Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Матрос Вячеслав Ігорович,
студент групи РЗ-171

# КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Удосконалення методу виявлення та локалізації областей клонування в цифрових зображеннях

Спеціальність:
125 Кібербезпека

Спеціалізація, освітня програма:
Кібербезпека

Керівник:
Лебедєва Олена Юріївна,
к.т.н., доцент

Одеса – 2022

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Рівень вищої освіти другий (магістерський)
Спеціальність 125 Кібербезпека
Спеціалізація, освітня програма Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри КБПЗ
_____
д.т.н.,проф. А.А.Кобозєва
____ _____2022р.

**ЗАВДАННЯ**
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

*Матросу Вячеславу Ігоровичу*

1. Тема роботи: *Удосконалення методу виявлення та локалізації областей клонування в цифрових зображеннях*, керівник роботи *Лебедєва Олена Юріївна, к. т. н., доцент,*

затверджені наказом ректора від „____”_____ 2022 р. №_____ .

2. Зміст роботи*: аналіз проблемної області, постановка задачі, удосконалення методу виявлення та локалізації областей клонування в цифрових зображеннях шляхом використання складних блоків та маркерів, програмна реалізувація удосконалений метод виявлення та локалізації областей клонування в цифрових зображеннях та оцінити її ефективність.*

3. Перелік ілюстративного матеріалу: *схеми складних блоків, рисунки інтерфейсу програмної реалізації, слайди презентації.*

4. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | Завдання видав | Завдання прийняв |
| | | | |

5. Дата видачі завдання "_____"_____2022 р.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання | Примітка |
|---|---|---|---|
| *1* | *Аналіз джерел з теми випускної кваліфікаційної роботи* | *30.07.2022* | *виконано* |
| *2* | *Обґрунтування вибору рішення. Збір даних* | *31.08.2022* | *виконано* |
| *3* | *Аналіз методу виявлення та локалізації областей клонування в цифрових зображеннях* | *30.09.2022* | *виконано* |
| *4* | *Вибір виду складних блоків для виявлення областей клонування* | *16.10.2022* | *виконано* |
| *5* | *Удосконалення методу виявлення та локалізації областей клонування цифрових зображень* | *30.10.2022* | *виконано* |
| *6* | *Підготовка тексту роботи* | *30.11.2022* | *виконано* |
| *7* | *Підготовка презентації та доповіді* | *06.2022* | *виконано* |
| *8* | *Попередній захист* | *07.12.22* | *виконано* |
| *9* | *Нормоконтроль, рецензування* | *24.12.2022* | *виконано* |
| *10* | *Занесення роботи в електронний архів* | *25.12.2022* | *виконано* |
| *11* | *Допуск до захисту у завідувача кафедри* | | *виконано* |

**Здобувач вищої освіти** _____ *Матрос В.І.*

**Керівник роботи** _____ *Лебедєва О.Ю.*

# АНОТАЦІЯ

Кваліфікаційна робота на тему “Удосконалення методу виявлення та локалізації областей клонування в цифрових зображеннях” на здобуття другого (магістерського) рівня вищої освіти за спеціальністю 125 Кібербезпека, спеціалізація, освітня програма: Кібербезпека, містить 25 рисунків, 7 таблиць, 9 формул, 32 літературних джерела за переліком посилань. Робота виконана на сторінках 47 загального тексту і 43 сторінках основного тексту.

Метою роботи є удосконалення методу виявлення та локалізації областей клонування в цифрових зображеннях шляхом використання блоків складної форми.

У роботі проведено аналіз сучасних рішень виявлення та локалізації областей клонування в цифрових зображеннях.

У результаті виконання кваліфікаційної роботи розроблено проект виявлення та локалізації областей клонування в цифрових зображеннях за удосконаленим алгоритмом.

Результати даної роботи можуть бути використані спеціалістами з цифрової криміналістики та правоохоронними органи для підтвердження оригінальності цифрового зображення.

ІНФОРМАЦІЯ, ІНФОРМАЦІЙНА БЕЗПЕКА, СИСТЕМА ВИЯВЛЕННЯ ТА ЛОКАЛІЗАЦІЇ ОБЛАСТЕЙ КЛОНУВАННЯ В ЦИФРОВИХ ЗОБРАЖЕННЯХ, ЦИФРОВІ ЗОБРАЖЕННЯ, ФАЛЬСИФІКАЦІЯ ЦИФРОВИХ ЗОБРАЖЕНЬ.

# ANOTATION

Qualification work on the topic "Improving the method of detection and localization of cloning areas in digital images" for obtaining the second (master's) level of higher education in the specialty 125 Cybersecurity, specialization, educational program: Cybersecurity, contains 25 figures, 7 tables, 9 formulas, 32 literary sources by reference list. The work was completed on 47 pages of the general text and 43 pages of the main text.

The aim of the work is to improve the method of detection and localization of cloning areas in digital images by using blocks of complex shape.

The paper analyzes modern solutions for detecting and localizing cloning areas in digital images.

As a result of the qualification work, a project was developed to detect and localize areas of cloning in digital images using an improved algorithm.

The results of this work can be used by digital forensics specialists and law enforcement agencies to confirm the originality of a digital image.

INFORMATION, INFORMATION SECURITY, SYSTEMS FOR DETECTION AND LOCATION OF CLONING AREAS IN DIGITAL IMAGES, DIGITAL IMAGES, FORGERY OF DIGITAL IMAGES.

ЗМІСТ

# ВСТУП

Разом із новою науково-технічною революцією у 21 столітті сучасні технології почали проникати майже до усіх сфер людської діяльності та перехід до цифрових технологій.

В Україні будь яке цифрове зображення може виступати у якості доказів (відповідно до ч. 1 статті 100 § 5 Глави 5 Розділу 1 ЦПК України) [1].

Більша частина громадян Сполучених Штатів Америки та Європейського союзу дізнаються про новини з інтернету. Згідно дослідженням Pew Research Center 2020 року, 86% дорослих у Сполучених Штатах Америки дізнаються про новини з інтернету «часто» чи «іноді». З них 60% «часто» звертаються до інтернет джерел новин [2].

Дослідження ж Eurostat за 2021 рік каже, що 72% населення Європейського союзу дізнаються про новини за інтернету [3].

У сервісі Google Images на сьогоднішній день можна знайти більше 136 мільярдів фотографій. Кожен рік по всьому світу роблять приблизно 1,72 трильйони фотографій [4]. Кожен день користувачі інтернету діляться між собою 3,2 мільярдами зображень [5]. Через це в інтернеті легко зіткнутися з фальсифікованими зображеннями, навіть у довірених новинних виданнях, оскільки у вільному доступі знаходяться такі графічні редактори, як Adobe Photoshop, Figma, GIMP та Photo Pos Pro.

У наш час цифрові зображення можуть бути просто змінені за допомогою найкращих комп’ютерів, тонкого редагування фотографій, тощо. Ці зміни вплинуть на достовірність зображень у законодавстві, політиці, ЗМІ, промисловості та медицині.

Фальсифікація – це маніпуляції з цифровим зображенням, щоб приховати певну значущу або корисну інформацію зображення [6]. Виявлення фальсифікації цифрових зображень є однією з найбільш критичних аналітичних

практик у наш час. Через це зростає необхідність у ефективних методах виявлення фальсифікованого зображення.

Однією з найчастіше використовуваних операцій під час фальсифікації цифрових зображень, реалізованої у всіх графічних редакторах, є операція клонування, у ході якої відбувається заміна частини (частин) цифрового зображення, частиною (частинами) того ж цифрового зображення.

Для підвищення ефективності боротьби з фальсифікацією цифрових зображень, створених методом клонування, доцільно буде удосконалити вже існуючи методи виявлення та локалізації областей клонування в цифрових зображеннях. Тому тема кваліфікаційної роботи є актуальною.

Метою даної кваліфікаційної роботи є удосконалення методу виявлення та локалізації областей клонування в цифрових зображеннях шляхом використання складних блоків та маркерів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. огляд методів та засобів виявлення та локалізації областей клонування в цифрових зображеннях;
2. вибір виду складних блоків для виявлення областей клонування;
3. удосконалення методу виявлення та локалізації областей клонування в цифрових зображеннях;
4. програмно реалізувати удосконалений метод виявлення та локалізації областей клонування в цифрових зображеннях та оцінити її ефективність.

Під ефективністю в цій роботі будемо розуміти зменшення часу роботи методу виявлення та локалізації областей клонування в цифрових зображеннях та збільшення точності виявлення області клонування.

Об’єктом дослідження є процеси несанкціонованої зміни цифрових зображень.

Предметом дослідження є виявлення та локалізація областей клонування в цифрових зображеннях.

Наукова новизна полягає в удосконаленні методу виявлення та локалізація областей клонування в цифрових зображеннях шляхом використання блоків та маркерів, що призвело до зменшення часу, необхідного для виявлення результатів клонування цифрових зображень.

Практична цінність роботи полягає у програмній реалізації розробленого удосконалення методу виявлення та локалізація областей клонування в цифрових зображеннях, яка може бути використана спеціалістами з цифрової криміналістики та правоохоронними органи для підтвердження оригінальності цифрового зображення.

# 1 АНАЛІЗ ДОСЛІДЖЕНЬ МЕТОДІВ ТА ЗАСОБІВ ПЕРЕВІРКИ ТА ВИЯВЛЕННЯ ФАЛЬСИФІКОВАННОГО ЗОБРАЖЕННЯ

## 1.1 Способи фальсифікації зображення

Існує два типу модифікації зображення, які поділяються на декілька підтипів: модифікація для захисту, та модифікація для фальсифікації [7].

До методів захисту зображення належать цифровий водяний знак та цифровий підпис.

Існують різні методи фальсифікації зображення. З огляду на техніку, що використовується для зміни зображень, існує три типи підробки цифрових зображень: зрощування зображень або фото колаж, копіювання-переміщення або підробка областей дублюванням і ретушування зображень [8].

