УДК 004.8:004.056

Д. Ю. Хома

МОДИФІКАЦІЯ СТЕГАНОГРАФІЧНОЇ МОДЕЛІ STEGANOGAN НА ОСНОВІ НОВІТНІХ ГЕНЕРАТИВНИХ АРХІТЕКТУР

У статті розглянуто створення стегозображень за допомогою методів на основі штучного інтелекту. Проведено огляд предметної області стеганографії з акцентом на нейромережеві підходи, визначено ключові метрики для оцінки стеганографічних моделей. Представлено три варіанти архітектури SteganoGAN на базі різних генеративних нейромереж, які навчено на даних із різними рівнями глибини корисного навантаження. Дослідження зосереджено на модифікації та вивченні архітектур DCGAN, WGAN і WGAN-GP як основи для створення SteganoGAN-моделей, здатних вбудовувати приховану інформацію в зображення із заданим корисним навантаженням . Усі моделі адаптовано до специфіки стеганографічного завдання шляхом інтеграції спеціалізованого декодувального модуля, застосування функцій втрат, що враховують якість відновлення повідомлення та візуальну схожість між контейнером і прикриттям, а також варіативності рівня шуму, що подається на вхід генератору залежно від заданого обсягу навантаження. Важливу роль у навчанні відіграє підбір співвідношення між втратами генератора та декодера, що забезпечує баланс між непомітністю та точністю відновлення. В експериментальній частині моделі перевірялись у контрольованих умовах із фіксованими параметрами тренування, а також із різним корисним навантаженням, що дозволило простежити реакцію моделей на зміну обсягу прихованої інформації. Окрему увагу приділено узагальненню методики оцінювання моделей за допомогою кількісних метрик, серед яких використовувалися показники точності декодування, RS-BPP (відношення кількості переданої інформації до розміру зображення), з урахуванням метрик структурної подібності (SSIM) та пікового відношення сигналу до шуму (PSNR). Виконано порівняльний аналіз отриманих стегозображень з точки зору різного корисного навантаження. На основі отриманих даних визначено сильні та слабкі сторони підходів, а також окреслено основні завдання для подальших досліджень у галузі стеганографії.

Ключові слова: *стеганографія, генеративна змагальна мережа, нейронна мережа, відстань Вассерштейна, штраф градієнту, корисне навантаження, схожість, кодування Ріда-Соломона, критик, обкладинка, стегозображення.*

Вступ

В умовах стрімкого розвитку цифрових технологій і зростання потреби у прихованому обміні інформацією стеганографія набуває нового значення як інструмент забезпечення конфіденційності. Одночасно генеративно-змагальні мережі (GAN) відкривають революційні можливості в галузі штучного інтелекту, дозволяючи створювати високоякісні синтетичні дані. Різноманітність архітектур GAN, таких як DCGAN із глибокими згортковими шарами, WGAN із покращеною стабільністю навчання через функцію втрат Вассерштейна, WGAN-GP із градієнтним штрафом для ще більшої якості генерації, а також інші моделі, наприклад CycleGAN чи StyleGAN, демонструють широкий спектр можливостей для синтезу даних. Поєднання стеганографії та GAN дає змогу розробляти інноваційні методи приховування інформації в зображеннях, відомих як стегозображення, що поєднують реалістичність і стійкість до виявлення.

Стеганографія зображень із використанням GAN дозволяє вбудовувати конфіденційні дані, такі як текст чи графічна інформація, у синтетичні зображення, приховуючи сам факт наявності повідомлення. Ефективність таких методів залежить від обраної архітектури нейромережі, здатності генератора створювати непомітні стегозображення та стійкості до аналізу дискримінаторами чи спеціалізованими стегоаналізаторами. Оцінка якості стегозображень вимагає комплексного підходу з використанням метрик, що враховують як візуальну достовірність, так і безпеку прихованої інформації.

Особливу увагу в сучасних дослідженнях приділяють адаптації GAN до стеганографічних задач, де архітектура мережі впливає на баланс між обсягом прихованої інформації та

непомітністю змін у зображенні. Наприклад, DCGAN забезпечує високу якість зображень завдяки згортковим шарам, тоді як WGAN і WGAN-GP вирішують проблеми стабільності навчання, що критично для створення стегозображень із високим корисним навантаженням. Такі відмінності між архітектурами відкривають простір для експериментів, спрямованих на оптимізацію стеганографічних систем у реальних сценаріях.

Метою роботи є дослід трьох різних підходів до навчання генеративно-змагальних мереж шляхомрозробки, модифікації, навчання та порівняльного аналізу архітектур моделі SteganoGAN на основі DCGAN, WGAN і WGAN-GP із різними рівнями глибини корисного навантаження, оцінки якості й стійкості стегозображень за допомогою метрик, а також визначення ключових викликів і напрямків розвитку нейромережевої стеганографії.

Стеганографія та роль генеративно-змагальних мереж

Стеганографія є методом приховування даних у носіях, таких як цифрові зображення, із метою маскування факту існування інформації, на відміну від криптографії, яка забезпечує її конфіденційність через шифрування. У контексті цифрових зображень стеганографія передбачає вбудовування даних у піксельну структуру носія з мінімальним впливом на його візуальні характеристики. Сучасні вимоги до конфіденційності даних у цифровому середовищі зумовлюють потребу в методах, що поєднують високу ємність корисного навантаження зі стійкістю до стегоаналізу. Традиційні підходи, такі як WOW [1], HUGO [2], S-UNIWARD [3] до стеганографії демонструють обмеження в обсязі прихованої інформації та вразливість до виявлення, що вимагає розробки більш ефективних технологій. У цьому контексті інтеграція штучного інтелекту, зокрема генеративно-змагальних мереж, розглядається як перспективний напрям для вдосконалення стеганографічних систем.

Генеративно-змагальні мережі (GAN) являють собою архітектуру штучного інтелекту, що складається з генератора, який синтезує дані, та дискримінатора, який оцінює їхню автентичність, забезпечуючи високу якість згенерованого контенту [4]. Приклад архітектури GAN наведено на рис. 1.

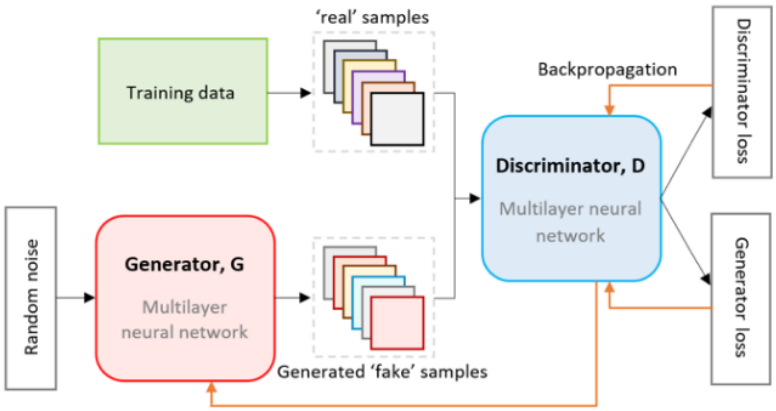

Рис. 1. Приклад класичної архітектури GAN [5]

У стеганографії GAN застосовуються для створення стегозображень – зображень із прихованою інформацією, що характеризуються реалістичністю та стійкістю до аналізу. Генератор вбудовує дані в синтетичне зображення, оптимізуючи його структуру для мінімізації помітних змін, тоді як дискримінатор сприяє підвищенню правдоподібності результату. Такий підхід дозволяє досягти кращого балансу між обсягом прихованої інформації та її непомітністю порівняно з традиційними методами. Застосування GAN у стеганографії відкриває можливості для створення зображень, адаптованих до специфічних вимог безпеки та якості.

Нейромережеві підходи, зокрема GAN, забезпечують значні переваги в стеганографії завдяки здатності генерувати нові носії з вбудованою інформацією, а не модифікувати наявні. Це дозволяє підвищити ємність корисного навантаження та ускладнити виявлення прихованої інформації сучасними стегоаналізаторами. Гнучкість архітектур GAN дає змогу

адаптувати моделі до різних сценаріїв використання, оптимізуючи параметри генерації для конкретних задач. Водночас використання GAN у стеганографії пов'язане з викликами, такими як вибір метрик оцінки якості стегозображень, обчислювальна складність і потреба в протидії вдосконаленим методам аналізу.

Архітектури GAN

Генеративно-змагальна мережа (GAN) була вперше представлена як архітектура з двох нейронних мереж: генератора, який створює синтетичні дані, та дискримінатора, який розрізняє справжні дані від згенерованих[4]. Навчання базується на функції втрат дискримінатора D:

$$L_D = -E_{x\sim p_{data}(x)}[log\ D\ (x)] - E_{z\sim p_z(z)}\left[log\left(1 - D\big(G(z)\big)\right)\right],$$

де$E_{x\sim p_{data}(x)}$ – математичне сподівання для реальних даних x, взятих із розподілу $p_{data}(x)$,$E_{z\sim p_z(z)}$ – сподівання для вектораz із розподілу шуму$p_z(z)$,$G(z)$ – згенеровані дані від генератора.

