Продовження таблиці Г.1

299	IQMethod	2985	3E-07	2E-06	1	116,3	104,7	1	1	0	0	7E-07	2E-05
300	LSB8bMethod	7E+06	0,093	102,1	1,029	119,7	28,04	0,986	0,985	1	0,096	0,205	45,79