Під час фото колажу два або більше зображень або частини зображень використовуються для створення з’єднаного (складеного) зображення [9]. Приклад фото колажу можна побачити на рисунку 1.1 та 1.2.

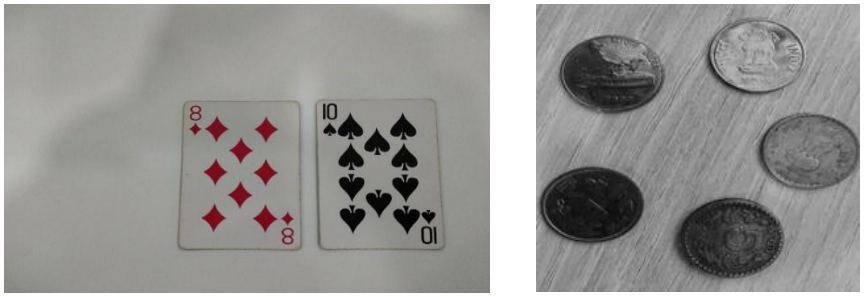


Рисунок 1.1 – Оригінальні зображення

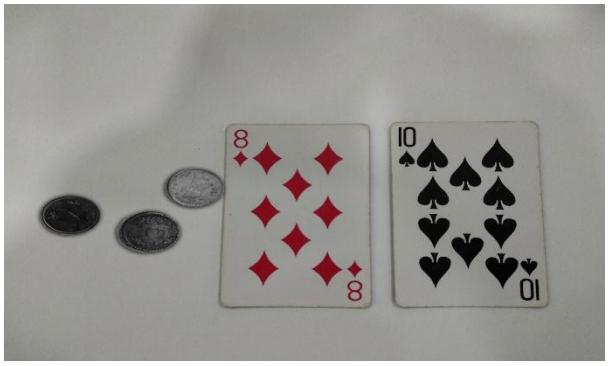


Рисунок 1.2 – Фото колаж

Характерними ознаками зрощення зображень для людського ока є неправильне місцезнаходження тіней на об’єктах відносно джерел світла, різний рівень освітлення об’єктів, різний відтінок кольору шкіри (при накладенні обличчя однієї людини замість обличчя іншої), різкі переходи від об’єкту до фону, при незграбному редагуванню зображень можна побачити кути замість плавних ліній силуету об’єкта, неправильна форма, через спробу надати об’єкту правильне положення у кадрі, чи невидалений фон, який залишився з оригінального зображення, з якого вирізався об’єкт, та інше.

При підробці областей дублюванням одна частина зображення копіюється та вставляється в інше місце того самого зображення, щоб приховати інформацію або змінити значення зображення(рисунок 1.3). Отже, між оригінальною та дубльованою областю існує сильна кореляція, яку можна використовувати як доказ для виявлення підробки копіювання-переміщення [109].

Для людського ока характерними ознаками копіювання-переміщення є неправильне місцезнаходження тіней на об’єктах відносно джерел світла, різний рівень освітлення об’єктів, розходження текстур фону (при приховуванні об’єкту), різкі переходи від об’єкту до фону та інше.

Рисунок 1.3 – Оригінальне зображення (ліворуч) та модифіковане методом копіювання-переміщення (праворуч)

Виявлення копіювання-переміщення може бути ускладнене:

– Зображення може бути збережено у форматі зі стисненням із втратами.

– До зображення може бути доданий шум, що ускладнить виявлення підробки.

– Область можуть обертати чи віддзеркалювати при підробці.

– Скопійована область може бути розмитою.

– Текстура скопійованих областей може бути змінена. Її можна зробити світлішою або темнішою.

Під класифікацію ретушування входить коригування зображення за допомогою будь-якого програмного забезпечення для досягнення певного результату, наприклад, щоб покращити зображення. Ця процедура принципово не змінює зображення, а скоріше покращує якість або зменшує певний елемент зображення. Наприклад існують нейронні мережі, які збільшують роздільну якість зображення та видаляють артефакти JPEG, проте трохи розмивають саме зображення. Однією з таких нейронних мереж є waifu2x. Приклад її роботи можна побачити на рисунку 1.4, де оригінальне зображення з роздільною якістю 188x200 порівнюється з обробленим з параметрами налаштування 2x збільшення та Дуже сильне видалення артефактів JPEG з роздільною якістю 376x400.

Рисунок 1.4 – Оригінальне зображення (ліворуч) та оброблене нейронною мережею (праворуч)

Щоб створити дивовижне підроблене зображення, деякі обрані локалі мають зазнати геометричних змін, таких як обертання, масштабування, розширення тощо. Вступний крок відіграє важливу роль у процесі ретушування та представляє незначні фактичні зміни. Ретушування привносить в зображення чіткі переривчасті зв'язки. Ці зв’язки можуть бути використані для сприйняття підробки, яка здійснюється шляхом ретуші. Незалежно від того, яка камера використовується для зйомки, можливо змінити кожну фотографію, щоб пізніше позбутися будь-яких дефектів. Ретуш включає в себе багато процедур, як суттєве коригування тіні, модифікація шкіри, відновлення фотографії тощо [11**Ошибка! Источник ссылки не найден.**]. Приклад відновлення фотографій можна побачити на рисунку 1.5.

Рисунок 1.5 – Оригінальне зображення (ліворуч) та відновлене (праворуч)

Ретушування серед усіх методів складніше за все виявити людським оком. Серед характерних ознак можна виділити артефакти JPEG, при збереженні зі значними втратами, та розмиття контурів об’єктів.

На кожен метод фальсифікації зображень існують декілька різних методів їх виявлення, які спираються на різні компоненти, наприклад пікселі або сингулярні числа, різні ускладнення, наприклад фальсифікація без додаткової обробки або зі збереженням у форматі з втратами. Кожен метод виявлення залежить від поставленої задачі: чи то обробити якомога швидше, чи то обробити якомога точніше. Під час кваліфікаційної роботи на здобуття магістерського ступеня буде розглядатись метод виявлення методу фальсифікації копіювання-переміщення без ускладнень.

## 1.2 Методи виявлення фальсифікації зображення, зроблених методом копіювання-переміщення

В даний момент при вирішенні завдань виявлення порушень цілісності цифрових сигналів, зокрема, цифрових зображень, все більшою популярністю користується загальний підхід до аналізу стану та технології та функціонування інформаційної системи, в основі якого лежать такі положення. Будь-яке цифрове зображення можна формалізувати як кінцевого набору двомірних матриць, однозначно визначається сингулярним спектром (спектрами) і набором (наборами) ортонормованих лексикографічно позитивних сингулярних векторів відповідної йому матриці (матриць). Тобто сингулярні числа (СНЧ) та сингулярні вектори несуть у собі всю інформацію про стан цифрового зображення та про будь-які впливи на нього. Фальсифікація, що є перетворення цифрового зображення, формально можна представити як обурення (отже у вигляді сукупності обурень сингулярних чисел і сингулярних векторів) вихідної матриці (множини матриць) цифрового зображення. Обурення сингулярних чисел можна порівняти з величиною впливу, що обурює, характеризують його силу, чого не можна сказати про сингулярні вектори. Таким чином, як набор формальних параметрів, що характеризують цифрове зображення, аналіз яких доцільно використовувати для виявлення порушення його цілісності, далі виступає набір сингулярних чисел [12].

Інші автори пропонують використання коефіцієнту Пірсона значень яскравості пікселів. В даному випадку алгоритм пошуку фальсифікованих ділянок цифрових зображень шляхом клонування ґрунтується на розбитті зображення на безліч блоків, що не перетинаються, і безліч блоків, що перетинаються. Оцінка близькості двох блоків із різних множин здійснюється за допомогою коефіцієнта кореляції Пірсона. Коефіцієнт кореляції обчислюється за значеннями яскравості пікселів блоків. Рівність коефіцієнта кореляції

одиниці свідчить, що два аналізованих блоки подібні, і ми вважаємо їх джерелом і результатом клонування [13].

Ще один спосіб виявлення та локалізації клонування використовує квадратуру Гаусса-Герміта. Для цього зображення переводиться до градацій сірого, після чого проходить розбиття зображення на безліч блоків, що не перетинаються, і безліч блоків, що перетинаються. З кожного блоку отримують моменти Гаусса-Герміта у 1-ій, 3-ій та 5-ій послідовності, які перетворюють на вектор. У результаті, кожен вектор представляє відповідний блок. Якщо знаходяться певні цифри, на одній і тієї ж самій позиції, то блоки вважаються клонованими [14].

Також, деякі автори пропонують використовувати алгоритми SURF та PCET. Спочатку зображення ділиться на блоки неправильної форми, які не перекриваються, а потім блоки поділяються на гладкі області та області текстури. По-друге, детектори SURF з різними порогами контрастності виконуються на гладких областях і областях текстури, щоб отримати достатню кількість точок. Коефіцієнти PCET точок виділяються та використовуються як дескриптори. Запропоновано вдосконалений алгоритм g2NN, який використовується для пошуку подібних ознак, і отримано відповідні точки. По-третє, алгоритм ітерації RANSAC і стратегія фільтрації, яка поєднує матрицю міток, використовуються для усунення помилкових збігів. Ділянки грубого прямокутника виявляються щільними точками. Потім ці прямокутні області поділяються на кругові блоки, що перекриваються, а коефіцієнти PCET витягуються з кругових блоків. Подібні властивості знайдені за допомогою вдосконаленого алгоритму g2NN. Нарешті, математична морфологія та ітераційна стратегія використовуються для визначення місцезнаходження пошкоджених областей [15].