Друга частина $\log\left(1 - D\big(G(z)\big)\right)$ оцінює, наскільки дискримінатор розпізнає $G(z)$ як фальшиві. Функція втрат генератора:

$$L_G = -E_{z\sim p_z(z)}\big[log\ D\ \big(G(z)\big)\big],$$

залежить від того, як D оцінює $G(z)$, стимулюючи генератор обманювати дискримінатор. Цей підхід став основою для генерації реалістичного контенту, однак початкова реалізація GAN зіткнулася з низкою проблем. 1) Колапс моделі, колигенератор починає генерувати обмежений набір схожих зразків замість різноманітних реалістичних зображень. 2) Нестабільність навчання між L_D і L_G, коли генератор і дискримінатор не досягали балансу, що призводило до низької якості результатів моделі. Ці недоліки зумовили потребу в удосконаленні архітектури, що дало початок подальшим розробкам, спрямованим на підвищення стабільності та якості синтезованих даних.

Подальший розвиток GAN привів до появи DCGAN (DeepConvolutional GAN), яка інтегрувала згорткові нейронні мережі в структуру генератора та дискримінатора. Такий підхід був викликаний необхідністю покращити обробку зображень, оскільки базові GAN, що використовували повнозв'язні шари, не могли ефективно вловлювати просторові закономірності в даних. DCGAN вирішила проблему низької якості зображень, забезпечивши більш деталізовані та реалістичні результати завдяки згортковим нейронним мережам (CNN) [6], які краще підходять для роботи з візуальним контентом. Однак нестабільність навчання залишилася суттєвим недоліком, оскільки процес все ще залежав від тонкого налаштування параметрів і часто призводив до ситуацій, коли генератор продукував одноманітні або неякісні зображення. Це підштовхнуло до пошуку рішень, які б усунули фундаментальні причини нестабільності GAN.

Наступним етапом стало введення WGAN (Wasserstein GAN) [7], [8], яка змінила підхід до оцінки різниці між справжніми та згенерованими даними. Основною проблемою попередніх моделей, включно з DCGAN, була функція втрат, яка базувалася на бінарній класифікації та часто призводила до зникнення градієнтів або неінформативного навчання генератора. WGAN замінила цю функцію на метрику відстані Вассерштейна [7], що дозволила оцінювати відстань між розподілами даних більш плавно та стабільно. Відстань Вассерштейна (WL) показує, наскільки далеко один від одного перебувають розподіли реальних і згенерованих даних, і рахується як:

$$max \rightarrow WL = real - fake.$$

Тобто чим більша ця відстань, тим краще. Під час навчання GAN треба використовувати мінімізацію функції втрат:

$$min \rightarrow -WL = fake - real.$$

Це вирішило проблему нестабільності навчання, зробивши процес менш чутливим до вибору гіперпараметрів і покращивши якість генерації.

В WGAN дискримінативна нейронна мережа не містить функції активації, зокрема $sigmoid$,на останньому шарі нейромережі, що обмежує дані на виході в діапазоні $[0, 1]$, тому що на виході нейромережі з WGAN архітектурою мають повертатись дані в діапазоні $[-\infty, \infty]$. Саме тому в WGAN дискримінативна модель називається критик C, а її функція втрат:

$$L_C = E_{z\sim p_z(z)}\big[C\big(G(z)\big)\big] - E_{x\sim p_{data}(x)}[C(x)],$$

де $C(x)$ – оцінка реальних даних x, $C\big(G(z)\big)$ – оцінка згенерованих даних $G(z)$.Різниця між ними максимізує відстань між розподілом реальних даних$p_{data}(x)$ і розподілом згенерованих даних $G(z)$.

Функція втрат генератора:

$$L_G = -E_{z\sim p_z(z)}\big[C\big(G(z)\big)\big],$$

мінімізує значення$C\big(G(z)\big)$, наближаючи згенерований розподіл до реального. На відміну від L_D у GAN, яка використовувала логарифми, L_C забезпечила стабільність, уникаючи зникнення градієнтів. Однак в роботі [7] для коректності C вимагається обмеження ваг через обрізання до значень $[-0.01, 0.01]$ задля збереження обмеження 1-Lipschitz [9], що дозволило спрощувати C і знижувати якість $G(z)$, створюючи потребу в оптимізації.Крім того, автори прийшли до висновку, що для цієї мети також краще використовувати декілька оновлень вагів критика на одне оновлення оновлення генератора.

Для подолання обмежень WGAN була розроблена WGAN-GP (Wasserstein GAN withGradientPenalty)[10], яка замінила обрізання ваг на штраф за градієнт. Функція втрат генератора залишається такою самою, як і в WGAN, тоді як функція втрат критика стала:

$$L_C = E_{z\sim p_z(z)}\big[C\big(G(z)\big)\big] - E_{x\sim p_{data}(x)}[C(x)] + \lambda E_{\hat{x}\sim p_{\hat{x}}}[(\|\nabla_{\hat{x}} C(\hat{x})\|_2 - 1)^2],$$

де $\hat{x}$ – проміжні дані між x і $G(z)$,$\nabla_{\hat{x}} C(\hat{x})$– градієнт D за $\hat{x}$,$\|\cdot\|_2$ – норма, λ – коефіцієнт штрафу, що регулює відхилення градієнта від 1.

Штраф градієнту $\lambda E_{\hat{x}\sim p_{\hat{x}}}[(\|\nabla_{\hat{x}} C(\hat{x})\|_2 - 1)^2]$ усунув проблему спрощення критика, підвищивши адаптивність до складних розподілів даних $p_{data}(x)$ і якість $G(z)$. Ця модифікація була викликана тим, що обрізання ваг у WGAN часто викликає втрату виразності дискримінатора або його надмірну простоту, що може негативно впливати на якість згенерованих зображень. Введення градієнтного штрафу дозволило зберегти стабільність навчання, одночасно підвищивши здатність моделі адаптуватися до складних розподілів даних. Для цієї архітектури не потрібне використовування обрізання вагів критика, але все ще є необхідність до декількох оновлень критика на одне оновлення генератора. Крім того, як зазначають автори, штраф градієнту робить використання батч-нормалізації в шарах нейронної мережі критика зайвим, що спрощує його архітектуру. Таким чином WGAN-GP стала логічним продовженням еволюції GAN і WGAN, вирішивши проблему балансу між стабільністю та якістю, що зробило її ефективним інструментом для генерації високоякісного контенту.

Розвиток від GAN до DCGAN, WGAN і WGAN-GP відображає поступове усунення ключових недоліків: низької якості зображень, нестабільності навчання та обмеженої адаптивності моделі.

SteganoGAN

У роботі [11] наведено модель SteganoGAN –генеративно-змагальну мережу, призначену для приховування двійкових даних у зображеннях, із детальним описом нотації, архітектури та процесу навчання. Стеганографія в цій моделі зводиться до двох основних операцій: кодування, яке вбудовує двійкове повідомлення в зображення-носій, та декодування, яке витягує це повідомлення зі стегозображення. Припустимо $C \in \mathbb{R}^{3\times W\times H}$ – зображення-обкладинка, а S–стегозображення, обидва є RGB-зображеннями розміром$W \times H$, і $M \in \{0,1\}^{D\times W\times H}$– повідомлення у двійковому виді з глибиною D, що визначає верхню межу відносного корисного навантаження. Фактична ємність, як зазначено в [11], обчислюється як $(1-2p)D$, де $p \in [0,1]$ – частота помилок декодування. Модель базується на WGAN, де C є вибіркою з розподілу природних зображень $\mathbb{P}_C$, S генерується кодером $\mathcal{E}(C,M)$, а M відновлюється декодером $\mathcal{D}(S)$.

Архітектура SteganoGAN складається з трьох основних модулів: кодера, декодера та критика [11] і показано на рис. 2.

