

СТВОРЕННЯ ТА ОПТИМІЗАЦІЯ ЛЕГКИХ ГЕНЕРАТИВНИХ ЗМАГАЛЬНИХ НЕЙРОМЕРЕЖ ДЛЯ ПОКРАЩЕННЯ ЯКОСТІ ВІДЕО НА КЛІЄНТСЬКИХ ПРИСТРОЯХ З ВИКОРИСТАННЯМ WEBGPU

М. Р. Максимів, Т. Є. Рак

Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин
E-mail: mykola.r.maksymiv@lpnu.ua, taras.y.rak@lpnu.ua

© Максимів М. Р., Рак Т. Є., 2025

Розглянуто проблеми для задач покращення якості цифрових відеозображень для хмарних середовищ, а також на клієнтській стороні за допомогою генеративно-змагальних нейромереж (GAN), адаптованих для роботи у браузері. Запропоновано метод, що використовує WebGPU для прискореного виконання згорткових обчислень, що дає змогу збільшувати роздільну здатність та покращувати якість низькоякісного відео в реальному часі без значного навантаження на сервери. Для оптимізації нейромережі додатково було використано методи Pruning та Knowledge Distillation, що дало змогу зменшити розмір моделі на 40–60 % без значної втрати якості.

Результати експериментів показали, що реалізація запропонованого методу підвищує продуктивність обробки відео у браузері у 2–4 рази порівняно з іншими неоптимізованими моделями. Оцінка якості відео показала покращення PSNR та підвищення SSIM порівняно з традиційними методами збільшення роздільної здатності. Запропонований підхід може бути інтегрований у потокові сервіси та вебзастосунки, що дасть змогу знизити навантаження на комп’ютерні мережі та забезпечити кращий користувацький досвід з меншими витратами на хмарні та серверні обчислення.

Ключові слова: відео високої чіткості, генеративні змагальні мережі, зображення високої роздільної здатності, моделі оптимізації, надроздільна здатність зображення, штучні нейронні мережі.

Вступ

Якість відео є одним із ключових параметрів, що визначають сприйняття мультимедійного контенту користувачами. Зростання попиту на відеосервіси, такі як потокове мовлення, відеоконференції та мобільний стрімінг, потребує удосконалення методів обробки зображень та відео для забезпечення високої якості відтворення в умовах обмеженої пропускної здатності мережі [1].

Сучасні методи покращення якості відео поділяються на класичні алгоритми масштабування (наприклад, бікубічна інтерполяція) та методи, засновані на глибокому навчанні, зокрема на GAN мережах [2]. Хоча GAN-архітектури, такі як ESRGAN [3], забезпечують високу якість реконструкції відео, їх обчислювальна складність створює значні виклики для інтеграції в реальному часі на стороні клієнта. Більшість сучасних систем відеообробки реалізовані у хмарному середовищі, що підвищує затримку, збільшує навантаження на сервери та потребує значних фінансових витрат на інфраструктуру [4].

Значним викликом є оптимізація нейронних мереж для роботи у середовищах з обмеженими ресурсами, таких як веббраузери та мобільні пристрої. Останнім часом розвиток технологій WebGPU [5] та TensorFlow.js [6] відкрив можливості для прискореного виконання нейронних мереж у браузері, що дає змогу частково перенести обчислення на сторону клієнта. Це може знизити затримку, оптимізувати використання серверних ресурсів та покращити користувацький досвід без необхідності розгортання додаткової хмарної інфраструктури.

У цій праці пропонується підхід до покращення якості відео на стороні клієнта шляхом використання оптимізованої GAN-мережі для масштабування відео в режимі реального часу. Метод передбачає використання WebGPU [5] для прискорення виконання згорткових обчислень та застосування технік оптимізації моделі.