Також, для вирішення проблеми визначення та локалізації дубльованих областей використовують алгоритм Scale Invariant Features Transform (SIFT),

який використовується для надійного виявлення та опису кластерів точок, що належать клонованим областям. Зображення переводять до градацій сірого, після чого знаходять ключові точки цифрового зображення, які характеризуються вектором, що збирає статистику зображення з області, поряд з ключовою точкою. Потім точки фільтрують, щоб відсіяти ті, які не є SIFT ключовими точками. До кожної точки знаходять передбачувані відповідності, щоб дізнатись параметри афінної трансформації. Потім до кожного пікселю відповідної області застосовують вже відомі параметри трансформації. Області клонування знаходяться через обчислення кореляції пікселів областей з накладеною трансформацією та без [16, 17].

Ще один метод використовує алгоритми AKAZE і FAST з автоматичним пороговим значенням контрастності. Для виявлення підроблених зображень, оброблених за допомогою композитних геометричних атак і атак після обробки, запропонована методологія розділена на шість кроків: вилучення ключових точок, обчислення дескрипторів, зіставлення ключових точок, кластеризація ключових точок, видалення викидів і створення кореляційної карти для локалізації підроблених областей. Ці кроки обговорюються в наступних розділах. Під час попередньої обробки вхідне кольорове зображення RGB перетворюється на зображення у градаціях сірого. Техніка FAST стійка до звичайних атак зі спотворенням зображення, але характерні точки, розташовані поблизу меж зображення, залишаються невиявленими через розгляд дуги кола для порівняння значень сусідніх пікселів під час виявлення ключових точок FAST. Для виявлення ключових точок використовується алгоритм AKAZE. Алгоритм AKAZE базується на нелінійній дифузійній фільтрації, яка формулює стабільний нелінійний масштабний простір для вирішення проблеми меж і втрати деталей [18]. Серед усіх наукових робіт можна побачити, що основним вектором напрямку досліджень у сфері виявлення фальсифікованих методом копіювання-переміщення цифрових зображень є не модифікація вже існуючих

алгоритмів, а розробка нових методів виявлення. Основним методом прискорення роботи серед усіх робіт є переведення зображення з кольорової схеми RGB до градацій сірого [1413, 16, 17, 18]. Також значні дослідження модифікації методів виявлення проводяться у напрямку модифікації методу виявлення та локалізації областей клонування в цифрових зображеннях за допомогою кореляції Пірсона. Відповідно до експериментів, проведених Лебедєвою О.Ю. обробка зображення блоками круглої форми була більш точною, ніж при обробці квадратною. Була знайдена значно більша кількість клонованих блоків, в той же час процент помилок був значно меншим. Результати експериментів можна побачити у таблиці 1.1 та таблиці 1.2 [19].

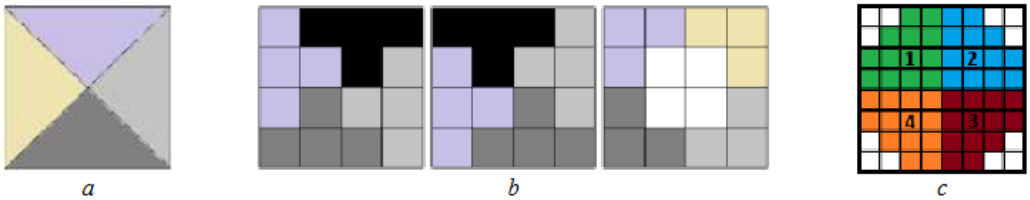
Таблиця 1.1 Точність пошуку клонованих областей різних типів розбивань блоків

| Типи розбивань | Процент знайдених клонованих областей відносно усіх клонованих областей | | |
|---|---|---|---|
| | Максимально | Мінімально | Середнє |
| Квадратні блоки | 69.17 | 32.13 | 47.51 |
| Круглі блоки | 74.69 | 49.69 | 63.42 |

Таблиця 1.2 Точність пошуку помилково визначених клонованими областей порівняно усієї області клонування

| Типи розбивань | Процент помилок |
|---|---|
| Квадратні блоки | 52.49 |
| Типи розбивань | Процент помилок |
| Круглі блоки | 36.58 |

Після цього, Зоріло В.В., Кобозєва А.А. та Лебедєва О.Ю. провели експерименти, де порівняли ефективність використання трикутникових, складних та секторних блоків(рис. 1.6).




*a* *b* *c*

Рисунок 1.6 – Приклади розбиття блоків на a – трикутникові, b – складні, c – круглі, поділенні на секторі.

У результаті проведеного експерименту, який можна побачити у таблиці 1.3, було виявлено, що використання трикутникових та секторних блоків також є ефективним [2019]. Проте, автори пропонують для підвищення точності виявлення області клонування використовувати усі види блоків водночас, що значно підвищує обчислювальну складність и сильно впливає на час обробки.

Таким чином можна зробити висновок, що використовувати прості квадратні блоки не ефективно. Замість них має сенс використовувати складні блоки, приклади яких можна побачити на рисунку 1.7.

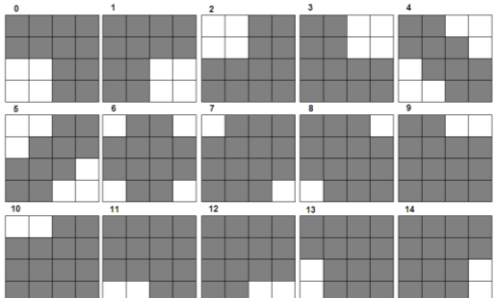


Рисунок 1.7 – Приклад різноманітних складних блоків. Сірим виділені пікселі, які будуть оброблятись.

Таблиця 1.3 – Порівняння точності виявлення клонованих областей при використанні різних блоків

| Зображення | Відносний процент виявленої області клонування від всієї. (%) | | | | |
|---|---|---|---|---|---|
| | Квадратні блоки | Трикутникові блоки | Складні блоки | Секторні блоки | Усі блоки |
| Зображення 1 | 54.62 | 69.60 | 64.23 | 68.99 | 77.58 |

Продовження таблиці 1.3

| Зображення | Відносний процент виявленої області клонування від всієї. (%) | | | | |
|---|---|---|---|---|---|
| | Квадратні блоки | Трикутникові блоки | Складні блоки | Секторні блоки | Усі блоки |

| | | | | | |
|---|---|---|---|---|---|
| Зображення 2 | 69.66 | 82.43 | 75.22 | 79.29 | 90.33 |
| Зображення 3 | 45.31 | 60.86 | 53.73 | 60.16 | 70.84 |
| Зображення 4 | 32.13 | 62.65 | 54.82 | 62.25 | 74.70 |
| Зображення 5 | 45.80 | 69.27 | 59.59 | 71.70 | 82.99 |
| Середнє значення | 49.50 | 68.69 | 61.52 | 68.48 | 79.29 |

Також, Баранов і Лебедєва проводили експеримент з 8x8 квадратними блоками та 16x16 трикутними блоками. Загалом 8x8 блоки використовуються через компроміс між часом обробки зображення та точністю виявлення областей клонування. Хоча зменшення розміру блоку дозволяє уточнювати область фальсифікації, чим більше зменшується розмір блоку, тим більше буде кількість помилок другого роду. Експериментально було виявлено, що квадратні блоки 4x4 роблять 40% помилок другого роду навіть у зображеннях, де фальсифікація відсутня.

Але при порівнянні квадратних блоків 8x8 та трикутникових блоків 16x16 було виявлено, що виявлені області клонування були майже ідентичної форми [2020].

Іншим методом модифікації методу виявлення та локалізації клонованих блоків можна вважати використання маркерів. Оскільки основна обчислювальна важкість полягає у визначенні коефіцієнту кореляції Пірсона двох блоків, автор запропонував у використання кутових пікселів квадратного блоку у якості маркерів, щоб вирішувати, чи потрібно обчислювати коефіцієнт кореляції Пірсона для обраних блоків. Якщо маркери у двох блоків співпадають, то припускається, що маємо справу зі схожими (клонованими та оригінальними) блоками та тільки тоді вираховується коефіцієнт кореляції для остаточного підтвердження цього припущення.

Завдяки даному алгоритму автору вдалося значно підвищити швидкість обробки. Результати експерименту з підвищення ефективності завдяки використання маркерів можна побачити у таблиці 1.4 [21].

Таблиця 1.4 – Порівняння часу роботи алгоритму без маркерів та з маркерами

| № зображення | Розмір зображення | Час роботи оригінального алгоритму (годин:хвилин:секунд) | Час роботи модифікованого алгоритму (годин:хвилин:секунд) |
|---|---|---|---|
| 1 | 240 x 480 | 00:09:44 | 00:00:54 |
| 2 | 272 x 400 | 00:21:59 | 00:00:49 |
| 3 | 320 x 880 | 00:47:52 | 00:03:51 |
| 4 | 320 x 832 | 00:39:43 | 00:03:27 |
| 5 | 272 x 704 | 00:21:00 | 00:01:49 |
| 6 | 320 x 480 | 00:14:30 | 00:01:11 |
| 7 | 880 x 512 | 02:03:02 | 00:10:07 |
| 8 | 240 x 512 | 00:08:28 | 00:00:44 |
| 9 | 240 x 352 | 00:05:41 | 00:00:23 |
| 10 | 176 x 352 | 00:02:17 | 00:00:11 |

Таким чином удосконалення методу виявлення та локалізації областей фальсифікації цифрового зображення методом клонування-переміщення буде полягати у використанні більшого розміру блоку для обробки, через що зросте швидкість обробки зображення, проте зменшиться точність, використанні складних фігур замість квадратних блоків, завдяки чому вдасться нівелювати зменшення точності через збільшення розміру блоку, та використанні системи маркерів, які значно підвищують швидкість обробки зображення, через

відкидання блоків, у яких не потрібно обчислювати коефіцієнт кореляції Пірсона.

# 2 УДОСКОНАЛЕННЯ МЕТОДУ ВИЯВЛЕННЯ ТА ЛОКАЛІЗАЦІЇ ОБЛАСТЕЙ КЛОНУВАННЯ В ЦИФРОВИХ ЗОБРАЖЕННЯХ

## 2.1 Види цифрових зображень

Цифрове зображення – масив даних, отриманий шляхом дискретизації (аналого-цифрового перетворення) оригіналу. Будучи закодованим за допомогою особливого алгоритму і записаним на носій, цей масив даних стає файлом [22].