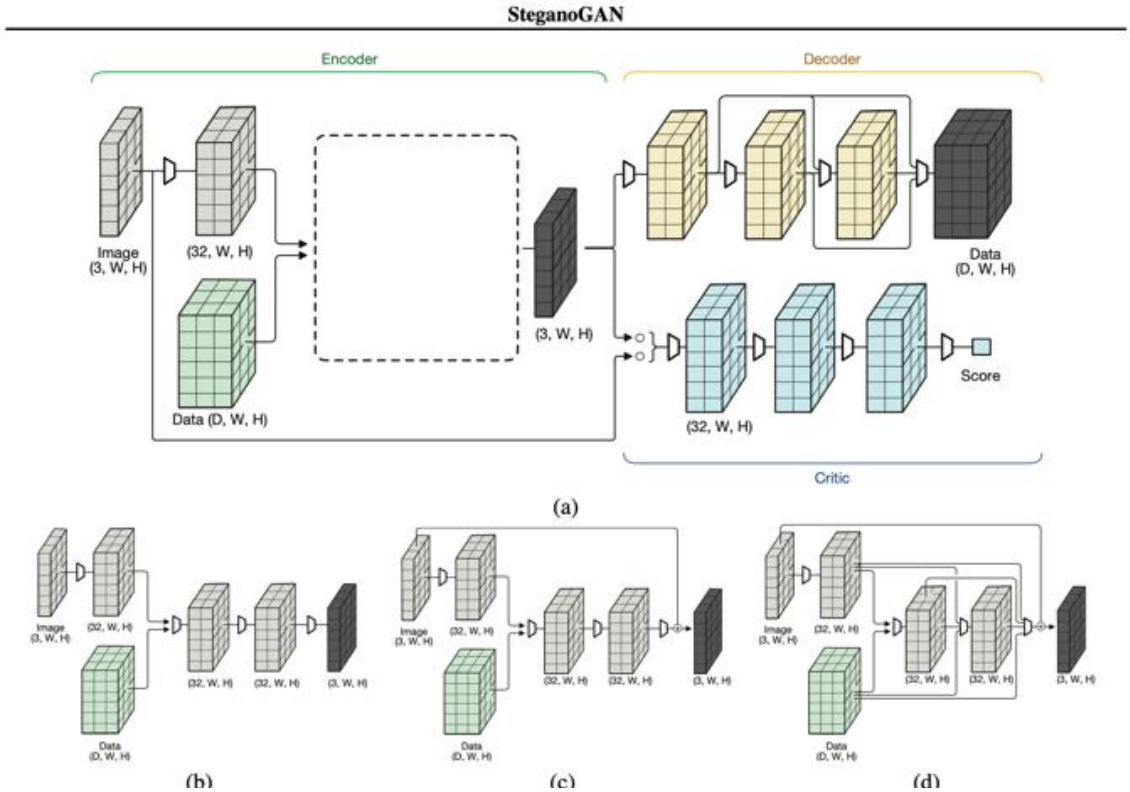

Рис. 2. Архітектура SteganoGAN [11]

Кодер $\mathcal{E}$ приймає $C \in \mathbb{R}^{3\times W\times H}$ та $M \in \{0,1\}^{D\times W\times H}$, об'єднуючи їх операцією конкатенації. $\mathrm{Cat}: (X,M) \to \Phi \in \mathbb{R}^{(D+D')\times W\times H}$ вздовж осі глибини, де $X \in \mathbb{R}^{D\times W\times H}$ і $Y \in \mathbb{R}^{D'\times W\times H}$ два тензори однакової ширини та висоти, але потенційно різної глибини, D і D'. Далі застосовуються згорткові блоки $\mathrm{Conv}_{D\to D'}: X \in \mathbb{R}^{D\times W\times H} \to \Phi \in \mathbb{R}^{D'\times W\times H}$, кожен із яких включає згортку з ядром 3, кроком 1, "same" заповненням, за якою слідують LeakyReLU та Batch Normalization (крім останнього блоку). Вихід кодера–S, наближене до C за розподілом. Декодер $\mathcal{D}$ обробляє S через подібні згорткові блоки, завершуючись операцією$\mathrm{Mean}: X \in \mathbb{R}^{D\times W\times H} \to \mathbb{R}^{D}$, яка обчислює середнє по $W \times H$ для відновленняM.

SteganoGAN пропонує три варіанти кодера: Basic, Residual і Dense, які різняться структурою та зв'язками між згортковими блоками.

Кожна з архітектур кодера починається з двох кроків:

1. Обробка зображення обкладинки S згортковим блоком для отримання тензора a:

$$a = Conv_{3\to 32}(C).$$

2. Конкатенація повідомлення M до a а потім обробка результату блоком згортки для отримання тензора b:

$$b = Conv_{32+D\to 32}\big(Cat(a, M)\big).$$

Basic архітектура починається з обробки C згортковим блоком для отримання проміжного тензора, до якого конкатенується M. Далі застосовуються два послідовні згорткові блоки, що генерують S як прямий результат останнього блоку, подібно до підходу в [12].

$$\mathcal{E}_b(C, M) = Conv_{32\to 3}\big(Conv_{32\to 32}(b)\big).$$

Ця проста структура зосереджена на базовому вбудовуванні даних без додаткових зв'язків.

Residual модифікує Basic, додаючи C до виходу кодера, щоб модель навчалася створювати залишкове (residual) зображення, як у роботі [13], підвищуючи стійкість і якість S.

$$\mathcal{E}_r(C, M) = C + \mathcal{E}_b(C, M).$$

Dense вводить щільні зв'язки [14] між згортковими блоками, об'єднуючи карти ознак від попередніх шарів із наступними, що сприяє повторному використанню ознак і покращенню швидкості вбудовування.

$$\begin{cases} c = Conv_{64+D\to 32}\big(Cat(a, b, M)\big) \\ d = Conv_{96+D\to 3}\big(Cat(a, b, c, M)\big) \\ \mathcal{E}_d(C, M) = C + d \end{cases}.$$

Нарешті, результатом кожного варіанту є стеганографічне зображення $S = \mathcal{E}_{\{b,r,d\}}(C, M)$ що має таку саму роздільну здатність і глибину, як і зображення обкладинки C.

Декодер обробляє стеганографічне зображення S через послідовність згорткових блоків із конкатенацією проміжних тензорів.

$$\begin{cases} a = Conv_{3\to 32}(S) \\ b = Conv_{32\to 32}(a) \\ c = Conv_{64\to 32}\big(Cat(a, b)\big) \\ \mathcal{D}(S) = Conv_{96\to D}\big(Cat(a, b, c)\big) \end{cases}.$$

Декодер видає $\widehat{M} = \mathcal{D}_d(S)$; іншими словами, він намагається відновити тензор даних M.

Критик складається з трьох згорткових блоків і одного вихідного шару, за яким слідує операція усереднення для скалярної оцінки S.

$$\begin{cases} a = Conv_{32\to 32}\Big(Conv_{32\to 32}\big(Conv_{3\to 32}(S)\big)\Big) \\ \mathcal{C}(S) = Mean\big(Conv_{32\to 1}(a)\big) \end{cases}.$$

Ці варіації SteganoGAN демонструють гнучкість архітектури, адаптуючи WGAN до задачі стеганографії з різними стратегіями обробки даних.

Для тренування нейронних мереж, SteganoGANвикористовує власну функцію втрат, що включає в себе функцію втрат парикодер-декодер $\mathcal{L}_{E,D}$, а такожфункцію втраткритика $\mathcal{L}_{Critic}$.

Для оптимізації мережі кодер-декодер автори [11] спільно оптимізовують три функції втрат.

1. Точність декодування за допомогою функції бінарної кросс-ентропії (BinaryCross-Entropy):

$$\begin{aligned} \mathcal{L}_d &= -\mathbb{E}_{X\sim\mathbb{P}_C} BinaryCrossEntropy\big(\mathcal{D}\big(\mathcal{E}(X, M)\big), M\big) \\ &= -\mathbb{E}_{X\sim\mathbb{P}_C}\Big[log\big(\mathcal{D}(M)\big) + log\Big(1 - \mathcal{D}\big(\mathcal{E}(X, M)\big)\Big)\Big], \end{aligned}$$

де M – оригінальне повідомлення в бінарному виді, X – зображення-обкладинка з$\mathbb{P}_C$

розподілу; $\mathcal{D}\big(\mathcal{E}(X,M)\big)$ – декодоване повідомлення зі стегозображення $\mathcal{E}(X,M)$.

2. Схожість між стеганографічним зображенням та зображенням обкладинки за середньо-квадратичною похибкою (Mean Squared Error):

$$\mathcal{L}_s = \mathbb{E}_{X\sim\mathbb{P}_C}\left[\frac{1}{3\times W\times H}\|X-\mathcal{E}(X,M)\|_2^2\right],$$

де W і H – ширина та висота зображення-обкладинки X, 3 – кількість кольорових каналів зображення.

3. Реалістичність стеганографічного зображенняза допомогою критика (Realness):

$$\mathcal{L}_r = \mathbb{E}_{X\sim\mathbb{P}_C}\big[\mathcal{C}\big(\mathcal{E}(X,M)\big)\big].$$

Слід зауважити що функція втрат реалістичності зображень не є від'ємною, що не збігає з подібною функцією мінімізації генераторав класичному WGAN.

Загальна функція втрат для пари кодер-декодер становитиме

$$\mathcal{L}_{E,D} = \alpha\cdot\mathcal{L}_d + \beta\cdot\mathcal{L}_s + \gamma\cdot\mathcal{L}_r,$$

де α,β,γ – це ваги для збалансування різних цілей. У випадку з SteganoGAN, параметри α і γдорівнюють одиниці, тоді як $\beta = 100$, це значить що при навчанні нейромережі кодера і декодера перевага надавалась саме на схожість між стеганографічним зображенням та зображенням обкладинки.

Щоб навчити критик, мінімізується втрати Вассерштейна

$$\mathcal{L}_c = \mathbb{E}_{X\sim\mathbb{P}_C}[\mathcal{C}(X)] - \mathbb{E}_{X\sim\mathbb{P}_C}\big[\mathcal{C}\big(\mathcal{E}(X,M)\big)\big].$$

Так само, як і в формулі (17), незрозумілим є використання невід'ємної форм функції втрат Вассерштейна в якості функції втрат для навчання критика.