Огляд літературних джерел

Покращення якості відео є важливим напрямом досліджень у сфері цифрової обробки зображень, особливо у контексті потокового передавання мультимедійного контенту, відеоконференцій та мобільних додатків. Традиційні методи масштабування, такі як бікубічна та білінійна інтерполяція, широко застосовуються, проте вони мають суттєві обмеження у відновленні дрібних деталей та текстур [1, 9].

За останні роки активно почали розвиватись глибокі нейронні мережі, які значно перевершують класичні алгоритми масштабування. Однією з перших успішних моделей стала Super-Resolution Convolutional Neural Network (SRCNN), яка демонструє високу ефективність у збільшенні роздільної здатності [2]. Подальший розвиток цієї технології призвів до появи архітектури GAN, які використовують змагальне навчання генератора та дискримінатора (рис. 1) для отримання реалістичних зображень.

Генератор G у GAN перетворює низькоякісне відео у відновлене високоякісне, генеруючи деталі, які не були у вихідних кадрах. Дискримінатор оцінює якість згенерованих кадрів, порівнюючи їх із реальними відео для покращення генерації текстур і деталей.

Процес навчання GAN побудований на концепції змагального навчання, де генератор намагається створювати реалістичні кадри, які дискримінатор не зможе відрізнити від справжніх. Дискримінатор натомість постійно вдосконалюється, покращуючи свої здатності розрізняти справжні та згенеровані зображення. У працях [3, 4] показано, що ESRGAN (Enhanced Super-Resolution GAN) перевершує попередні методи завдяки покращеній текстурній реконструкції та зменшення артефактів.

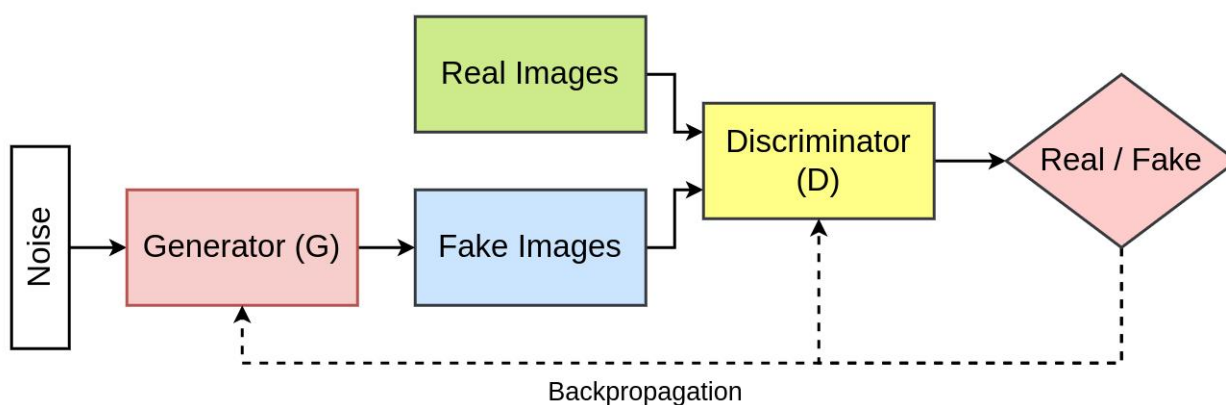


Рис. 1. Загальна схема роботи методів покращення якості цифрових зображень на основі GAN архітектури

Однак також варто зазначити обмеження традиційних GAN для відео, зокрема:

1. Високі обчислювальні витрати – традиційні GAN потребують значних ресурсів для навчання та виконання.
2. Стабільність навчання – тренування GAN може бути нестабільним, що призводить до артефактів або “зависання” навчання.
3. Міжкадрова узгодженість – більшість GAN моделей працюють на окремих кадрах, що може спричинити розбіжності між послідовними кадрами.

Також варто зазначити факт, що більшість досліджень у цій сфері зосереджені на серверних рішеннях, які потребують значних обчислювальних ресурсів. Наприклад, у [4] аналізуються підходи до використання GAN у хмарних обчисленнях для підвищення якості потокового відео. Хоча такі методи забезпечують високу точність реконструкції, вони створюють додаткове навантаження на серверну інфраструктуру та збільшують затримки передавання.