Цифрові зображення можна класифікувати за наступними властивостями:

1) Тип роздільної здатності

a) Статичний – растрова графіка. Отримується двома способами. Перший – сканування оригіналу – проводиться за допомогою особливого пристрою - сканера – в якому кожен оптичний елемент ПЗС-лінійки (або ПЗС-матриці) зчитує яскравості і колірні характеристики оригіналу. Ці характеристики перетворюються в двійковий код кольору і надсилаються в осередку двомірного масиву даних (матриці пікселів). Другий спосіб отримання растрового зображення - проектування оригіналу на ПЗС-матрицю через систему лінз (об'єктивів). Цей спосіб реєстрового аналого-цифрового перетворення характерний для цифрових фотоапаратів і відеокамер [23].

b) Динамічний – векторна графіка, яка складаються з від'їзд векторної графіки які залежать від деяких контрольних точок, завдяки своєму змісту математичних формул вони здатні формувати кривизни між однією точкою та іншою, а коли застосовується програма редагування, вона просто обчислює формулу та адаптує зображення до вимог користувача [24].

2) Формат збереження (тільки для растрової графіки)

a) Без втрат – зображення зберігається без змін матриці пікселів. Наприклад, .tiff, .png та .bmp.

b) З втратами – розмір графічних файлів зменшується за рахунок втрати частини даних і погіршення якості зображення. Стандартні розширення імен файлів — .jpg або .jpeg.

3) Кольоровий режим

a) Монохромний – палітра складається лише з чорного та білого кольору.

b) Градації сірого – палітра складається з чорного, білого та відтінків сірого.

c) RGB – матриця зображення складається з 3 матриць яскравості червоного, зеленого та синього кольорів.

d) YUV – матриця зображення складається з матриці градації сірого, та двох матриць кольороворізності (U – різність зеленого та червоного, Y – різність зеленого та синього).

Зображення, збережене у кольоровому режимі RGB можливо перевести до режиму YUV за формулою 2.1:

$$\begin{aligned} Y &= 0{,}299R + 0{,}587G + 0{,}114B \\ U &= -0{,}147R - 0{,}259G + 0{,}436B \\ V &= 0{,}615R - 0{,}515G - 0{,}100B \end{aligned} \tag{2.1}$$

Для зворотної операції необхідно застосувати формулу 2.2:

$$\begin{aligned} c &= Y - 16 \\ d &= U - 128 \\ e &= V - 128 \end{aligned} \tag{2.2}$$

$$R = [(298c + 409e + 128) \gg 8]_0^{255}$$
$$G = [(298c - 100d - 208e + 128) \gg 8]_0^{255} \quad (2.2)$$
$$B = [(298c + 516d + 128) \gg 8]_0^{255}$$

Для дослідження та удосконалення методу виявлення та локалізації областей клонування в цифрових зображеннях будуть використовуватись растрові зображення з кольоровим режимом RGB, збережені у форматі без втрат.

## 2.2 Оцінка подібності блоків зображення

У багатьох завданнях пов'язаних з обробкою та аналізом зображень виникає проблема оцінки якості зображення. На кінцеве якість зображення впливають як умови реєстрації зображення (переважно зумовлено освітленням), і параметри реєструючої системи (сюди належить і тракт передачі). Незважаючи на велику кількість розроблених методів оцінки якості зображення, завдання є досить складним і не має універсального рішення, і багато в чому залежить від завдань і методів обробки зображень.

У рамках вирішення завдання оцінки якості зображення розглядається два підходи – суб'єктивний та об'єктивний. Суб'єктивні оцінки ґрунтуються на тому, як люди сприймають якість зображення (експертна оцінка). Об'єктивна (кількісна) оцінка якості зображення реалізується за допомогою різних математичних методів та алгоритмів, розроблених у тому числі з урахуванням особливостей сприйняття зображень людиною. Як суб'єктивні, і об'єктивні оцінки може бути абсолютними чи порівняльними. Існуючі та розроблювані заходи оцінки якості можуть бути отримані шляхом порівняння з еталоном, так і у вигляді безеталонної міри. Методи, розроблені для оцінки відмінності одновимірних сигналів, знайшли своє застосування і при обчисленні міри якості зображень. У такому випадку, одне з зображень приймається як зразок, а всі наступні зображення порівнюються з ним. Як міра близькості, отже, мірою

якості, може виступати, наприклад, середньоквадратичне відхилення або коефіцієнт кореляції. В даний час, при оцінці якості цифрових зображень, більшість метрик базується на обчисленні індексу структурної подібності SSIM (від англ. Structural SIMilarity index), який є розвитком методів обчислення пікового відношення сигналу до шуму (PSNR) та середньоквадратичної помилки (MSE) [25].

Індекс структурної схожості (SSIM) – це перцепційна метрика, яка кількісно визначає погіршення якості зображення, спричинене такою обробкою, як стиснення даних, або втратами під час передачі даних. Це повна еталонна метрика, для якої потрібні два зображення з одного знімка — еталонне зображення та оброблене зображення. Оброблене зображення зазвичай стискається. Його можна, наприклад, отримати, зберігши еталонне зображення у форматі JPEG (будь-якого рівня якості), а потім знову зчитуючи його. SSIM є найвідомішим у відеоіндустрії, але має потужне застосування для фотозйомки [26]. SSIM розраховується за формулою 2.3:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2.3}$$

Де μ(x) – середнє значення для першої картинки, μ (y) – для другої, σ(x) – середньоквадратичне відхилення для першої картинки, і відповідно σ(y) – для другої, σ(x,y) це вже коваріація.

Середня квадратична помилка (MSE) вимірює кількість помилок у статистичних моделях. Він оцінює середню квадратичну різницю між спостережуваними та прогнозованими значеннями. Якщо модель не має помилок, MSE дорівнює нулю. Зі збільшенням похибки моделі її значення зростає. Середня квадратична помилка також відома як середньоквадратичне відхилення (MSD) [27]. MSE розраховується за формулою 2.4:

$$MSE = \frac{\sum(y_i - \hat{y}_i)^2}{n} \tag{2.4}$$

де $y_i$ – і-те спостережене значення, $\hat{y}_i$ – відповідне прогнозоване значення, n – кількість спостережень.

Загальноприйнятою величиною для оцінки втрат при відновленні зображень є метрика, звана пікове відношення сигнал/шум або PSNR. При цьому чим більше значення PSNR, тим менше втрат при відновленні і навпаки. Цей критерій визначається формулою 2.5:

$$PSNR = 20\log_{10} \frac{\max_{ij} |x_{ij}|}{\sigma_\varepsilon} \tag{2.5}$$

де $x_{ij}$ – значення яскравості точки з координатами (i; j); $\sigma_\varepsilon$ - середньоквадратичне відхилення між вихідним та відновленим сигналами зображень.

Коефіцієнт кореляції належить до SSIM та є статистичним показником сили взаємозв'язку між відносними рухами двох змінних. Значення коливаються від -1,0 до 1,0. Розраховане число більше 1,0 або менше -1,0 означає, що в кореляційному вимірі була помилка. Кореляція -1,0 показує ідеальну негативну кореляцію, тоді як кореляція 1,0 – ідеальну позитивну кореляцію. Кореляція 0,0 показує відсутність лінійної залежності між рухом двох змінних [28].

Коефіцієнт кореляції Пірсона (позначають «r») – в статистиці, показник кореляції (лінійної залежності) між двома змінними X та Y, який набуває значень від −1 до +1 включно. Він широко використовується в науці для вимірювання ступеня лінійної залежності між двома змінними.

Коефіцієнт кореляції Пірсона між двома змінними дорівнює сумі добутків відхилень, поділеній на добуток їх стандартних відхилень. Нехай, є дві вибірки $x^m = (x_1, \ldots, x_m)$, $y^m = (y_1, \ldots, y_m)$; Коефіцієнт кореляції Пірсона розраховують за формулою 2.6:

$$r_{xy} = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{m}(x_i - \bar{x})^2 \sum_{i=1}^{m}(y_i - \bar{y})^2} \tag{2.6}$$

де $\bar{x}$, $\bar{y}$ – вибіркові середні $x^m$ і $y^m$ [29].

2.3 Базовий метод виявлення та локалізації областей клонування в цифрових зображеннях

Перш за все розглянемо базовий метод, який потім буде модифікуватися. За основу візьмемо квадратні блоки 8x8.

Метод матиме наступні кроки:

1. Перевести зображення з режиму RGB до YUV.

2. Ініціалізувати область, що треба виявити та розбити матрицю яскравості цифрового зображення на множину квадратних блоків, що перетинаються (формула 2.7), розміром пікселей таких, що:

$$\bigcup_{i=1}^{s} c_i = Y, \tag{2.7}$$

(тут кожний наступний блок відрізняється від попереднього зсувом на 1 піксель вправо, вліво, вниз та угору).

3. Кожний блок $c_i$, $i = 1, \ldots, s$ розглянути в парі з усіма $c_j$, $j = i+1, \ldots, s$, відповідно. Для кожної пари розраховується коефіцієнт кореляції за формулою 2.6.

3.1. Якщо кореляція дорівнює 1, то блоки $c_i$ та $c_j$ – це оригінальний та клонований, після чого до результату додаються обидва блоки, за формулою 2.8:

$$Res = res \vee ci \vee cj. \tag{2.8}$$

3.2. Якщо кореляція не дорівнює одиниці, то перейти до наступної пари блоків

4. Вивести знайдену область res.

На вхід буде подаватись цифрове зображення (приклад на рисунку 2.1).

5. Рисунок 2.1 – Вхідне зображення

На виході ми отримаємо зображення, на якому будуть усі подібні області (приклад на рисунку 2.2).

Рисунок 2.2 – Вихідне зображення

В подальшому цей метод буде модифікований додаванням маркерів та використанням інших видів та розмірів блоків.