Хоч SteganoGAN і заснований на WGAN, на відміну від класичної WGAN, де критик оновлюється кілька разів за ітерацію, SteganoGAN використовує одноразове оновлення критика $C(\cdot)$, який оцінює відстань $dis(\mathbb{P}_C,\mathbb{P}_S)$ між розподілами C і S. Крім того, для навчання використовується32 епохи, вирізання вагів критику в проміжку $[-0.1, 0.1]$, а не рекомендовані $[-0.01, 0.01]$, що робить алгоритм навчання таким, що показаний на рис.3.

Оригінальний Алгоритм Навчання SteganoGAN Параметри для навчання нейронної мережі: $\eta_{Critic} = \eta_{E,D} = 0.0001$; $c = 0.1$; $m = 4$; $k = 32$; $W = H = 360$; $D = 1,\dots,6$; $\alpha = 1$; $\beta = 100$; $\gamma = 1$.
Необхідно: η_{Critic} – швидкість навчання критику; $\eta_{E,D}$ – швидкість навчання енкодера-декодера; c – гіперпарамерт для вирізання ваг критика; m – розмір батчу; k – кількість епох; W, H – ширина і висота зображень датасету; D – глибина повідомлення, α,β,γ – learning rates for: декодування повідомлення, схожості зображення, реалістичності зображення.
Необхідно: $\theta_{Critic,0}$ – початкові параметри критика; $\theta_{E,D,0}$ – початкові параметри енкодера-декодера.

for k шагів **do**
- Зразок міні-батчу обкладинок $\{X^1, X^2, \dots, X^{(i)}\}_{i=1}^m \sim \mathbb{P}_C$ (розподіл реальних зображень).
- Зразок міні-батчу секретних повідомлень $\{M^1, M^2, \dots, M^{(i)}\}_{i=1}^m$ (бінарні значення).
- Генеруємо стего-зображення: $\mathcal{E}\big(X^{(i)}, M^{(i)}\big)$ для $i = 1,\dots,m$.
- $\mathcal{L}_{Critic} = \frac{1}{m}\sum_{i=1}^m \mathcal{C}\big(X^{(i)}\big) - \frac{1}{m}\sum_{i=1}^m \mathcal{C}\left(\mathcal{E}\big(X^{(i)}, M^{(i)}\big)\right)$
- $g_{\theta_{Critic}} \leftarrow \nabla_{\theta_{Critic}}[\mathcal{L}_{Critic}]$
- $\theta_{Critic} \leftarrow \theta_{Critic} - \eta_{Critic}\cdot Adam(\theta_{Critic}, g_{\theta_{Critic}})$
- $\theta_{Critic} \leftarrow clip(\theta_{Critic}, -c, c)$
- $\mathcal{L}_d = -\frac{1}{m}\sum_{i=1}^m \left(log\left(\mathcal{D}\big(M^{(i)}\big)\right) + log\left(1 - \mathcal{D}\left(\mathcal{E}\big(X^{(i)}, M^{(i)}\big)\right)\right)\right)$
- $\mathcal{L}_s = \frac{1}{m}\sum_{i=1}^m \left(\frac{1}{3\times W\times H}\left\|X^{(i)} - \mathcal{E}\big(X^{(i)}, M^{(i)}\big)\right\|_2^2\right)$
- $\mathcal{L}_r = \frac{1}{m}\sum_{i=1}^m \mathcal{C}\left(\mathcal{E}\big(X^{(i)}, M^{(i)}\big)\right)$
- $g_{\theta_{E,D}} \leftarrow \nabla_{\theta_{E,D}}[\alpha\cdot\mathcal{L}_d + \beta\cdot\mathcal{L}_s + \gamma\cdot\mathcal{L}_r]$
- $\theta_{E,D} \leftarrow \theta_{E,D} - \eta_{E,D}\cdot Adam(\theta_{E,D}, g_{\theta_{E,D}})$

end for

Рис. 3. Алгоритм навчання оригінальної SteganoGAN

Модифікації SteganoGAN на основі різних архітектур GAN

Оригінальна модель SteganoGAN, заснована на WGAN із одноразовим оновленням критика та нестандартними параметрами, такими як вирізання ваг у межах $[-0.1, 0.1]$ і акцент на схожості стегозображень із обкладинкою через $\beta = 100$, демонструє певну ефективність у приховуванні даних. Проте ці модифікації відходять від класичних підходів WGAN і можуть обмежувати стабільність, якість генерації чи стійкість до стегоаналізу. Для оцінки потенційного покращення стеганографічних характеристик у цьому дослідженні розроблено три варіанти SteganoGAN на основі архітектур DCGAN, WGAN і WGAN-GP. Мета полягає в порівнянні їхньої продуктивності з оригінальною моделлю за якістю стегозображень, ємністю корисного навантаження та стійкістю до виявлення. При цьому для WGAN і WGAN-GP використано методи навчання, наближені до оригінальних статей, зокрема багаторазове оновлення критика для WGAN і градієнтний штраф для WGAN-GP, тоді як DCGAN застосовує базову згорткову структуру для аналізу її впливу на задачу стеганографії.

Для кодеру обрано використовувати Dense архітектуру нейромережі, так як в роботі [11] вона показала найкращі результати за всіма метриками. Додатково покращено мережу критика для підвищення якості класифікації між зображеннями-обкладинками C та стегозображеннями S. Порівняно з оригінальною версією SteganoGAN, де критик складався з послідовності згорткових шарів із фіксованою глибиною та завершувався простим усередненням, запропонована архітектура включає шари:

1. $\mathrm{Dropout}_p : X \in \mathbb{R}^{D\times W\times H} \to \Phi \in \mathbb{R}^{D\times W\times H}$, де X і Φ – вхідний і вихідний тензори розміром $D \times W \times H$;p – ймовірність виключення (dropoutrate), наприклад, $p = 0.3$,

2. $\mathrm{MaxPool}_D : X \in \mathbb{R}^{D\times W\times H} \to \Phi \in \mathbb{R}^{D\times \frac{W}{\mathrm{S}}\times \frac{H}{\mathrm{S}}}$, де X – вхідний тензор розміром $D \times W \times H$; Φ – вихідний тензор розміром $D \times \frac{W}{\mathrm{S}} \times \frac{H}{\mathrm{S}}$;S – розмір кроку (stride) і розмір вікна (poolsize), наприклад, S=2 для 2×2 пулингу.

3. $\mathrm{GlobalAvgPool}_D : X \in \mathbb{R}^{D\times W\times H} \to \Phi \in \mathbb{R}^{D}$, де X – вхідний тензор розміром $D \times W \times H$;Φ – вихідний тензор розміром D;

Шари MaxPooling, додані після кожного згорткового блоку з глибинами 32, 64 і 128 каналів, зменшують просторові розміри тензорів (з $H \times W$ до $H/8 \times W/8$), що сприяє узагальненню ознак і зниженню обчислювальної складності, зберігаючи ключову інформацію для класифікації. Операція GlobalAveragePooling2D замінює усереднення по всіх пікселях на автоматичне обчислення середнього значення по кожному каналу. GlobalAveragePooling2D перетворює вихідний тензор у вектор фіксованої довжини, який подається на кінцевий Dense шар із сигмоїдною (sigmoid) активацією при DCGANархітектурі. При WGANі WGAN-GP функції активації на останньому шарі в такому випадку не буде. Архітектура такої нейромережі має вигляд:

$$\begin{cases} a = \mathrm{Dropout}\Big(\mathrm{MaxPool}\big(\mathrm{Conv}_{3\to 32}(S)\big)\Big) \\ b = \mathrm{Dropout}\Big(\mathrm{MaxPool}\big(\mathrm{Conv}_{32\to 64}(a)\big)\Big) \\ c = \mathrm{Dropout}\Big(\mathrm{MaxPool}\big(\mathrm{Conv}_{64\to 128}(b)\big)\Big) \\ C(S) = \mathrm{Conv}_{128\to 1}\big(\mathrm{GlobalAvgPool}(c)\big) \end{cases}.$$

Ці зміни підвищують здатність критика розрізняти C і S, що є критично важливим для оцінки реалістичності стегозображень у всіх трьох варіантах SteganoGAN.

DCGAN SteganoGAN

У варіанті SteganoGAN на основі DCGAN використано згорткові нейронні мережі для кодера, декодера та дискримінатора, зберігаючи базову структуру оригінальної моделі, але замінюючи метрику Вассерштейна на бінарну крос-ентропію для функцій втрат.

