

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему: «Дослідження технологій стиснення відео для сервісів
відео-хостингу»

Виконав: студент групи Іт-61

Спеціальності 126 «Інформаційні системи та
технології»

(шифр і назва)

Драгула Роман Васильович

(Прізвище та ініціали)

Керівник: к.е.н., доцент Шувар Б.І.

(Прізвище та ініціали)

Рецензент: к.т.н., доцент Сиротюк С.В.

(Прізвище та ініціали)

ДУБЛЯНИ-2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____

д.т.н., проф. А.М. Тригуба

«____» _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Драгулі Роману Васильовичу

1. Тема роботи: «Дослідження технологій стиснення відео для сервісів відео-хостингу»

Керівник роботи Шувар Богдан Іванович, доцент
затверджені наказом по університету від 12.09.2024 року № 616/к-с.

2. Строк подання студентом роботи 10.12.2024 р.

3. Вихідні дані до роботи: вимоги до узгодження змісту проектів збирання врожаю; методика проектування інформаційно-аналітичних систем.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити) _____

Вступ.

1. Аналіз існуючого стану стиснення відео та чинних стандартів.

2. Основи теорії стиснення відео з використанням нейронних мереж та аналіз даних для навчання моделей.

3. Результати дослідження технологій стиснення відео для сервісів відео-хостингу.

4. Охорона праці та безпека у надзвичайних ситуаціях.

5. Визначення економічної ефективності від використання моделі стиснення відео для сервісів відео-хостингу.

Висновки та пропозиції.

Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових слайдів): . аналіз існуючого стану стиснення відео та чинних стандартів; основи теорії стиснення відео звикористанням нейронних мереж та аналіз даних для навчання моделей; результати дослідження технологій стиснення відео для сервісів відео-хостингу; визначення економічної ефективності.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 5	<i>Шувар Б.І., доцент кафедри інформаційних технологій</i>		
4	<i>Городецький І.М., доцент кафедри фізики, інженерної графіки та безпеки виробництва</i>		

7. Дата видачі завдання

12 вересня 2024 р.

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	Примітка
1	<i>Написання першого розділу</i>	<i>12.09-20.09.24</i>	
2	<i>Виконання другого розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>21.09-14.10.24</i>	
3.	<i>Виконання третього розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>15.10-10.11.24</i>	
4.	<i>Написання розділу «Охорона праці та безпека у надзвичайних ситуаціях»</i>	<i>11.11-20.11.24</i>	
5.	<i>Оцінення ефективності запропонованої системи</i>	<i>21.11-30.30.24</i>	
6.	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу</i>	<i>01-04.12.24</i>	
7.	<i>Завершення роботи в цілому</i>	<i>05-10.12.24</i>	

Студент _____ Драгула Р.В.
(підпис)

Керівник роботи _____ Шувар Б.І.
(підпис)

УДК 004.932.4:004.738.5

Дослідження технологій стиснення відео для сервісів відео-хостингу.

Драгула Р.В. Кафедра інформаційних технологій – Дубляни, ЛНУП, 2024.

Кваліфікаційна робота: 81 с. текст. част., 23 рис., 7 табл., 15 арк. ілюстраційного матеріалу, 51 джерел0.

У роботі проведено аналіз існуючого стану технологій стиснення відео та їх застосування для сервісів відео-хостингу. Особлива увага приділена методам H.264/AVC та H.265/HEVC, які на сьогодні є основою для обробки та передачі відеоданих.

У другому розділі розглянуто основи теорії стиснення відео із застосуванням нейронних мереж. Проведено аналіз можливостей використання глибокого навчання, зокрема згорткових нейронних мереж (CNN) та генеративних змагальних мереж (GANs), для підвищення якості обробки відео.

Третій розділ присвячено розробці модулів кодування відео на основі CNN. Представлено чотири модулі: постобробка (PP), фільтрація в циклі (ILF), адаптація просторової роздільної здатності (SRA) та адаптація ефективної глибини бітів (EBDA). Проведено порівняння архітектур нейронних мереж для оцінки їх ефективності під час стиснення відео.

Четвертий розділ стосується питань охорони праці та безпеки у процесі стиснення відео для відео-хостингів. Проведено аналіз можливих небезпек, розроблено заходи безпеки та створено логічно-імітаційну модель, яка дозволяє оцінити ризики виникнення травм під час стиснення відео.

У п'ятому розділі оцінено економічну ефективність використання розроблених моделей стиснення відео для відео-хостингів. Визначено потенційне зниження витрат на зберігання та передачу даних, а також підвищення якості відеоконтенту.