## 2.4 Удосконалення методу виявлення та локалізації областей клонування в цифрових зображеннях

Перш за все необхідно визначити, які блоки будуть використані. Оскільки було виявлено, що складні блоки розміром 16x16 мають приблизно ту ж саму точність, що і квадратні розміром 8x8, то розмір блоку буде 16x16. Далі необхідно визначити форму блоку. Спочатку було вирішено використовувати складні блоки, що показані на рисунку 2.3.

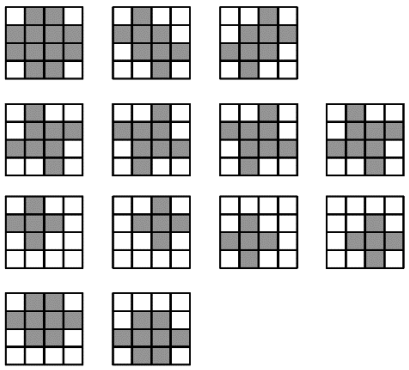


Рисунок 2.3 – Види складних блоків

Але, після експериментів, було виявлено, що усі блоки дають майже однакові результати, оскільки усі вони входять до самого першого блоку. Після цього було вирішено залишити лише 3 блоки, але змінити їх розмір. Блоки можна побачити на рисунку 2.4.

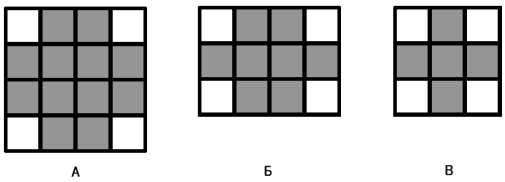



Рисунок 2.4 – Складні блоки, поділені на блоки 4х4

Блок А залишився 16х16. Розмір блоку Б змінився на 16х12, а блоку В – на 12х12. Пікселі на кутах блоків прибрані, щоб краще відстежувати нерівні контури області клонування. Оскільки ці блоки краще себе проявляють на периметрі області клонування, то, щоб уникнути пустих дир (рисунок 2.5), що будуть утворені через відсутність кутових пікселів блоку, до алгоритму буде додано додатковий пункт. Якщо коефіцієнт кореляції між двома складними блоками дорівнюватиме одиниці, то буде проведена додаткова перевірка на кореляцію, використовуючи вже квадратні блоки того ж розміру. Це підвищить час обробки зображення, проте дозволить підвищити точність.

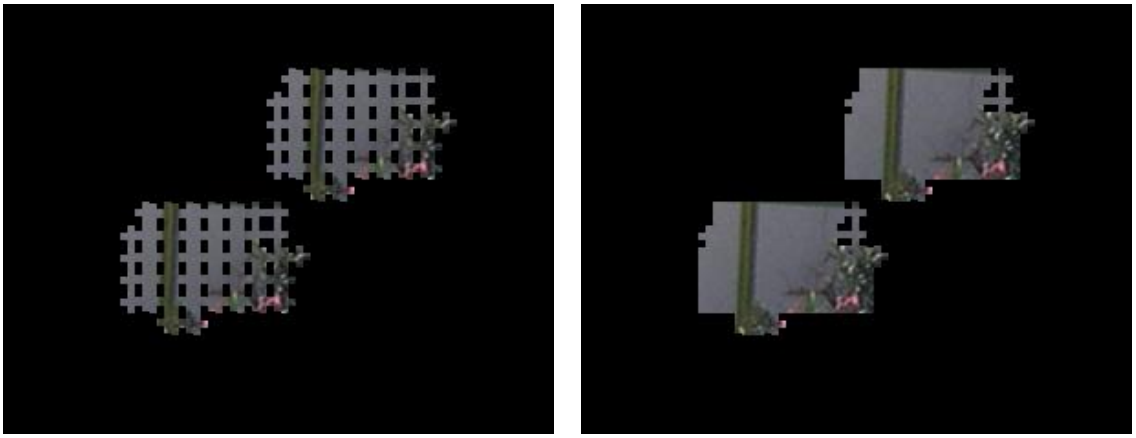


Рисунок 2.5 – Результат роботи алгоритму з використанням хрестового блоку 12x12 без додаткової перевірки (ліворуч) та з додатковою перевіркою (праворуч)

Таким чином удосконалений метод буде мати наступні кроки:

1. Перевести зображення з режиму RGB до YUV.

2. Ініціалізувати область, що треба виявити res = ∅ та розбити матрицю яскравості цифрового зображення на множину блоків, що перетинаються, за формулою 2.7

3. Кожний блок $c_i$, $i = 1, ..., s$ розглянути в парі з усіма $c_j$, $j = i+1, ..., s$, відповідно. Для кожної пари розраховується коефіцієнт кореляції за формулою 2.6.

3.1 Якщо коефіцієнт кореляції для квадратного блоку дорівнює одиниці, то додати до результату res квадратний блок.

3.2 Інакше получити складні блоки, обчислити коефіцієнт кореляції для них. Якщо кореляція дорівнює одиниці, то додати до результату res блок складної форми, інакше перейти до наступної пари блоків.

4. Вивести знайдену область res.

Після модифікації алгоритму складними блоками, необхідно додати до нього систему маркерів.

Маркери – це заздалегідь визначені пікселі обох блоків, які будуть зрівнюватись. Оскільки при порівнянні блоків ми вважаємо що один з них

клонований, коли коефіцієнт кореляції дорівнює одиниці, то можна зробити висновок: якщо хоча б один піксель блоків буде відрізнятись, то коефіцієнт кореляції не буде одиничним, тож немає сенсу їх порівнювати. Таким чином, порівняння блоків буде проходити лише якщо результат формули 2.9 буде 1:

$$comp = \left(a^i_{x1y1} == a^j_{x1y1}\right) \cap \left(b^i_{x2y2} == b^j_{x2y2}\right) \cap \left(c^i_{x3y3} == c^j_{x3y3}\right) \cap \left(d^i_{x4y4} == d^j_{x4y4}\right) \quad (2.9)$$

де:

$a^i_{x1y1}, b^i_{x2y2}, c^i_{x3y3}, d^i_{x4y4} \in c_i$ – пікселі-маркери блоку з яким порівнюють;

$a^j_{x1y1}, b^j_{x2y2}, c^j_{x3y3}, d^j_{x4y4} \in c_j$ – пікселі-маркери блоку, який порівнюють;

$x1y1, x2y2, x3y3, x4y4$ – координати маркерів.

Для квадратних блоків різних розмірів було запропоновано використовувати кутові пікселі у якості маркерів(рис. 2.6) [2121].

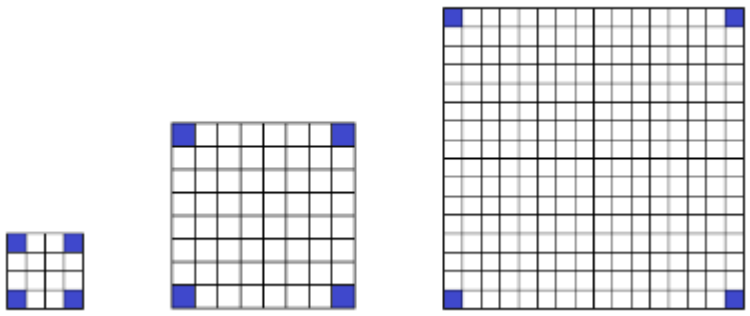


Рисунок 2.6 – Маркери блоку квадратної форми розміру 4x4, 8x8, 16x16

Проте для складних блоків кутові пікселі не підходять через те, що знаходяться поза області порівняння. Через це необхідно обрати нові позиції для маркерів. Для цього було вирішено використовувати кутові пікселі внутрішніх квадратів блоків(рис. 2.7).

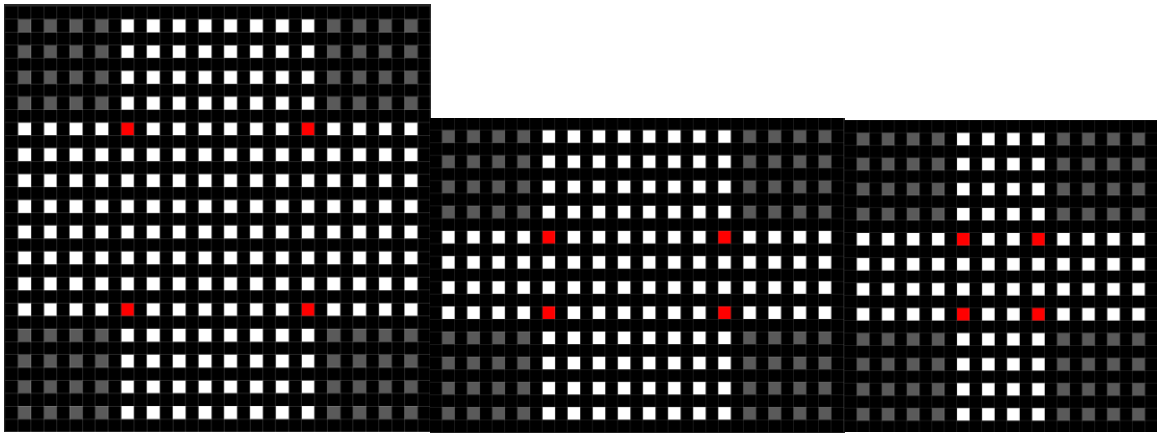


Рисунок 2.7 – Маркери складних блоків – великого хреста(ліворуч), овалу(по центру) та малого хреста(праворуч)

В ході експериментів були випробувані позиції маркерів ближче до периметру блоків, проте різниця часу була незначною та вимірялася у сотих секунди. Таким чином, остаточний модифікований алгоритм буде мати наступну структуру:

1. Перевести зображення з режиму RGB до YUV.