На відміну від оригінального SteganoGAN, де критик $C(\cdot)$ оптимізувався через втрати Вассерштейна для оцінки $dis(\mathbb{P}_C, \mathbb{P}_S)$, тут змагальна втрата (Adversarial Loss) для кодера адаптована до бінарної крос-ентропії, стимулюючи генерацію стегозображеньS, які дискримінатор $\mathcal{D}isc(\cdot)$ сприймає як реальні:

$$\mathcal{L}_{Adv} = -\mathbb{E}_{X\sim\mathbb{P}_C}\left[log\left(\mathcal{D}isc\big(\mathcal{E}(X,M)\big)\right)\right].$$

Інші дві функції втрат для кодера-декодера $\mathcal{L}_d$ і $\mathcal{L}_s$ за формулами (15) і (16) будемо тепер обозначати через $\mathcal{L}_{BCE}$ і $\mathcal{L}_{MSE}$. В такому випадку загальна функція втрат для пари кодер-декодер має вигляд:

$$\mathcal{L}_{E,D} = \alpha\cdot\mathcal{L}_{BCE} + \beta\cdot\mathcal{L}_{MSE} + \gamma\cdot\mathcal{L}_{Adv}.$$

Функція втрат дискримінатора (DiscriminatorLoss) відповідно перероблена для класифікації реальних зображеньC та стегозображеньS за допомогою цієї ж метрики, замінюючи оцінку відстані між розподілами на задачу бінарної класифікації:

$$\begin{aligned}\mathcal{L}_{Disc} &= -\mathbb{E}_{X\sim\mathbb{P}_C} BinaryCrossEntropy(\mathcal{E}(X,M),X)\\ &= -\mathbb{E}_{X\sim\mathbb{P}_C}\left[log\big(\mathcal{D}isc(X)\big) + log\left(1-\mathcal{D}isc\big(\mathcal{E}(X,M)\big)\right)\right].\end{aligned}$$

Ця заміна повертає алгоритм навчання SteganoGAN до підходу класичних GAN і DCGAN, усуваючи потребу в обмеженні ваг чи градієнтному штрафі, як у WGAN і WGAN-GP. Алгоритм навчання DCGAN SteganoGAN показаний на рис. 4.

Алгоритм 1 Навчання SteganoGAN на основі DCGAN. Параметри для навчання нейронної мережі: $\eta_{Critic} = \eta_{E,D} = 0.0001$; $m = 4$; $k = 32$; $W = H = 360$; $D = 1,\dots,6$; $\alpha = 1$; $\beta = 100$; $\gamma = 0.01$.

Необхідно: η_{Disc} – швидкість навчання дискримінатора; $\eta_{E,D}$ – швидкість навчання енкодера-декодера; m – розмір батчу; k – кількість епох; W, H – ширина і висота зображень датасету; D – глибина повідомлення, α, β, γ – learning rates for: декодування повідомлення, схожості зображення, реалістичності зображення.

Необхідно: $\theta_{Disc,0}$ – початкові параметри дискримінатора; $\theta_{E,D,0}$ – початкові параметри енкодера-декодера.

for k шагів **do**

Зразок міні-батчу обкладинок $\{X^1, X^2, \dots, X^{(i)}\}_{i=1}^m \sim \mathbb{P}_C$ (розподіл реальних зображень).

Зразок міні-батчу секретних повідомлень $\{M^1, M^2, \dots, M^{(i)}\}_{i=1}^m$ (бінарні значення).

Генеруємо стего-зображення: $\mathcal{E}\big(X^{(i)}, M^{(i)}\big)$ для $i = 1,\dots,m$.

$\mathcal{L}_{Disc} = -\frac{1}{m}\sum_{i=1}^m \left(log\left(\mathcal{D}isc\big(X^{(i)}\big)\right) + log\left(1-\mathcal{D}isc\left(\mathcal{E}\big(X^{(i)}, M^{(i)}\big)\right)\right)\right)$

$g_{\theta_{Disc}} \leftarrow \nabla_{\theta_{Disc}}[\mathcal{L}_{Disc}]$

$\theta_{Disc} \leftarrow \theta_{Disc} - \eta_{Disc}\cdot Adam(\theta_{Disc}, g_{\theta_{Disc}})$

$\mathcal{L}_{BCE} = -\frac{1}{m}\sum_{i=1}^m \left(log\left(\mathcal{D}\big(M^{(i)}\big)\right) + log\left(1-\mathcal{D}\left(\mathcal{E}\big(X^{(i)}, M^{(i)}\big)\right)\right)\right)$

$\mathcal{L}_{MSE} = \frac{1}{m}\sum_{i=1}^m \left(\frac{1}{3\times W\times H}\left\|X^{(i)} - \mathcal{E}\big(X^{(i)}, M^{(i)}\big)\right\|_2^2\right)$

$\mathcal{L}_{Adv} = -\frac{1}{m}\sum_{i=1}^m \left(log\left(\mathcal{D}isc\left(\mathcal{E}\big(X^{(i)}, M^{(i)}\big)\right)\right)\right)$

$g_{\theta_{E,D}} \leftarrow \nabla_{\theta_{E,D}}[\alpha\cdot\mathcal{L}_{BCE} + \beta\cdot\mathcal{L}_{MSE} + \gamma\cdot\mathcal{L}_{Adv}]$

$\theta_{E,D} \leftarrow \theta_{E,D} - \eta_{E,D}\cdot Adam(\theta_{E,D}, g_{\theta_{E,D}})$

end for

Рис. 4. Алгоритм навчання DCGAN SteganoGAN

WGAN SteganoGAN

Для роботи з оригінальною структурою WGAN використовуються імплементовані до WGAN SteganoGAN функції втрат, а саме від'ємна форма для функції втрат реалістичності стеганографічного зображення за допомогою критика:

$$\mathcal{L}_{Realness} = -\mathbb{E}_{X\sim\mathbb{P}_C}\big[\mathcal{C}\big(\mathcal{E}(X,M)\big)\big],$$

а також від'ємна форма втрат Вассерштейна для навчання критика:

$$\mathcal{L}_c = \mathbb{E}_{X\sim\mathbb{P}_C}\left[\mathcal{C}\left(\mathcal{E}(X,M)\right)\right] - \mathbb{E}_{X\sim\mathbb{P}_C}[\mathcal{C}(X)].$$

Ідентично до DCGANSteganoGAN, використовуються такі ж самі значення ваг α,β,γ в функції втрат кодера-декодера, тоді як замість змагальної втрати $\mathcal{L}_{Adv}$ використовується втрата реалістичності $\mathcal{L}_{Realness}$.

$$\mathcal{L}_{E,D} = \alpha\cdot\mathcal{L}_{BCE} + \beta\cdot\mathcal{L}_{MSE} + \gamma\cdot\mathcal{L}_{Realness}\ .$$

Алгоритм навчання WGAN SteganoGAN також як і оригінальний алгоритм WGAN використовує декілька оновлень значень вагів критику на одне оновлення значень вагів кодеру та декодеру, і показаний на рис. 5.

Алгоритм 2 Навчання SteganoGAN на основі WGAN. Параметри для навчання нейронної мережі: $\eta_{Critic} = \eta_{E,D} = 0.0001$; $n_{critic} = 5$; $c = 0.01$; $m = 4$; $k = 32$; $W = H = 360$; $D = 1,\ldots,6$; $\alpha = 1; \beta = 100; \gamma = 0.01$.

Необхідно: η_{Critic} – швидкість навчання критику; $\eta_{E,D}$ – швидкість навчання енкодера-декодера; n_{critic} – кількість оновлень критика; c – гіперпарамерт для вирізання ваг критика; m – розмір батчу; k – кількість епох; W, H – ширина і висота зображень датасету; D – глибина повідомлення, α,β,γ – learning rates for: декодування повідомлення, схожості зображення, реалістичності зображення.

Необхідно: $\theta_{Critic,0}$ – початкові параметри критика; $\theta_{E,D,0}$ – початкові параметри енкодера-декодера.

for k шагів **do**
  Зразок міні-батчу обкладинок $\{X^1, X^2, \ldots, X^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_C$ (розподіл реальних зображень).
  Зразок міні-батчу секретних повідомлень $\{M^1, M^2, \ldots, M^{(i)}\}_{i=1}^{m}$ (бінарні значення).
  Генеруємо стего-зображення: $\mathcal{E}(X^{(i)}, M^{(i)})$ для $i = 1,\ldots,m$.
  for n_{critic} шагів **do**
    $\mathcal{L}_{Critic} = \frac{1}{m}\sum_{i=1}^{m}\mathcal{C}\left(\mathcal{E}(X^{(i)}, M^{(i)})\right) - \frac{1}{m}\sum_{i=1}^{m}\mathcal{C}(X^{(i)})$
    $g_{\theta_{Critic}} \leftarrow \nabla_{\theta_{Critic}}[\mathcal{L}_{Critic}]$
    $\theta_{Critic} \leftarrow \theta_{Critic} - \eta_{Critic}\cdot Adam(\theta_{Critic}, g_{\theta_{Critic}})$
    $\theta_{Critic} \leftarrow clip(\theta_{Critic}, -c, c)$
  end for
  $\mathcal{L}_{BCE} = -\frac{1}{m}\sum_{i=1}^{m}\left(log\left(\mathcal{D}(M^{(i)})\right) + log\left(1-\mathcal{D}\left(\mathcal{E}(X^{(i)}, M^{(i)})\right)\right)\right)$
  $\mathcal{L}_{MSE} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{3\times W\times H}\left\|X^{(i)} - \mathcal{E}(X^{(i)}, M^{(i)})\right\|_2^2\right)$
  $\mathcal{L}_{Realness} = -\frac{1}{m}\sum_{i=1}^{m}\mathcal{C}\left(\mathcal{E}(X^{(i)}, M^{(i)})\right)$
  $g_{\theta_{E,D}} \leftarrow \nabla_{\theta_{E,D}}[\alpha\cdot\mathcal{L}_{BCE} + \beta\cdot\mathcal{L}_{MSE} + \gamma\cdot\mathcal{L}_{Realness}]$
  $\theta_{E,D} \leftarrow \theta_{E,D} - \eta_{E,D}\cdot Adam(\theta_{E,D}, g_{\theta_{E,D}})$
end for

Рис. 5. Алгоритм навчання WGANSteganoGAN

WGAN-GP SteganoGAN

Так само, як і в WGAN SteganoGAN, отримуємо значення $\mathcal{L}_{BCE}$, $\mathcal{L}_{MSE}$ і $\mathcal{L}_{Realness}$ і розраховуємо загальну функція втрат для пари кодер-декодер $\mathcal{L}_{E,D}$.