Розвиток WebGPU [5] відкрив можливості для перенесення обчислень безпосередньо на клієнтські пристрої, що дає змогу виконувати складні тензорні обчислення у браузері з продуктивністю, наближеною до настільних GPU. У публікаціях [6, 7] продемонстровано, що WebGPU може значно прискорити виконання глибоких нейромереж у вебсередовищі. Окрім того, такі бібліотеки як TensorFlow.js [6] та ONNX.js [8] надають інструменти для інтеграції попередньо навчених моделей у браузер, що відкриває можливості для використання GAN безпосередньо на клієнтській стороні.

Актуальність використання інших архітектур для покращення відео в клієнтських середовищах також розглядається у [9], де аналізуються методи відновлення деталей після стиснення відео. У праці показано, що складні для обчислення методи, такі як BasicVSR так BasicVSR++, можуть забезпечувати значне підвищення якості відео завдяки генерації додаткових пікселів і проміжних кадрів, які традиційні методи не можуть відновити.

Однією з основних проблем використання нейромереж у браузері є їхня висока обчислювальна складність. Вирішенням цієї проблеми можуть стати методи оптимізації нейромереж, такі як Pruning та Knowledge Distillation, які дають змогу значно зменшити обсяг моделей без значної втрати якості [10, 11, 12]. Зокрема, у [12] досліджено методи стиснення нейромереж для ефективного виконання глибокого навчання в реальному часі.

Отже, сучасні дослідження підтверджують ефективність GAN для покращення якості відео, проте їхнє застосування у браузері залишається недостатньо вивченим. Запропоновано адаптований підхід до реалізації Super-Resolution GAN, оптимізованої для WebGPU, що дає змогу поєднати високу продуктивність із ефективною реконструкцією відео в реальному часі без необхідності використання серверних ресурсів.

Постановка завдання

Метою цього дослідження є розробка та оптимізація GAN-архітектури для покращення якості відео в браузері із використанням WebGPU. Основні цілі:

1. Забезпечення високої швидкості обробки відео в реальному часі.
2. Зменшення серверного навантаження завдяки клієнтській обробці.
3. Оптимізація нейромережі для ефективної роботи в браузері.
4. Відновлення втрачених деталей відео після стиснення кодеками (H.264, AV1, VP9).

Основним завданням цього дослідження є:

1. Розробка оптимізованої GAN-моделі для клієнтських пристроїв.
2. Використання WebGPU для прискорення згорткових обчислень.
3. Дослідження ефективності компактних нейромереж у браузері.
4. Оцінка продуктивності методу (FPS, PSNR, SSIM) порівняно з традиційними методами.

Запропонований підхід дає змогу зменшити залежність від серверних обчислень, забезпечуючи покращення відео без значних затримок.

Реалізація запропонованого методу

Запропонований метод покращення якості відео в браузері ґрунтується на використанні Super-Resolution GAN (SRGAN), оптимізованої для роботи у клієнтському середовищі, тобто у WEB-браузері за допомогою WebGPU. Основною ідеєю є виконання реконструкції відеокадрів у браузері, що дає змогу зменшити залежність від серверних ресурсів та підвищити продуктивність відтворення.