Ключові слова: стиснення відео, нейронні мережі, відео-хостинг, H.264/AVC, H.265/HEVC, генеративні змагальні мережі, ефективність.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧОГО СТАНУ СТИСНЕННЯ ВІДЕО ТА ЧИННИХ СТАНДАРТІВ.....	9
1.1. Аналіз існуючого стану стиснення відео для сервісів відео-хостингу.....	9
1.2. Аналіз існуючих методів стиснення відео.....	11
1.3. Характеристики систем H.264/AVC ТА H.265/HEVC	16
1.4. Завдання кваліфікаційної роботи	19
РОЗДІЛ 2. ОСНОВИ ТЕОРІЇ СТИСНЕННЯ ВІДЕО З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ ТА АНАЛІЗ ДАНИХ ДЛЯ НАВЧАННЯ МОДЕЛЕЙ.....	21
2.1. Особливості відеокодування та використання глибокого навчання для обробки зображень	21
2.2. Основи теорії стиснення відео з використанням нейронних мереж.....	25
2.3. Згортова нейронна мережа (CNN) для стиснення відео.....	27
2.4. Генеративні змагальні мережі (GANS).....	29
2.5. Вибір та аналіз навчальних та тестових баз даних	32
РОЗДІЛ 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ СТИСНЕННЯ ВІДЕО ДЛЯ СЕРВІСІВ ВІДЕО-ХОСТИНГУ	35
3.1. Розробка модулів кодування на основі cnn	35
3.1.1. Модуль кодування 1 (post processing - PP)	35
3.1.2. Модуль кодування 2 (фільтрація в циклі - ILF).....	37
3.1.3. Модуль кодування 3 (адаптація просторової роздільності - SRA).....	39
3.1.4. Модуль кодування 4 (адаптація ефективної глибини бітів - EBDA).....	42
3.2. Опис бази даних	44
3.3. Порівняння архітектур нейронних мереж під час стиснення відео	46
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ	53

4.1. Аналіз небезпек та шкідливих виробничих під час стиснення відео для сервісів відео-хостингу	53
4.2. Заходи безпеки під час стиснення відео для сервісів відео-хостингу	55
4.3. Розробка логічно-імітаційної моделі процесу виникнення травм під час стиснення та монтажу відео	56
4.4. Розробка заходів щодо безпеки у надзвичайних ситуаціях.....	57
РОЗДІЛ 5. ВИЗНАЧЕННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД ВИКОРИСТАННЯ МОДЕЛІ СТИСНЕННЯ ВІДЕО ДЛЯ СЕРВІСІВ ВІДЕО- ХОСТИНГУ	60
ВИСНОВКИ І ПРОПОЗИЦІЇ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	67
ДОДАТКИ.....	73
Додаток А. Фрагмент коду для coding module 1	74
Додаток Б. Фрагмент коду для coding module 2.....	76
Додаток В. Фрагмент коду для coding module 3	78
Додаток Г. Фрагмент коду для coding module 4.....	80

ВСТУП

На даний час спостерігається стрімке зростання популярності відео-контенту, розповсюдженого через платформи відео-хостингу, такі як YouTube, Vimeo, TikTok тощо, що створює значний попит на ефективні методи стиснення відео. Це пов'язано зі зниженням обсягів передачі даних, забезпеченням високої якості зображення та зменшенням витрат на зберігання даних [11]. Технології стиснення відео змінюють ключову роль зі збереженням стабільної роботи відео-сервісів, оптимізацією мережевого трафіку та підвищеною задоволеністю користувачів. Враховуючи збільшення обсягів відео у форматах високої чіткості (HD, 4K, 8K).

Ефективні технології стиснення відео є важливими для вирішення кількох задач, таких як обмеження пропускної здатності мережі, збільшення вартості зберігання даних та забезпечення високої якості контенту для користувачів. Відсутність оптимального підходу до завантаження може призвести до затримки у завантаженні відео, зниження якості зображення та, як наслідок, втрати аудиторії [20]. Сучасні алгоритми стиснення, такі як H.264, H.265/HEVC, VP9, AV1 та інші, пропонують різні підходи до стиснення відео, кожен із яких має свої переваги та недоліки залежно від використання. Існує потреба у проведенні дослідження та впровадження інноваційних алгоритмів стиснення, що набуває особливого практичного значення.

Метою роботи є аналіз сучасних технологій стискування відео, оцінка їх ефективності для сервісів відеохостингу, створення сучасних моделей та розробка рекомендацій щодо оптимального вибору та застосування методів стиснення для різних умов використання.

Об'єктом дослідження є процеси стиснення відео для передачі та зберігання на платформі відео-хостингу.

Предметом дослідження є методи та алгоритми стиснення відео, їх ефективність, зокрема вплив на якість відео, швидкість кодування та розмір кінцевих файлів.

Результати дослідження можуть бути використані для вдосконалення технологій обробки відео-контенту на платформі відео-хостингу. Запропоновані рекомендації сприяють підвищенню ефективності роботи сервісів через зменшення завантаження в мережу, оптимізацію зберігання даних та забезпечення високої якості контенту. Це, у свою чергу, сприятиме підвищенню задоволеності користувачів і конкурентоспроможності відео-сервісів.

РОЗДІЛ 1.