Критик у WGAN-GP оцінює відстань Вассерштейна між розподілами реальних зображень обкладинок X та стегозображення $\mathcal{E}(X,M)$ з додаванням штрафу градієнта:

$$\mathcal{L}_{Critic} = \mathbb{E}_{X\sim\mathbb{P}_C}\left[\mathcal{C}\left(\mathcal{E}(X,M)\right)\right] - \mathbb{E}_{X\sim\mathbb{P}_C}[\mathcal{C}(X)] + \lambda\mathbb{E}_{\hat{X}\sim\mathbb{P}_{\hat{X}}}\left[\left(\left\|\nabla_{\hat{X}}\mathcal{C}(\hat{X})\right\|_2 - 1\right)^2\right],$$

де $\lambda\mathbb{E}_{\hat{X}\sim\mathbb{P}_X}\left[\left(\left\|\nabla_{\hat{X}}\mathcal{C}(\hat{X})\right\|_2 - 1\right)^2\right]$ – штраф градієнту, що виконує обмеження 1-Lipschitz, штрафуючи критика, якщо норма градієнта відхиляється від 1,$\hat{X}$ – це випадкова інтерполяція між реальними зображеннями обкладинками X та стегозображення $\mathcal{E}(X,M)$ що визначається як $\hat{X} = \epsilon X + (1-\epsilon)\cdot\mathcal{E}(X,M)$ де $\epsilon \sim U[0,1]$, λ – градієнтний штрафний коефіцієнт (наприклад, $\lambda = 10$).

Алгоритм навчання WGAN-GP SteganoGAN наведений на рис. 6.

Алгоритм 3 Training SteganoGAN на основі WGAN-GP. Параметри для навчання нейронної мережі: $\eta_{Critic} = \eta_{E,D} = 0.0001$; $n_{critic} = 5$; $\lambda = 10$; $m = 4$; $k = 32$; $W = H = 360$; $D = 1,\ldots,6$; $\alpha = 1$; $\beta = 100$; $\gamma = 0.01$.

Необхідно: η_{Critic} – швидкість навчання критику; $\eta_{E,D}$ – швидкість навчання енкодера-декодера; n_{critic} – кількість оновлень критика; λ – коефіцієнт градієнтного пенальті; m – розмір батчу; k – кількість епох; W, H – ширина і висота зображень датасету; D – глибина повідомлення, α, β, γ – learning rates for: декодування повідомлення, схожості зображення, реалістичності зображення.

Необхідно: $\theta_{Critic,0}$ – початкові параметри критика; $\theta_{E,D,0}$ – початкові параметри енкодера-декодера.

for k шагів **do**

Зразок міні-батчу обкладинок $\{X^1, X^2, \ldots, X^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_C$ (розподіл реальних зображень).

Зразок міні-батчу секретних повідомлень $\{M^1, M^2, \ldots, M^{(i)}\}_{i=1}^{m}$ (бінарні значення).

Генеруємо стего-зображення: $\mathcal{E}(X^{(i)}, M^{(i)})$ для $i = 1, \ldots, m$.

  for n_{critic} шагів **do**

    $\hat{X}^{(i)} = \epsilon X^{(i)} + (1 - \epsilon^{(i)}) \cdot \mathcal{E}(X^{(i)}, M^{(i)})$, де $\epsilon^{(i)} \sim U[0,1]$

    $\mathcal{L}_{Critic} = \frac{1}{m}\sum_{i=1}^{m} \mathcal{C}\left(\mathcal{E}(X^{(i)}, M^{(i)})\right) - \frac{1}{m}\sum_{i=1}^{m} \mathcal{C}(X^{(i)}) + \frac{\lambda}{m}\sum_{i=1}^{m}\left(\|\nabla_{\hat{X}^{(i)}}\mathcal{C}(\hat{X}^{(i)})\|_2 - 1\right)^2$

    $g_{\theta_{Critic}} \leftarrow \nabla_{\theta_{Critic}}[\mathcal{L}_{Critic}]$

    $\theta_{Critic} \leftarrow \theta_{Critic} - \eta_{Critic} \cdot Adam(\theta_{Critic}, g_{\theta_{Critic}})$

  end for

$\mathcal{L}_{BCE} = -\frac{1}{m}\sum_{i=1}^{m}\left(log\left(\mathcal{D}(M^{(i)})\right) + log\left(1 - \mathcal{D}\left(\mathcal{E}(X^{(i)}, M^{(i)})\right)\right)\right)$

$\mathcal{L}_{MSE} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{3\times W\times H}\|X^{(i)} - \mathcal{E}(X^{(i)}, M^{(i)})\|_2^2\right)$

$\mathcal{L}_{Realness} = -\frac{1}{m}\sum_{i=1}^{m} \mathcal{C}\left(\mathcal{E}(X^{(i)}, M^{(i)})\right)$

$g_{\theta_{E,D}} \leftarrow \nabla_{\theta_{E,D}}[\alpha \cdot \mathcal{L}_{BCE} + \beta \cdot \mathcal{L}_{MSE} + \gamma \cdot \mathcal{L}_{Realness}]$

$\theta_{E,D} \leftarrow \theta_{E,D} - \eta_{E,D} \cdot Adam(\theta_{E,D}, g_{\theta_{E,D}})$

end for

Рис. 6. Алгоритм навчання WGAN-GP SteganoGAN

Результати навчання моделей і метрична оцінка

Реалізація нейронних мереж для трьох варіантів SteganoGAN – на основі DCGAN, WGAN і WGAN-GP – виконана з використанням фреймворку TensorFlow, що забезпечило гнучкість у налаштуванні архітектур і процесі навчання. Навчання проводилось на дата сеті DIV2K [15], розмір зображень для навчання $W = H = 360$ пікселів. Для навчання нейромережвикористовувався алгоритм оптимізації градієнтного спуску Adam, з швидкістю навчання $\alpha = 1e^{-4}$. Кожна з трьох архітектур була навчена на рівнях $D = [1,\ldots,6]$, а кількість епох становила 32.

Зважаючи на значну складність архітектур розроблених модифікацій SteganoGAN, навчання проводилось на T4 GPU онлайн середи розробки Google Colaboratory.

На рис. 7 показано оригінальне зображення, а також стегозображення з рівнем $D = 2$, зроблені за допомогою: класичного стеганографічного методу LSB, оригінальної SteganoGAN, а також варіаціями SteganoGAN описаними в цій статті. До кожного стегозображення внизу наведено різницю між оригінальним зображенням, і цим стегозображенням.

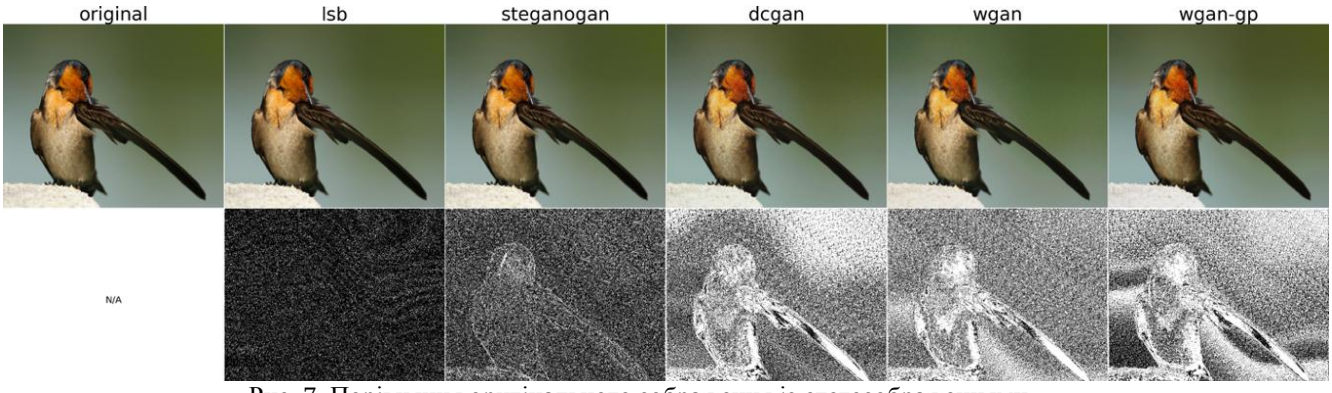

Рис. 7. Порівняння оригінального зображення із стегозображеннями

Для функції втрат кодера-декодера, яка комбінує точність декодування, схожість стего-зображень із обкладинкою та змагальну складову, було використано коефіцієнти $\beta = 100$ і $\gamma = 0.01$, тоді як α залишався стандартним. Вибір $\beta = 100$ відповідає оригінальному SteganoGAN, надаючи значну вагу схожості між стегозображенням і зображенням-обкладинкою, що сприяє збереженню візуальної якості. Значення $\gamma = 0.01$ для змагальної складової було визначено експериментально, оскільки за більших значень γ (наприклад, 1 або 0.1) нейронна мережа демонструвала нестабільність або повну нездатність до навчання, що проявлялося у розбіжності втрат або низькій якості стегозображень. Зменшення γ до 0.01 дозволило збалансувати вплив змагальної компоненти, забезпечуючи поступове вдосконалення генерації без перевантаження моделі, що особливо важливо для DCGAN із бінарною крос-ентропією та для WGAN і WGAN-GP із їхніми методами стабілізації.