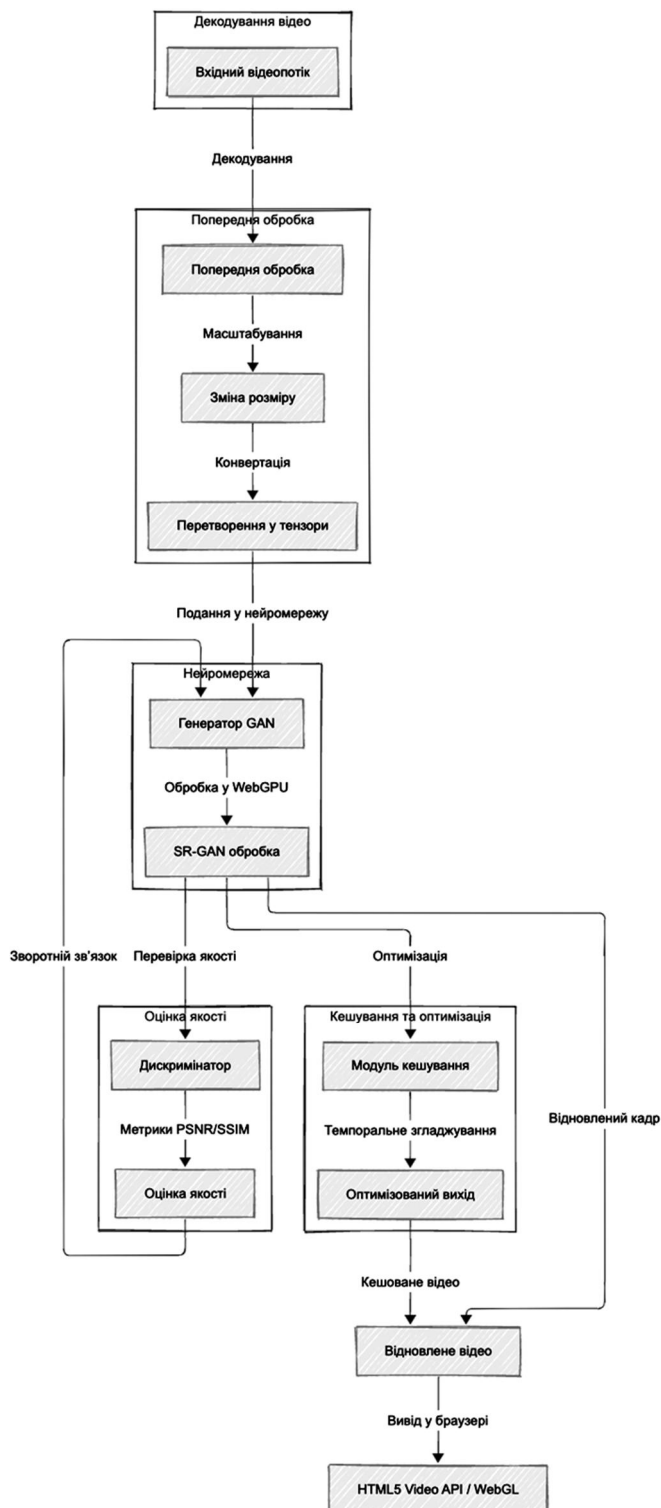


Рис. 2. Схема роботи запропонованого методу покращення якості відео у веббраузері

Нейромережа складається з двох основних компонентів: генератора та дискримінатора. Вхідне зображення проходить велику кількість етапів (рис. 2), перед тим як вивести результуюче відео.

Генератор (Generator G)

Генератор є основним компонентом системи, що виконує реконструкцію відео від низької (LR) до високої (HR). Архітектура генератора базується на блоках ResNet, які зменшують розмиття та артефакти, які з'являються під час традиційного масштабування.

Формально генератор G можна записати так:

$$I_{HR} = G(I_{LR}, \theta_G), \quad (1)$$

де I_{LR} - вхідне зображення низької якості; I_{HR} - вихідне покращене зображення; θ_G - параметри генератора.