2. Ініціалізувати область, що треба виявити res = ∅ та розбити матрицю яскравості цифрового зображення на множину блоків, що перетинаються, за формулою 2.7

3. Кожний блок $c_i$, $i = 1, \ldots, s$ розглянути в парі з усіма $c_j$, $j = i+1, \ldots, s$, відповідно. Для кожної пари провести перевірку 2.9.

3.1 Якщо маркери для квадратного блоку дорівнюють одиниці, то обчислюємо коефіцієнт кореляції для квадратного блоку. Якщо коефіцієнт корреляції дорівнює одиниці, то додати до результату res квадратний блок.

3.2 Інакше перевірити маркери для складних блоків. Якщо comp == 1, обчислити коефіцієнт кореляції для них. Якщо кореляція дорівнює одиниці, то додати до результату res блок складної форми, інакше перейти до наступної пари блоків.

3.3 Якщо comp не дорівнює одиниці, то перейти до наступної пари блоків.

4. Вивести знайдену область res.

В даному розділі розглянуто метод виявлення та локалізації областей клонування в цифрових зображеннях, та запропоновано вирішення таких недоліків, час роботи та точність. Для цього були змінені форми та розміри блоків, додана система маркерів та розроблено удосконалення методу виявлення та локалізації областей клонування з використанням складних блоків та маркерів, який в подальшому можна використовувати для виявлення областей фальсифікації цифрових зображень.

# 3 ПРОГРАМНА РЕАЛІЗАЦІЯ УДОСКОНАЛЕНОГО МЕТОДУ ВИЯВЛЕННЯ ТА ЛОКАЛІЗАЦІЇ ОБЛАСТЕЙ КЛОНУВАННЯ В ЦИФРОВИХ ЗОБРАЖЕННЯХ

## 3.1 Програмне середовище

Visual Studio Community 2019 – комплексне середовище IDE для розробників .NET і C++ у Windows, що має повноцінний набір інструментів та функцій для покращення та вдосконалення кожного етапу розробки програмного забезпечення. Надається безкоштовно для студентів, учасників створення відкритого вихідного коду та окремих користувачів [30].

Має наступні засоби підвищення продуктивності [31]:

– Хвилясті лінії позначають помилки або потенційні проблеми коду під час введення. Ці візуальні підказки допомагають негайно усунути проблеми, не чекаючи появи помилок під час складання чи виконання. Якщо навести вказівник на хвилясту лінію, на екран буде виведено додаткові відомості про помилку. Також у полі зліва може відображатися лампочка, що вказує на наявність відомостей про швидкі дії для усунення помилки.

– Одним натисканням кнопки можна відформатовати код та застосувати до нього виправлення, запропоновані параметрами стилю коду, угодами у файлі .editorconfig та (або) аналізаторами Roslyn. Очищення коду, яке зараз доступне лише для коду C#, допомагає усувати проблеми в коді перед переходом до його перевірки.

– Рефакторинг включає такі операції, як інтелектуальне перейменування змінних, вилучення однієї або декількох рядків коду в новий метод і зміна порядку параметрів методів.

– IntelliSense – це набір можливостей, що відображають відомості про код безпосередньо в редакторі та в деяких випадках автоматично створюють

невеликі уривки коду. По суті, це вбудована редактор базова документація, яка позбавляє необхідності шукати інформацію в інших джерелах.

– Функції IntelliSense залежить від мови. Для отримання додаткових відомостей див. посібники з IntelliSense для C# , IntelliSense для Visual C++, IntelliSense для JavaScript та IntelliSense для Visual Basic.

– Щоб швидко знаходити функції інтегрованого середовища розробки або елементи коду, Visual Studio представлений єдиний компонент пошуку (CTRL + Q).

– Додаткові відомості та поради щодо підвищення продуктивності див. у розділі Практичний посібник.

– Є можливості спільного редагування та налагодження в реальному часі незалежно від типу програми або мови. Можливо миттєво надавати спільний доступ до свого проекту за допомогою високого рівня безпеки. Крім того, можливо надавати спільний доступ до сеансів, екземплярів терміналу, веб-додатків на локальному комп'ютері, голосових дзвінків тощо.

– У вікні Ієрархія дзвінків показано методи, що викликають вибраний метод, що корисно, при зміні або видаленні методу або відстежуванні помилки.

– CodeLens допомагає знаходити посилання на код, зміни коду, пов'язані з кодом помилки, робочі елементи, перевірки коду та модульні тести – не виходячи з редактора.

– Функція Перейти до визначення дозволяє перейти до розташування, де визначено вибрану функцію або тип.

– У вікні Показати визначення можна відобразити метод або визначення типу, не відкриваючи окремий файл.

C# – сучасна об'єктно-орієнтована мова програмування. C# дозволяє розробникам створювати різні типи безпечних та надійних програм, що виконуються в .NET. C# відноситься до широко відомого сімейства мов C, і здається добре знайомим будь-кому, хто працював з C, C++, Java або JavaScript.

Функції мови C#, які дозволяють створювати надійні та стійкі програми [32]:

– Складання сміття автоматично звільняє пам'ять, зайняту недосяжними об'єктами, що не використовуються.

– Типи, що допускають значення null, забезпечують захист від змінних, які посилаються виділені об'єкти.

– Обробка винятків надає структурований та розширюваний підхід до виявлення помилок та відновлення після них.

– Лямбда-вираження підтримують прийоми функціонального програмування.

– Синтаксис LINQ створює загальний шаблон для роботи з даними будь-якого джерела.

– Підтримка мов для асинхронних операцій забезпечує синтаксис для створення розподілених систем.

– C# є Єдина система типів. Всі типи C#, включаючи типи-примітиви, такі як int та double, успадковують від одного кореневого типу об'єкта. Всі типи використовують загальний набір операцій, а значення будь-якого типу можна зберігати, передавати та обробляти таким чином.

– C# підтримує як визначені користувачами типи посилань, так і типи значень.

– C# дозволяє динамічно виділяти об'єкти та зберігати спрощені структури у стеку.

– C# підтримує універсальні методи та типи, що забезпечують підвищену безпеку типів та продуктивність.

– C# надає ітератори, які дозволяють розробникам класів колекцій визначати варіанти поведінки для клієнтського коду.

Windows Forms — це платформа інтерфейсу користувача для створення класичних програм Windows. Вона забезпечує один з найефективніших способів створення класичних програм за допомогою візуального конструктора в Visual Studio. Такі функції, як розміщення візуальних елементів керування шляхом перетягування, полегшують створення класичних додатків.

У Windows Forms можна розробляти графічно складні програми, які просто розгортати, оновлювати і з якими зручно працювати як в автономному режимі, так і в мережі. Програми Windows Forms можуть отримувати доступ до локального обладнання та файлової системи комп'ютера, на якому працює програма [32].

## 3.2 Інтерфейс програми

Для зручного використання модифікованого методу виявлення та локалізації областей клонування в цифрових зображеннях була розроблена програма з користувацьким інтерфейсом на базі мови програмування C# та платформи Windows Forms. Програма розроблена для використання на машинах з операційною системою Windows 10.

При запуску програми викликається основне вікно програми, що можна побачити на рисунку 3.1.

В основному вікні знаходяться декілька меню з налаштуваннями. Меню File складається з кнопок Open image, завдяки якому можна обрати цифрове зображення, яке буде оброблятись, та Exit, яка закриває програму.

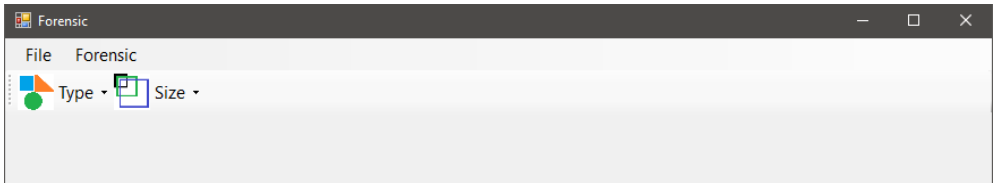



Рисунок 3.1 – основне вікно

Вид меню можна побачити на рисунку 3.2. Кнопка Open image викликає діалогове вікно, в якому можна обрати файл цифрового зображення.

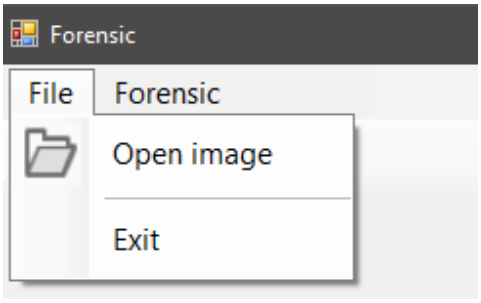



Рисунок 3.2 – Меню File

Діалогове вікно можна побачити на рисунку 3.3. Оскільки алгоритм працює лише з зображеннями, що збережені у форматі без втрат, можна обрати лише файли з розширенням .bmp або .tiff.