Для оцінки продуктивності трьох варіантів SteganoGAN на основі DCGAN, WGAN і WGAN-GP використано чотири метрики, які дозволяють комплексно проаналізувати якість стегозображень, ємність прихованих даних і точність їх відновлення.

RS-BPP (ReedSolomonBit-Per-Pixel) вимірює ефективну ємність корисного навантаження, тобто кількість бітів інформації, яку модель може надійно приховати в стегозображенні на один піксель, враховуючи частоту помилок декодування. Ця метрикастворена авторами статті [11] і залежить від заданої глибини даних D, яка визначає максимальну кількість бітів на піксель (bpp), що теоретично можна приховати, і частоти помилок p, яка відображає ймовірність неправильного декодування окремого біта в повідомленні M. Частота помилок p обчислюється як відношення кількості неправильно декодованих бітів до загальної кількості бітів у M, і залежить від якості декодера $\mathcal{D}$, а також від впливу шуму, спричиненого вбудовуванням даних у зображення C. RS-BPP дозволяє оцінити реальну ємність моделі, оскільки не всі біти з D можуть бути коректно відновлені через помилки, і чим нижчий p, тим більша ефективна ємність. Наприклад, при $p = 0$ (ідеальне декодування) RS-BPP дорівнює D, а при $p = 0.5$ (випадкове декодування) RS-BPP падає до 0. Ця метрика є ключовою для оцінки стеганографічних систем, особливо за високих значень D, де баланс між ємністю та точністю стає критичним. Формула для RS-BPP має вигляд:

$$RS - BPP = (1 - 2p) \cdot D,$$

де $p \in [0,1]$– частота помилок декодування,D– задана глибина даних у бітах на піксель.

SSIM (Structural Similarity Index) оцінює структурну схожість між зображенням-обкладинкою C і стегозображенням S, враховуючи сприйняття змін у яскравості, контрасті та текстурі, що критично для забезпечення непомітності прихованих даних. На відміну від піксельних метрик, SSIM імітує людське сприйняття, порівнюючи локальні характеристики зображень, і повертає значення в діапазоні $[-1,1]$, де 1 означає ідеальну схожість. Ця метрика є важливою для оцінки того, наскільки стегозображення залишається візуально подібним до оригіналу, уникаючи підозр при аналізі. Формула SSIM визначається як:

$$SSIM_{X,Y} = \frac{(2\mu_X\mu_Y + c_1) \cdot (2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1) \cdot (\sigma_X^2 + \sigma_Y^2 + c_2)},$$

де μ_X – середнє значення X, μ_Y – середнє значення Y,σ_X^2 – дисперсія X,σ_Y^2 – дисперсія Y, σ_{XY} – коваріантністьXiY,$c1 = (k_1L)^2$,$c2 = (k_2L)^2$ – дві змінні, в яких $k_1 = 0.01$, $k_1 = 0.03$, L – динамічний діапазон значень пікселів, що за використання 8-бітного зображення дорівнює 255.

PSNR (Peak Signal-to-Noise Ratio)вимірює співвідношення між максимальною можливою потужністю сигналу та потужністю шуму, що впливає на якість стегозображення S, і слугує показником візуальних спотворень. PSNR обчислюється через середньоквадратичну похибку (*MSE*) між C і S, і вищі значення вказують на менші спотворення, що є важливим для забезпечення якості зображення після вбудовування даних. Ця метрика часто

використовується для оцінки стеганографічних систем, оскільки вона відображає, наскільки зміни в пікселях впливають на загальну якість. Формула PSNR має вигляд:

$$MSE = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}\left(X_{i,j} - Y_{i,j}\right)^2,$$

$$PSNR = 20 \cdot log_{10}(sc) - 10 \cdot log_{10}(MSE),$$

де sc– максимальне значення пікселя (наприклад, 255 для 8-бітних зображень), MSE – середньоквадратична похибка між зображеннями X і Y з висотою H і шириною W.

Decoding Accuracy визначає частку правильно відновлених бітів прихованого повідомлення M декодером $\mathcal{D}$ із стегозображення S, відображаючи здатність моделі точно передавати інформацію. Ця метрика є ключовою для оцінки ефективності стеганографічної системи, оскільки низька точність декодування може зробити приховані дані непридатними для використання, навіть якщо візуальна якість S висока. Decoding Accuracy обчислюється як відношення кількості правильно декодованих бітів до загальної кількості бітів у M, і значення ближче до 1 вказують на кращу продуктивність. Decoding Accuracy обчислюється як:

$$Decoding\ Accuracy = \frac{1}{N}\sum_{i=0}^{N-1}\mathbb{I}\left(M_i = \widehat{M}_i\right),$$

де N – загальна кількість бітів у повідомленні M, $\widehat{M} = \mathcal{D}(S)$ – декодоване повідомлення, $\mathbb{I}\left(M_i = \widehat{M}_i\right)$ – індикаторна функція, що дорівнює 1, якщо $M_i = \widehat{M}_i$, і 0 інакше.

Результати навчання трьох варіантів SteganoGAN на основі DCGAN, WGAN і WGAN-GP з $D \in [1,..,6]$ наведено в таблиці 1.

Таблиця 1

Порівняння метрик розроблених архітектур SteganoGAN

Model	D	DecoderAccuracy	RS-BPP	PSNR	SSIM
DCGAN	1	0,97	0,94	31,45	0,92
	2	0,92	1,70	31,24	0,91
	3	0,78	1,71	32,18	0,92
	4	0,69	1,55	32,29	0,93
	5	0,65	1,56	32,97	0,94
	6	0,64	1,69	32,59	0,94
WGAN	1	0,97	0,94	31,24	0,92
	2	0,92	1,69	31,07	0,90
	3	0,77	1,64	31,72	0,92
	4	0,69	1,54	31,66	0,92
	5	0,65	1,57	32,27	0,93
	6	0,63	1,64	32,48	0,94
WGAN-GP	1	0,96	0,92	32,49	0,94
	2	0,90	1,61	31,65	0,92
	3	0,75	1,51	31,98	0,93
	4	0,68	1,48	32,50	0,94
	5	0,64	1,47	33,06	0,94
	6	0,62	1,46	32,99	0,95

Для DCGAN спостерігається тенденція до зниження точності декодера зі збільшенням D, при цьому ємність корисного навантаження зростає, що вказує на здатність моделі приховувати більше даних, але з погіршенням якості декодування. Показники PSNR і SSIM залишаються відносно стабільними, демонструючи прийнятну візуальну якість, хоча з певними коливаннями.

WGAN показує схожий паттерн: точність декодера зменшується зі зростанням D, ємність

зростає, але PSNR і SSIM дещо нижчі порівняно з DCGAN за малих значень D, що свідчить про меншу візуальну якість на початкових рівнях навантаження. WGAN-GP демонструє кращі результати за PSNR і SSIM за всіх значень D, забезпечуючи вищу якість стегозображень і структурну схожість, хоча ємність корисного навантаження зростає повільніше порівняно з іншими моделями. Також, кожна з навчених архітектур показала найкращі результати за RS-BPP на рівні $D = 2$. Саме з таким рівнем D рис. 8 демонструє порівняння стегозображень згенеровані за допомогою WGAN-GP SteganoGAN з зображеннями-обкладинками.

Рис. 8. Оригінальні (зліва) і згенеровані за допомогою WGAN-GP SteganoGAN (справа) зображення при рівні $D = 2$ з датасету DIV2K

Порівняння моделей показує, що WGAN-GP незначно перевершує DCGAN і WGAN за показниками якості стегозображень, зберігаючи кращий баланс між ємністю та візуальною достовірністю, незважаючи на дещо нижчий приріст ємності. Точність декодера у всіх моделей знижується зі збільшенням D, але WGAN-GP забезпечує стабільніші результати завдяки градієнтному штрафу, який покращує адаптацію до складних розподілів даних. Це робить WGAN-GP оптимальним вибором для подальших досліджень, оскільки вона уникає проблем із вирізанням вагів, як у WGAN, і нестабільністю навчання, характерною для DCGAN із бінарною крос-ентропією, що дозволяє досягати вищої якості стегозображень при різних рівнях навантаження, як це показано на рис. 9, де оригінальне зображення порівнюється з стегозображеннями з різним рівнем D.