Для реконструкції використовується функція втрат, що комбінує перцепційні втрати (VGG-based), L_1 -різницю та змагальні втрати:

$$L_G = \lambda_1 \mathcal{L}_{L1} + \lambda_2 \mathcal{L}_{\text{perc}} + \lambda_3 \mathcal{L}_{\mathcal{GAN}}, \quad (2)$$

де $\lambda_1 \lambda_2 \lambda_3$ - коефіцієнти до кожної функції втрати, які корегуються впродовж навчання;

L_1 - абсолютна помилка між реальним та згенерованим кадрами:

$$\mathcal{L}_{L1} = \frac{1}{N} \sum_{i=1}^N |G(I_{LR}^{(i)}) - I_{HR}^{(i)}|, \quad (3)$$

$\mathcal{L}_{\text{perc}}$ - перцепційні втрати на основі VGG-19, що порівнюють високорівневі особливості:

$$\mathcal{L}_{\text{perc}} = \frac{1}{N} \sum_{i=1}^N |\Phi(G(I_{LR}^{(i)})) - \Phi(I_{HR}^{(i)})|^2, \quad (4)$$

де Φ - це VGG-19 feature extractor.

$\mathcal{L}_{\mathcal{GAN}}$ змагальні втрати за стандартною функцією GAN:

$$\mathcal{L}_{\mathcal{GAN}} = E_{I_{HR}}[\log D(I_{HR})] + E_{I_{LR}}[\log(1 - D(I_{LR}))], \quad (5)$$

де D є дискримінатором.

Дискримінатор (Discriminator D)

Дискримінатор використовує багаторівневу згорткову нейромережу для оцінки ймовірності того, що зображення є справжнім або згенерованим:

$$D(I) = \sigma(f(I, \theta_D)), \quad (6)$$

де σ є функцією сигмоїди, що визначає ймовірність реального зображення, а $(f(I, \theta_D))$ є виходом дискримінатора.

Загальна функція втрати дискримінатора:

$$\mathcal{L}_D = E_{I_{HR}}[\log D(I_{HR})] + E_{I_{LR}}[\log(1 - D(G(I_{LR})))] \quad (7)$$

Ця функція втрат є стандартною для GAN, де перша частина $E_{I_{HR}}[\log D(I_{HR})]$ максимізує ймовірність того, що справжні кадри класифіковані правильно. Друга частина $E_{I_{LR}}[\log(1 - D(G(I_{LR})))]$ зменшує ймовірність того, що дискримінатор "повірить" у підробку, що визначає ймовірність реального зображення.

Оптимізація моделі для клієнтських пристроїв

Оскільки WebGL має обмежену підтримку тензорних обчислень, було прийнято рішення використовувати WebGPU [5], що дає змогу прискорити виконання нейромережі у браузері.

Щоб зменшити вимоги до обчислювальних ресурсів у браузері:

- 1) зменшено кількість ResNet-блоків (від 16 до 8);
- 2) використано групові згортки (Grouped Convolutions) для зменшення кількості вхідних параметрів мережі;

3) використано Winograd Convolutions [13] для того, щоб пришвидшити обчислення нейронних згорток.

Отримана модель конвертується у ONNX-формат для подальшого використання у браузері через WebGPU, що дає змогу досягти реального часу обробки відео на клієнтських пристроях.

Для зменшення розміру моделі використані:

1. Pruning – видалення нейронів, що мають низький внесок у прийняття рішень [12].
2. Knowledge Distillation – навчання меншої моделі на основі передбачень великої моделі [11].

Це дає змогу зменшити використання пам'яті на 40- 60 % без значних втрат якості.

Результати дослідження

Для оцінки ефективності запропонованого методу було проведено серію експериментів, які містили тестування продуктивності запропонованого методу, оцінку якості покращеного відео та аналіз використання апаратних ресурсів у браузері.

Було використано відеофрагменти з відкритих датасетів: Vimeo-90K (HD та 4K відео), YouTube-8M (відео з високим стисненням), Vid4 (стандартний набір для тестування Super-Resolution).

Кожне відео було попередньо закодоване у три різні формати:

- 1) H.264 (бітрейт 2-5 Mbps) – стандарт для потокових сервісів;
- 2) VP9 (бітрейт 1-3 Mbps) – використовується в YouTube;
- 3) AV1 (бітрейт 0.5-2 Mbps) – новий стандарт, знижує пропускну здатність.