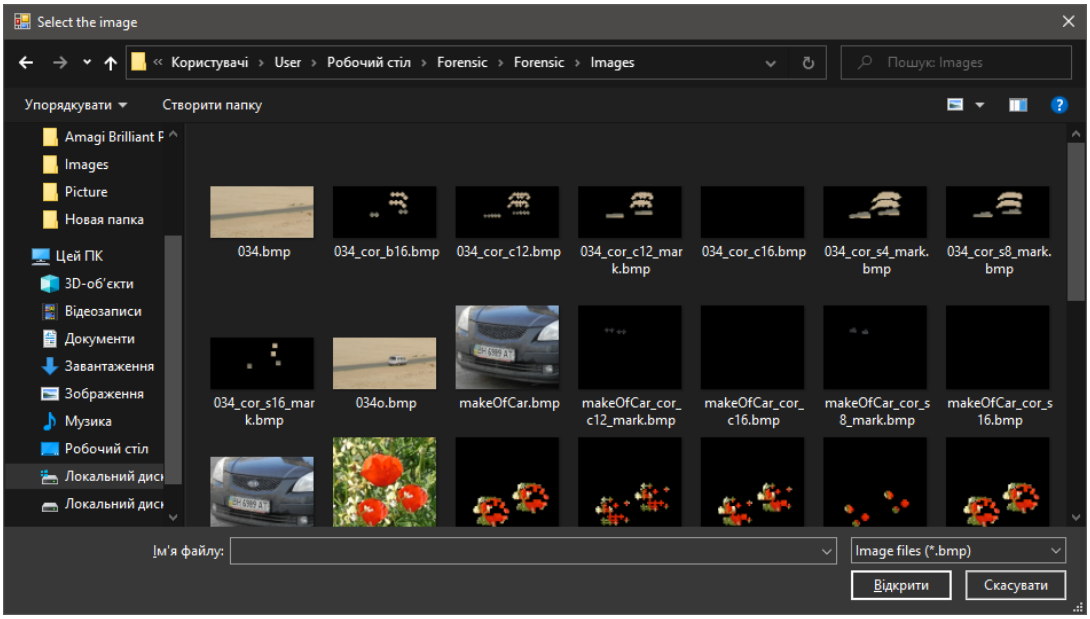



Рисунок 3.3 – Діалогове вікно вибору файлу

Обравши необхідне зображення, далі необхідно визначитись із формою блоку, яким буде оброблятись зображення. Для цього треба натиснути на кнопку Type. Після цього відкриється меню, де буде представлен вибір із доступних видів блоків(рисунок 3.4).

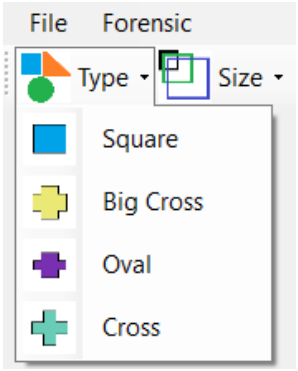



Рисунок 3.4 – Меню вибору форми блоку

Далі, якщо був обраний квадратний блок, необхідно обрати його розмір у меню Size, де, у випадаючому меню, на вибір даються наступні розміри: 4x4, 8x8 та 16x16(рисунок 3.5). Для блоків форми великий хрест, овал та хрест розмір блоків фіксований – 16x16, 16x12 та 12x12 відповідно, тож обирати цей параметр не потрібно.

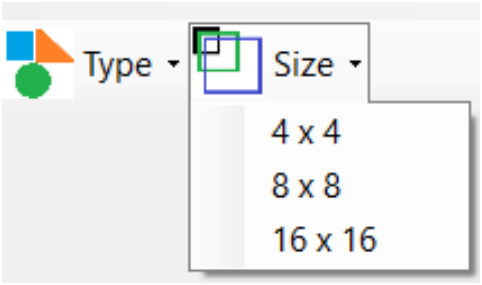



Рисунок 3.5 – Меню вибору розміру блоку

Закінчивши з налаштуванням, необхідно обрати, чи буде алгоритм обробляти зображення, використовуючи маркери, чи ні. Для цього потрібно відкрити меню Forensic, де обирається метод обробки(рисунок 3.6).

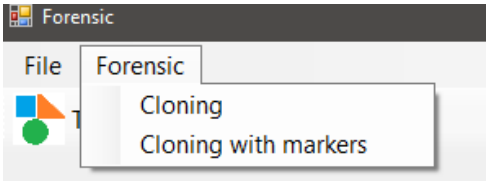



Рисунок 3.6 – Меню вибору варіанту алгоритму

Після натискання на одну з двох кнопок в меню відкриється вікно обробки. У верхній частині вікна будуть відображені усі обрані параметри алгоритму та розмір зображення, а з лівої сторони буде показане обране

зображення. У правому верхньому куту буде кнопка Start, натискання на яку запустить обробку(Рисунок 3.7).

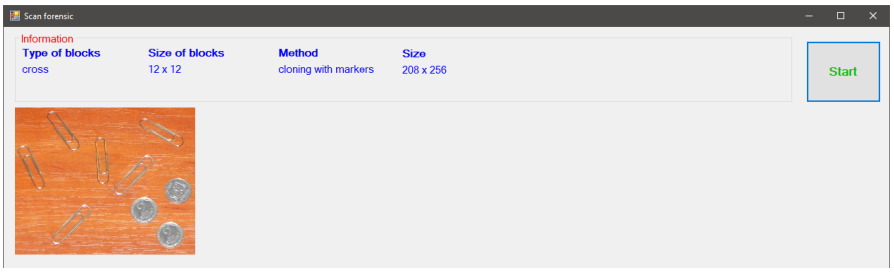



Рисунок 3.7 – Вікно обробки

Після початку обробки, прогрес бар у нижній частині вікна почне заповнюватись. Коли він заповниться до кінця, з’явиться повідомлення про завершення обробки та час, за котрий була проведена обробка(рисунок 3.8).

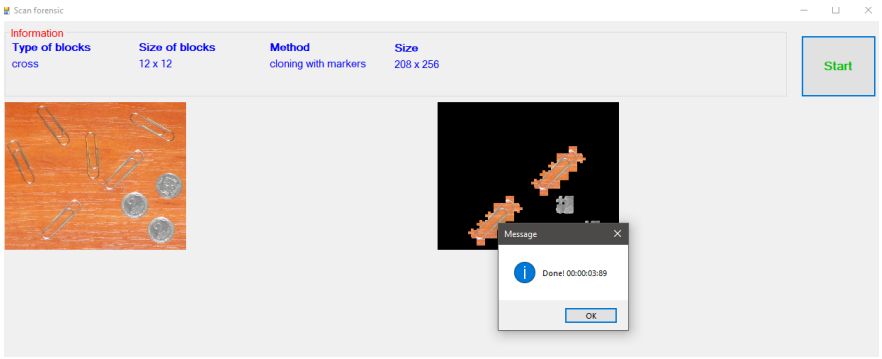



Рисунок 3.8 – Кінець обробки

Разом з цим, у правій частині вікна обробки з’явиться зображення з результатом обробки. Там будуть показані усі блоки, які дублюються у зображенні. Також, програма створює нове зображення у папці вхідного зображення з результатами обробки (рисунок 3.9).

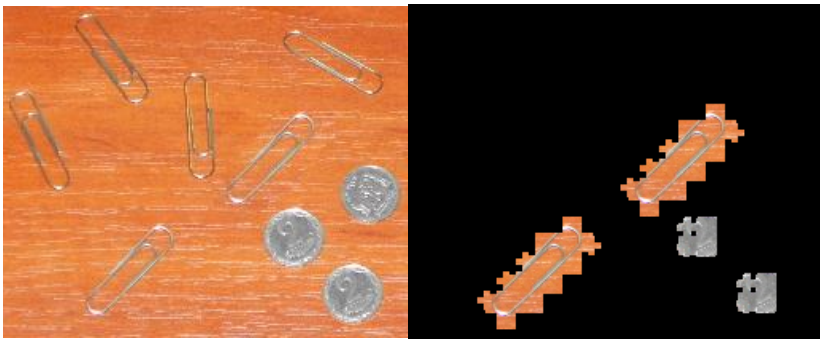


Рисунок 3.9 – Вхідне (ліворуч) та вихідне (праворуч) зображення.

Якщо у вихідному зображенні повністю чорний прямокутник, то алгоритм не зміг знайти сліди клонування. Це може бути у разі, якщо зображення не було модифіковане простим клонуванням без модифікації, або клонована область занадто мала, щоб алгоритм її розпізнав.

### 3.3 Результати експериментів

Для дослідження ефективності модифікованого методу виявлення та локалізації областей клонування в цифрових зображеннях проведено обчислювальний експеримент з використанням 17 зображень.

Експеримент проводився на обчислювальній машині з процесором Intel Core I5-11400 та оперативною пам’яттю обсягом 32548 Мб.

Оцінку ефективності виконано у вимірюванні часу та точності роботи методу виявлення та локалізації областей клонування.

Результати експериментів наведені нижче у таблиці 3.1, 3.2 та 3.3.

Таблиця 3.1 – Точність блоків

| Типи блоків та їх розмір | Процент знайденої площі клонованої області | | |
|---|---|---|---|
| | Максимальне | Мінімальне | Середнє |
| Квадратні 8x8 | 85,22% | 19,22% | 55,84% |
| Великий хрест 16x16 | 68,28% | 0% | 40,30% |
| Овал 16x12 | 80,07% | 13,12% | 47,63% |
| Хрест 12x12 | 80,46% | 19,56% | 52,48% |

Як можна побачити з результатів експерименту, задачу підвищення точності не вдалося досягти, проте точність залишилася на приблизно тому ж рівні при використанні хрестових блоків розміром 12x12. Використання блоків форми великий хрест та овал виявилося нерезультативним.

Таблиця 3.2 – Час роботи алгоритму без маркерів при використанні різних блоків у форматі (ГГ:ХХ:СС)

| Номер зображення | Розмір | Час роботи при квадратних блоках | Час роботи при великих хрестових блоках | Час роботи при овальних блоках | Час роботи при хрестових блоках |
|---|---|---|---|---|---|
| 1 | 301x235 | 0:03:08 | 0:01:44 | 0:01:49 | 0:02:26 |
| 2 | 352x176 | 0:02:27 | 0:01:13 | 0:01:29 | 0:01:56 |
| 3 | 256x208 | 0:01:56 | 0:01:07 | 0:01:07 | 0:01:27 |
| 4 | 216x216 | 0:01:30 | 0:00:47 | 0:00:47 | 0:01:09 |
| 5 | 704x272 | 0:25:06 | 0:17:40 | 0:17:10 | 0:18:05 |
| 6 | 880x320 | 1:02:02 | 0:44:54 | 0:37:54 | 0:43:31 |
| 7 | 832x320 | 0:49:50 | 0:28:38 | 0:28:03 | 0:41:06 |
| 8 | 512x240 | 0:11:34 | 0:06:39 | 0:06:01 | 0:09:58 |

Як можна побачити, навідмінно від квадратних блоків, при використанні складних блоків обробка проходить значно швидше. При тому чим більше розмір зображення, тим більша буде різниця часу обробки. Хрестові блоки 12x12, які мають приблизно таку ж точність, що і квадратні блоки 8x8, обробляються майже в 2 рази швидше.