Слід зауважити, що незважаючи на приріст корисного навантаження, метрики якості зображень залишались на приблизно одному рівні, тоді як в статті [11]показники цих метрик хоч і були вище, але показували значний спад при рості D. Це може свідчити про більш стабільну роботи нейронних мереж за використання оригінальних архітектур DCGAN, WGAN, WGAN-GP в стеганографії.

Рис. 9. Порівняння оригінального зображення зліва із стегозображеннями з рівнями BPP від 1 до 4 зроблених за допомогою WGAN-GP SteganoGAN

Висновки

Дослідження присвячене розробці та аналізу трьох варіантів моделі SteganoGAN на основі архітектур DCGAN, WGAN і WGAN-GP із метою оцінки їхньої ефективності у стеганографії зображень. Проведений огляд предметної області показав, що стеганографія з використанням генеративно-змагальних мереж (GAN) відкриває нові можливості для приховування даних, поєднуючи високу ємність корисного навантаження з реалістичністю стегозображень. Еволюція архітектур GAN – від базових GAN до DCGAN із згортковими шарами, WGAN із метрикою Вассерштейна та WGAN-GP із градієнтним штрафом – дозволила вирішити ключові проблеми нестабільності навчання, низької якості зображень і обмеженої адаптивності, що зробило ці моделі придатними для стеганографічних задач. Оригінальна модель SteganoGAN, заснована на WGAN, була модифікована для трьох варіантів із різними архітектурами: DCGAN із бінарною крос-ентропією, WGAN із поверненням до багаторазового оновлення критика, і WGAN-GP із градієнтним штрафом. Для всіх моделей використано однакову функцію втрат кодера-декодера з акцентом на схожість стегозображень $\beta = 100,\ \gamma = 0.01$ та покращений критик із шарами MaxPooling і GlobalAveragePooling2D для якіснішої класифікації. Оцінка проводилася за метриками. RS-BPP, SSIM, PSNR і DecodingAccuracy, які показали, що WGAN-GP забезпечує найкращий баланс між якістю стегозображень і ємністю корисного навантаження, перевершуючи DCGAN і WGAN за показниками PSNR і SSIM, завдяки стабільності навчання, що досягається через градієнтний штраф. Таким чином, WGAN-GP є оптимальним вибором для стеганографії серед досліджених архітектур.

Подальше покращення метрик, таких як RS-BPP, SSIM, PSNR чи DecodingAccuracy, має обмежений сенс, оскільки WGAN-GP уже демонструє високі показники якості та стабільності, а додаткові покращення можуть призводити до надмірної складності моделі без значного приросту ефективності. Натомість доцільно зосередитися на розвитку стегоаналізу та підвищенні безпеки передачі даних. Одним із перспективних напрямів є інтеграція механізмів кодування пароля в зображення, коли приховане повідомлення кодується з використанням пароля, а декодування можливе лише за умови збігу паролів на стороні приймача, що додає додатковий рівень захисту. Зі сторони стегоаналізу передбачається розробити власний стегоаналізатор, адаптований до особливостей SteganoGAN, і імплементувати його в мережу як частину змагального процесу. Це дозволить моделі навчатися протистояти виявленню, підвищуючи її стійкість до атак, і водночас сприятиме створенню більш досконалих методів стегоаналізу, що є важливим для розвитку галузі. Крім того, для подальшого покращення нейромережі варто використовувати датасет зображень COCO, так як в роботі [11] навчена на ньому нейромережа SteganoGAN показувала значно кращі результати ніж датасету DIV2K, а використання таких структур для кодера-декодера як U-NET [16], [17] або ResNet50 [18], [19], [20] потенційно може дати кращі результати в якості та корисному навантаженні зображень.

СПИСОК ЛІТЕРАТУРИ

1. Holuband V., Fridrich J. Designing steganographic distortion using directional filters. *WIFS 2012 –Proceedings of the 2012 IEEE International Workshop on Information Forensics and Security*. 2012. P. 234–239. DOI: 10.1109/WIFS.2012.6412655.

2. Pevný T., Filler T., Bas P. Using High-Dimensional Image Models to Perform Highly Undetectable Steganography. *Information Hiding*. 2010. Vol. 6387 LNCS. P. 161–177. DOI: 10.1007/978-3-642-16435-4_13.

3. Holub V., Fridrich J., Denemark T. Universal distortion function for steganography in an arbitrary domain. *EURASIP J InfSecur*. 2014. Vol. 2014. DOI: 10.1186/1687-417X-2014-1.

4. Generative Adversarial Networks / I. J. Goodfellow et al. *SciRobot*. 2014. Vol. 3, № January. P. 2672–2680. URL: https://arxiv.org/abs/1406.2661v1 (date of access: 16.12.2024).

5. Generative Adversarial Networks for Synthetic Data Generation: A Comparative Study / C. Little et al. *Arxiv.org*. Dec. 2021. URL: https://arxiv.org/abs/2112.01925v1 (date of access: 19.03.2025).

6. O'Shea K., Nash R. An Introduction to Convolutional Neural Networks. *Int J Res Appl Sci Eng Technol*. Nov. 2015. Vol. 10, № 12. P. 943–947. DOI: 10.22214/ijraset.2022.47789.

7. Arjovsky M., Chintala S., Bottou L. Wasserstein GAN. Jan. 2017.

URL: https://arxiv.org/abs/1701.07875v3 (date of access: 04.12.2024).

8. Read-through: Wasserstein GAN. URL: https://www.alexirpan.com/2017/02/22/wasserstein-gan.html# (date of access: 21.12.2024).

9. Openai L. W. From GAN to WGAN. Apr. 2019. URL: https://arxiv.org/abs/1904.08994v1 (date of access: 21.12.2024).

10. Improved Training of Wasserstein GANs / I. Gulrajani et al. *Adv Neural Inf Process Syst.* Mar. 2017. Vol. 2017-December. P. 5768–5778. URL: https://arxiv.org/abs/1704.00028v3 (date of access: 04.12.2024).

11. SteganoGAN: High Capacity Image Steganography with GANs / K. A. Zhang et al. Jan. 2019. DOI: 10.48550/arXiv.1901.03892.

12. Baluja S. Hiding Images in Plain Sight: Deep Steganography. *Neural Information Processing Systems*. 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/838e8afb1ca34354ac209f53d90c3a43-Paper.pdf (date of access: 05.11.2024).

13. Deep residual learning for image recognition / K. He et al. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Dec. 2016. Vol. 2016-December. P. 770–778. DOI: 10.1109/CVPR.2016.90.

14. Densely connected convolutional networks / G. Huang et al. *Proceedings - 30th IEEE Conferenceon Computer Visionand Pattern Recognition, CVPR 2017*. Nov. 2017. Vol. 2017-January. P. 2261–2269. DOI: 10.1109/CVPR.2017.243.

15. Agustssonand E., Timofte R. NTIRE 2017 ChallengeonSingleImageSuper-Resolution: Dataset and Study. *IEEE Computer Society Conferenceon Computer Vision and Pattern Recognition Workshops*. Aug. 2017. Vol. 2017-July. P. 1122–1131. DOI: 10.1109/CVPRW.2017.150.

16. Wengand W., Zhu X. U-Net: Convolutional Networks for Biomedical Image Segmentation. *IEEE Access*. May 2015. Vol. 9. P. 16591–16603. DOI: 10.1109/ACCESS.2021.3053408.

17. Reversible image steganography scheme based on a U-net structure / X. Duan et al. Jan. 2019. Volume 7. P. 9314–9323. DOI: 10.1109/ACCESS.2019.2891247.

18. Hybrid Deep Learning Model Based on GAN and RESNET for Detecting Fake Faces / S. Safwat et al. *IEEE Access.* 2024. Vol. 12. P. 86391–86402. DOI: 10.1109/ACCESS.2024.3416910.

19. Identity Mappings in Deep Residua lNetworks / K. He et al. Lecture Notesin Computer Science (including subseries Lecture Notesin Artificial Intelligence and Lecture Notesin Bioinformatics). Mar. 2016. Vol. 9908 LNCS. P. 630–645. DOI: 10.1007/978-3-319-46493-0_38.

20. Ledigetal C. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. Proceedings - 30th IEEE Conferenceon Computer Vision and Pattern Recognition, CVPR 2017. Sep. 2016. Vol. 2017-January. P. 105–114. DOI: 10.1109/CVPR.2017.19.

Стаття надійшла до редакції 02.06.2025.
Стаття пройшла рецензування 28.06.2025.

Хома Дмитро Юрійович – аспірант кафедри прикладної математики та інформатики, e-mail: dmytro.khoma@donntu.edu.ua.
Донецький національний технічний університет.