Відео тестувалися у трьох роздільних здатностях: SD (480p → 1080p), HD (720p → 1080p), Full HD (1080p → 1440p).

Загальні параметри навчання мережі:

- 1) Оптимізатор: Adam;
- 2) Коефіцієнти моменту: $\beta_1 = 0.9$, $\beta_2 = 0.999$;
- 3) Розмір міні-батчу: 16;
- 4) Розмір вхідного кадру: 64×64 ;
- 5) Функція втрат: комбінована втрата відповідно до формули 2 ($\mathcal{L}_{L1}$ -різниця + перцепційна + змагальна втрати).

Коефіцієнти ваг у загальній функції втрати: $\lambda_1 = 1.0$, $\lambda_2 = 0.01$, $\lambda_3 = 0.005$. На початку навчання основний внесок у втрати забезпечує L_1 -різниця для стабільного оновлення ваг. Після 50 000 ітерацій ваги перцепційних втрат поступово збільшуються для кращого відновлення текстур. Навчання тривало до досягнення PSNR > 33.5 дБ на тестовому наборі.

Оцінка продуктивності проводилася за допомогою таких метрик:

- 1) FPS (кадри в секунду) – швидкість обробки.
- 2) PSNR (Peak Signal-to-Noise Ratio) – рівень шуму у відновленому відео:

$$PSNR = 10 \left(\frac{MAX^2}{MSE} \right), \quad (8)$$

де MSE - середня квадратична помилка між створеним і оригінальним зображеннями.

- 3) SSIM (Structural Similarity Index Measure) – подібність між відновленим і реальним відео.

$$SSIM(I, Y) = \frac{2u_I u_Y + k_1}{u_I^2 + u_Y^2 + k_1} * \frac{2\sigma_{IY} + k_2}{\sigma_I^2 + \sigma_Y^2 + k_2}, \quad (9)$$

де $u_I u_Y$ - середні інтенсивності двох порівнюваних зображень; σ_I^2, σ_Y^2 - дисперсія даних зображень і σ_{IY} їхня коваріація.

Таблиця 1

Результат експерименту вимірювання FPS після використання запропонованого методу

Роздільна здатність	WebGL (FPS)	WebGPU (FPS)	Прискорення
480p → 720p	35 FPS	97 FPS	2.8x
720p → 1080p	12 FPS	35 FPS	2.9x
1080p → 1440p	5 FPS	18 FPS	3.6x

З результатів (табл. 1) видно, що WebGPU забезпечує 2-3.6x прискорення порівняно з WebGL, що дає змогу відтворювати Full HD відео у реальному часі. Проте варто зазначити, що на слабких пристроях WebGL не дозволяє досягти навіть 10 FPS, тоді як WebGPU покращує цей показник.

Порівняно зі стандартними статичними методами збільшення роздільної здатності (бікубічна інтерполяція), GAN-модель дає значно кращий PSNR/SSIM, що означає менший рівень шуму у відео.

Таблиця 2

Результат замірів структурних метрик PSNR та SSIM в збільшенні роздільної здатності відео

Метод	PSNR (720p → 1080p)	PSNR (1080p → 1440p)	SSIM (720p → 1080p)	SSIM (1080p → 1440p)
Бікубічна інтерполяція	27.3 дБ	25.5 дБ	0.78	0.70
ESRGAN (WebGL)	33.2 дБ	31.1 дБ	0.85	0.83
Запропонований метод	34.5 дБ	32.8 дБ	0.87	0.89

З даних табл. 2 випливає, що використання GAN значно покращує візуальну якість відео порівняно з традиційними методами. WebGPU-оптимізований GAN на 7-9 % кращий за WebGL-версію у браузерних задачах.