Таблиця 3.3 – Час роботи алгоритму з маркерами при використанні різних блоків у форматі (ГГ:ХХ:СС)

| Номер зображення | Розмір | Час роботи при квадратних блоках | Час роботи при великих хрестових блоках | Час роботи при овальних блоках | Час роботи при хрестових блоках |
|---|---|---|---|---|---|
| 1 | 301x235 | 0:00:14 | 0:00:03 | 0:00:04 | 0:00:06 |
| 2 | 352x176 | 0:00:16 | 0:00:04 | 0:00:05 | 0:00:07 |
| 3 | 256x208 | 0:00:11 | 0:00:03 | 0:00:03 | 0:00:05 |

Продовження таблиці 3.3

| Номер зображення | Розмір | Час роботи при квадратних блоках | Час роботи при великих хрестових блоках | Час роботи при овальних блоках | Час роботи при хрестових блоках |
|---|---|---|---|---|---|
| 4 | 216x216 | 0:00:09 | 0:00:02 | 0:00:02 | 0:00:04 |
| 5 | 704x272 | 0:01:53 | 0:00:27 | 0:00:37 | 0:00:51 |
| 6 | 880x320 | 0:04:07 | 0:01:06 | 0:01:24 | 0:01:54 |
| 7 | 832x320 | 0:03:45 | 0:00:56 | 0:01:12 | 0:01:41 |
| 8 | 512x240 | 0:00:45 | 0:00:11 | 0:00:14 | 0:00:21 |

При використанні маркерів, швидкість обробки складних блоків перевищує швидкість обробки квадратних блоків від 2, при використанні малих хрестових блоків 12x12, до 4, при використанні великих хрестових боків 16x16, разів. Виходячи з результатів експериментів, хоч і не вдалося підвищити точність алгоритму, проте хрестовий блок 12x12 можна використовувати у якості заміни стандартних блоків 8x8.

# ВИСНОВКИ

У результаті виконання кваліфікаційної роботи, були проаналізовані вітчизняні та іноземні дослідження з виявлення та локалізації областей клонування в цифрових зображеннях.

Були запропоновані блоки складної форми для виявлення та локалізації областей клонування в цифрових зображеннях та перевірена доцільність використання запропонованих видів блоків складної форми. Були запропоновані маркери для визначених блоків складної форми.

Було удосконалено метод виявлення та локалізації областей клонування в цифрових зображеннях шляхом застосування складних блоків та маркерів, завдяки чому вдалося зменшити час обробки зображення більш ніж у 10 разів.

Було програмно реалізовано удосконалений метод виявлення та локалізації областей клонування в цифрових зображеннях та оцінити його ефективність.

# ПЕРЕЛІК ПОСИЛАНЬ


1. Цивільний процесуальний кодекс України: від 18.03.2004 № 1618-IV Редакція від 13.08.2020. URL: https://zakon.rada.gov.ua/laws/show/1618-15
2. Shearer E. More than eight-in-ten Americans get news from digital devices. *Pew Research Center*. 12.01.2021. URL: https://www.pewresearch.org/fact-tank/2021/01/12/more-than-eight-in-ten-americans-get-news-from-digital-devices/.
3. Consumption of online news rises in popularity. *Eurostat*. 24.08.2022. URL: https://ec.europa.eu/eurostat/en/web/products-eurostat-news/-/ddn-20220824-1.
4. Broz M., Number of Photos: Statistics, Facts, & Predictions. *Phototutorial*, 2022 URL: https://photutorial.com/photos-statistics/.
5. Smith K., 126 Amazing Social Media Statistics and Facts. *Brandwatch*.2019. URL: https://www.brandwatch.com/blog/amazing-social-media-statistics-and-facts/.
6. Gowri S.M., Ganesh B.C., Velusamy S., Vidyavathi K., Gandhi R. Digital Image Falsification Detection Based on Dichromatic Model. *Data Engineering and Communication Technology. Lecture Notes on Data Engineering and Communications Technologies*. 2021. V 63. Springer, Singapore. URL: https://doi.org/10.1007/978-981-16-0081-4_36
7. Sridevi M., Mala C., Sanyam S. Comparative Study of Image Forgery and Copy-Move Techniques. *Advances in Computer Science, Engineering & Applications. Advances in Intelligent and Soft Computing*. 2012. V 166. Springer, Berlin, Heidelberg.
8. Thapaliya A., Atonge D., Mazzara M., Chakraborty S., Afanasyev I., Ahmad M. Digital Image Forgery. *VI International Young Scientists Conference- Information technologies, telecommunications and control systems.* 2020. Innopolis. URL: https://www.researchgate.net/publication/337719445_Digital_Image_Forgery
9. Meena K.B., Tyagi V. Image Splicing Forgery Detection Techniques: A Review. *International Conference on Advances in Computing and Data Sciences*

*ICACDS*. 2021. V 1441. Springer, Cham. URL: https://doi.org/10.1007/978-3-030-88244-0_35

10. Rani P., Rani J. Copy-move morgery attack detection in digital images. *International Journal of Engineering Research and Technology IJERT*. 2015. V.4. Is.6. P. 118-132. URL: http://dx.doi.org/10.17577/IJERTV4IS061110

11. Зоріло В.В., Лебедєва О.Ю. Комплексний метод виявлення і локалізації областей клонування у цифрових зображеннях. *Праці Одеського політехнічного університету.* 2015. Вип. 1(45). С. 101-106

12. Лебедева, Е. Ю. Обнаружение зеркально отраженных клонированных участков изображения. *Вісник Східноукр. нац. ун-ту ім. В. Даля.* 2012. № 8 (179). С. 309-314.

13. Meena K. B., Tyagi V. A copy-move image forgery detection technique based on Gaussian-Hermite moments. *Multimedia Tools and Applications*. 2019. V 78.

14. Wang C., Zhang Zh., Li Q., Zhou X. An Image Copy-Move Forgery Detection Method Based on SURF and PCET. *IEEE Access*. 2019. V.7. P. 170032-170047.

15. Pan X., Lyu S. Region Duplication Detection Using Image Feature Matching. *IEEE Transactions* on Information Forensics and Security. 2011 V.5(4). P. 857 - 867.

16. Amerini I., Ballan L., Caldelli R., Del Bimbo A., Serra G. A SIFT-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery. *IEEE Transactions on Information Forensics and Security*. 2011. V.6. P. 1099 - 1110. URL: https://ieeexplore.ieee.org/document/5734842

17. Dixit A., Bag S. Composite attacks-based copy-move image forgery detection using AKAZE and FAST with automatic contrast thresholding. *IET Image Processing*. 2020. No.14. URL: https://www.researchgate.net/publication/350036897_Composite_attacks-based_copy-move_image_forgery_detection_using_AKAZE_and_FAST_with_automatic_contrast_thresholding

18. Лебедєва О.Ю. Використання круглих блоків для виявлення області фальсифікації в цифрових зображеннях. *Інформатика та математичні методи в моделюванні*. 2015. Том 5, №1. С. 71-76

19. Lebedeva Ye. Yu., Kobozeva A. A., Zorilo V. V. Accuracy improvement of cloning area detection. *Odes'kyi Politechnichnyi Universytet, Pratsi*. 2016. Is.3 (50). P. 47-53.

20. Баранов П. Ю., Лебедєва О.Ю. Виявлення області фальсифікації цифрового зображення блоками трикутної форми. *Інформатика та математичні методи в моделюванні*. 2011. Т. 1, № 3. С. 274-281.

21. Васалатій Р.І. Модифікація алгоритму виявлення та локалізації областей клонування в цифрових зображеннях: кваліфікаційна робота бакалавра. Одеса : НУ «ОП», 2022. 54 с.

22. Цифрові зображення і їх види. ni.biz.ua - Навчальна Інформація для українських студентів : веб-сайт. URL: http://www.ni.biz.ua/3/3_2/3_2549 _tsifrovie-izobrazheniya-i-ih-vidi.html

23. Jorge N. Цифрові зображення, типи та характеристики. *ТворчийOnline*. URL: https://www.creativosonline.org/uk/характеристики-типів-цифрових-зображень.html

24. Коефіцієнт кореляції. *Фінансова енциклопедія*. URL: https://ua.nesrakonk.ru/correlationcoefficient/ (дата звертання: 17.12.2022)

25. Галилейский В.П., Елизаров А.И., Кокарев Д.В., Морозов А.М. Меры оценки качества изображения. *25 Международный симпозиум «Оптика атмосферы и океана. Физика атмосферы»*. 2019. С 163-168

26. SSIM: Structural Similarity Index. *Imatest Documentation – Current v22.2*. URL: https://www.imatest.com/docs/ssim/ (Дата звертання: 19.12.2022)

27. Mean Squared Error : Overview, Examples, Concepts and More. Data Science & Business Analytics. *Simplilearn.* URL: https://www.simplilearn.com/tutorials/statistics-tutorial/mean-squared-error

28. Коефіцієнт_кореляції_Пірсона. *Матеріал з вікіпедії – вільної енциклопедії*. URL: https://uk.wikipedia.org/wiki/Коефіцієнт_кореляції_Пірсона

29. Visual Studio 2022. *Microsoft*. URL: https://visualstudio.microsoft.com/ru/downloads/

30. Добро пожаловать в интегрированную среду разработки Visual Studio. *Microsoft Learn*. URL: https://learn.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2022

31. Краткий обзор языка C#. *Microsoft Learn*. URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/

32. Руководство по классическим приложениям (Windows Forms .NET). *Microsoft Learn*. URL: https://learn.microsoft.com/ru-ru/dotnet/desktop/winforms/overview/?view=netdesktop-6.0