Всі заміри і отримані результати з експериментів проводились на 3 різних пристроях:

1. Ноутбук (Intel Core i7, RTX 3060, 16GB RAM) – висока продуктивність.
2. Смартфон (Snapdragon 888, Adreno 660, 8GB RAM) – середній рівень продуктивності.
3. Старий ноутбук (Intel Core i5, UHD Graphics, 8GB RAM) – слабкий пристрій.

Таблиця 3

Статистика використання апаратних ресурсів у браузері з використанням WebGPU та WebGL

Пристрій	WebGL (GPU %)	WebGPU (GPU %)	WebGL (CPU %)	WebGPU (CPU %)	Температура (°C)	Стійкість до навантаження
Ноутбук (RTX 3060)	45 %	78 %	35 %	12 %	65 °C (стабільна)	Висока, без перегріву
Смартфон (Adreno 660)	58 %	85 %	42 %	18 %	45 °C (порівняно стабільна, але помітне помірне нагрівання корпусу)	Середня, можливе зниження продуктивності під час довготривалого використання
Старий ноутбук (UHD Graphics)	75 %	92 %	55 %	22 %	74 °C (помітний перегрів та можливий тротлінг)	Низька, значне падіння продуктивності через нагрів

За результатами табл. 3 можна стверджувати, що WebGPU ефективніше використовує GPU, що дає змогу зменшити навантаження на CPU, яке своєю чергою зазвичай більше завантажено у браузерних задачах. На слабких пристроях WebGPU дозволяє значно зменшити використання процесора, що позитивно впливає на енергоспоживання.

З точки зору температури нагріву чипа і навантаження на систему можна зазначити, що ноутбуки з дискретною графікою (RTX 3060) показують стабільну продуктивність, температура залишається в безпечних межах. Мобільні пристрої (Adreno 660) демонструють хорошу продуктивність, але тривале використання може призвести до нагрівання корпусу, що може спричинити термальне обмеження продуктивності (throttling). Старі ноутбуки (UHD Graphics) мають високі температури (~74°C), що може спричинити автоматичне зниження продуктивності вже через 15 хвилин роботи.

Висновки

Досліджено проблему покращення якості відео на стороні клієнта за допомогою генеративних змагальних нейронних мереж та реалізації їх роботи в браузері за допомогою інтерфейсу WebGPU. Запропонований підхід дає змогу частково перенести математичні обчислення, які зазвичай виконується на сервері, на клієнтські пристрої, що допомагає знизити навантаження на інфраструктуру потокових сервісів і покращити якість відео без необхідності збільшення бітрейта.

Результати експериментальної перевірки підтвердили ефективність цього рішення. Впровадження WebGPU дало змогу збільшити швидкість обробки відео в браузері в 2-4 рази порівняно з інтерфейсом WebGL, що забезпечило стабільну роботу в режимі реального часу для відео у FullHD якості. Аналіз якості вихідного відео показав збільшення статистичних метрик PSNR на 4–7 дБ та SSIM на 7–9 % порівняно з традиційними методами збільшення роздільної здатності. Крім того, оптимізація нейронної мережі за допомогою Knowledge Distillation та Pruning дозволила зменшити обсяг використовуваної пам'яті на 40–60 %, що дало можливість використовувати її навіть на низькопродуктивних пристроях, на зразок мобільних телефонів. Проте слабкі ноутбуки довго не витримують високого навантаження, що потрібно враховувати під час розгортання технології у клієнтських застосунках.

Отримані результати підтверджують, що використання GAN для покращення відео на стороні клієнта є перспективним напрямом, який можна застосувати в потокових сервісах, відеоконференціях і браузерних мультимедійних платформах. Планується подальше дослідження для адаптації нейронної мережі для більш ефективного обліку руху між кадрами та вивчення впливу енергоспоживання на мобільні пристрої під час використання WebGPU для обробки відео в реальному часі.

Список літератури

1. Madarasingha C., Muramudalige S., Jourjon G. VideoTrain++: GAN-based Adaptive Framework for Synthetic Video Traffic Generation. *Computer Networks*. 2022. DOI: 10.1016/j.comnet.2022.108785.
2. Lee R., Venieris S. I., Lane N. D. Deep neural network-based enhancement for image and video streaming systems: A survey and future directions. *ACM Computing Surveys*. 2021. DOI: 10.1145/3469094.
3. Liborio J. M., Melo C., Silva M. Internet Video Delivery Improved by Super-Resolution with GAN. *MDPI Future Internet*. 2022. DOI: 10.3390/fi14120364.
4. Shi W., Li Q., Yu Q., Wang F., Shen G. A Survey on Intelligent Solutions for Increased Video Delivery Quality in Cloud-Edge-End Networks. *IEEE Surveys & Tutorials*. 2024. DOI: 10.1109/COMST.2024.3427360.
5. WebGPU Working Group, "WebGPU Specification", W3C Technical Report, 2023. URL: <https://www.w3.org/TR/WGSL/>.
6. TensorFlow.js Team, "TensorFlow.js: Machine Learning for the Web and Beyond". *Proceedings of Machine Learning and Systems*, 2019. DOI: <https://doi.org/10.48550/arXiv.1901.05350>.
7. He Z., Yang Y., Qiu L., Park K. Real-Time Neural Video Recovery and Enhancement on Mobile Devices. DOI: 10.48550/arXiv.2307.12152.
8. Microsoft ONNX Team. URL: <https://github.com/microsoft/onnxjs>.

9. Maksymiv M., Rak T. *Method of Video Quality-Improving. Artificial Intelligence*. 2023. 28(3):47- 62. DOI: 10.15407/jai2023.03.047.
10. Hinton G., Vinyals O., Dean J. *Distilling the Knowledge in a Neural Network* (arXiv preprint arXiv:1503.02531), 2015. DOI: 10.48550/arXiv.1503.02531.
11. Mishra R., Gupta H. P. *Designing and training of lightweight neural networks on edge devices using early halting in knowledge distillation. IEEE Transactions on Mobile Computing*. 2023. DOI: 10.1109/TMC.2023.10187678.
12. Zhang L., Bao C., Ma K. *Self-distillation: Towards efficient and compact neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021. DOI: 10.1109/TPAMI.2021.9381661.
13. Alam S. A., Anderson A., Barabasz B., Gregg D. *Winograd Convolution for Deep Neural Networks: Efficient Point Selection. ACM Transactions on Embedded Computing Systems*. 2022. DOI:10.48550/arXiv.2201.10369.

BUILDING AND OPTIMIZING LIGHTWEIGHT GENERATIVE ADVERSARIAL NEURAL NETWORKS TO ENHANCE VIDEO QUALITY IN THE CLIENT DEVICES USING WEBGPU

M. R. Maksymiv, T. Y. Rak

Lviv Polytechnic National University,
Department of Electronic Computational Machines

© Maksymiv M. R., Rak T. Y., 2025

The paper considers problems for the tasks of improving the quality of digital video images for cloud environments, as well as on the client side using generative adversarial neural networks (GANs) adapted for work in the browser. A method is proposed that uses WebGPU for accelerated execution of convolutional calculations, which allows to increase the resolution and improve the quality of low-quality video in real time without significant load on servers. Optimization of the neural network includes the use of Pruning and Knowledge Distillation methods, which made it possible to reduce the size of the model by 40–60 % without significant loss of quality.

The results of the experiments showed that the implementation of the proposed method increases the performance of video processing in the browser by 2–4 times compared to models based on the WebGL interface. The video quality assessment showed an improvement in PSNR and an increase in SSIM compared to traditional methods of increasing resolution. The proposed approach can be integrated into streaming services and web applications, which will reduce the load on computer networks and provide a better user experience with lower costs for cloud and server computing.

Keywords: Generative adversarial networks, High definition video, High-resolution imaging, Image Superresolution, Optimization models, Neural network.