АНАЛІЗ ІСНУЮЧОГО СТАНУ СТИСНЕННЯ ВІДЕО ТА ЧИННИХ СТАНДАРТІВ

1.1. Аналіз існуючого стану стиснення відео для сервісів відео-хостингу

Відео є важливим засобом комунікації та розваг протягом багатьох десятиліть. В останні кілька років спостерігається постійне збільшення обсягу мережевого трафіку, що несе відеоконтент. Це створює простір для досліджень у сфері відеопотоків. Потокове передавання – це не що інше, як передача відеоконтенту і перегляд відеоконтенту до того, як він буде повністю завантажений на стороні клієнта. Відео транслюється безпосередньо з серверів і доступ до нього на стороні клієнта здійснюється за допомогою різних електронних пристроїв.

Високоякісне відео вимагає високої пропускної здатності. Якщо пропускна здатність низька, це призводить до низької якості відео на стороні кінцевого користувача. У таких випадках використовується стиснення відео разом з трансрейтингом щоб вписатися в доступну смугу пропускання. Трансляція – це процес, за допомогою якого відеофайли конвертуються зі зниженим бітрейтом, зберігаючи при цьому вихідний формат.

Захист відеоконтенту в потоковому передаванні – ще одне завдання яке вимагають такі додатки, як відеоконференції, відео на вимогу та платне телебачення. У кожній програмі автентифікація повинна бути підтверджена для доступу до відеоконтенту. високий рівень безпеки потрібний для відеоконференцій, в той час як для VOD і платного телебачення користувач повинен довести, що він є автентифікованим користувачем, щоб отримати доступ до відеоконтенту.

У сучасному світі стрімке зростання популярності відео-контенту та розвиток платформи відео-хостингу, таких як YouTube, TikTok, Instagram, Vimeo

тощо, обумовили необхідність ефективного тиснення відео. Основною сферою призначення технологій є забезпечення максимальної якості відео при мінімально можливому обсязі даних, що дозволяє зменшити завантаження в мережу, економити ресурси для зберігання та зменшити витрати контенту на передачу.

Стиснення відео базується на стандартах кодування, які постійно вдосконалюються для підвищення ефективності. Найпоширенішими стандартами на сьогодні є:

➤ H.264 (Advanced Video Coding, AVC) – один із найпоширеніших стандартів, що забезпечує високу якість при низькому бітрейті. Використовується для потокової передачі відео, зберігання контенту та трансляцій.

➤ H.265 (High Efficiency Video Coding, HEVC) – наступник H.264, який забезпечує приблизно вдвічі ефективніше тиснення даних при тій же якості. Цей стандарт дозволяє передавати відео у форматах 4K та 8K, що робить його ідеальним для сучасної платформи.

➤ VP9 – стандарт, розроблений Google, який є відкритим і широко використовується для YouTube. VP9 забезпечує ефективне тиснення відео і є конкурентом H.265.

➤ AV1 – новий стандарт кодування відео, який забезпечує ще кращу стійкість із H.265 та VP9. Він розроблений консорціумом Alliance for Open Media і активно впроваджується на платформах.

Основними викликами у стиску відео для відео-хостингів є:

1. Різноманітність форматів та роздільної здатності – відеоконтент може бути знятий у різних форматах (HD, Full HD, 4K, 8K), що потребує гнучких рішень для друку.

2. Сумісність з іншими пристроями – сервіси відео-хостингу повинні забезпечити коректне відтворення на різноманітних пристроях – від смартфонів до телевізорів.

3. Баланс між якістю та обсягом даних – зниження бітрейту часто призводить до втрати якості відео, що може негативно впливати на досвідчених користувачів.

4. Затримки у кодуванні – для потокової передачі відео необхідно забезпечити низьку затримку при стиску.

На окремій платформі відео-хостингу використовується H.264 завдяки його універсальності, проте ваш перехід до більш сучасних стандартів H.265, VP9 та AV1. Наприклад, YouTube активно використовує VP9 для зниження завантаження на сервери та забезпечення високої якості відео. Більшого H.265 впроваджується в преміальних сервісах, таких як Netflix, для передачі відео у форматі 4K.

Основними напрямками вдосконалення технологій стиснення є:

- ✓ інтеграція машинного навчання для адаптивного тиснення відео;
- ✓ оптимізація алгоритмів для роботи з відео з високою роздільною здатністю;
- ✓ змінення затримок при кодуванні для потокової передачі відео.

Таким чином, сучасний стан технологій стиснення відео для сервісів відео-хостингу характеризується впровадженням новітніх стандартів, оптимізацією процесів обробки відео та вирішенням викликів, пов'язаних із великим обсягом даних і вимогами до якості контенту.

1.2. Аналіз існуючих методів стиснення відео

Наразі існує два сімейства стандартів стиснення відео, розроблених під егідою Міжнародного союзу електров'язку телекомунікації (ITU-T, раніше Міжнародний консультативний комітет телеграфу і зв'язку Міжнародного консультативного комітету з питань телеграфу і зв'язку (ССІТТ) та Міжнародної організації зі стандартизації (ISO). Міжнародної організації зі стандартизації (ISO)).

Першим стандартом стиснення відео, який отримав широке визнання, був стандарт МСЕ H.261 [8], який був призначений для проведення відеоконференцій через інтегровані послуги цифрової мережі (ISDN). H.261 був прийнятий як стандарт у 1990 році.

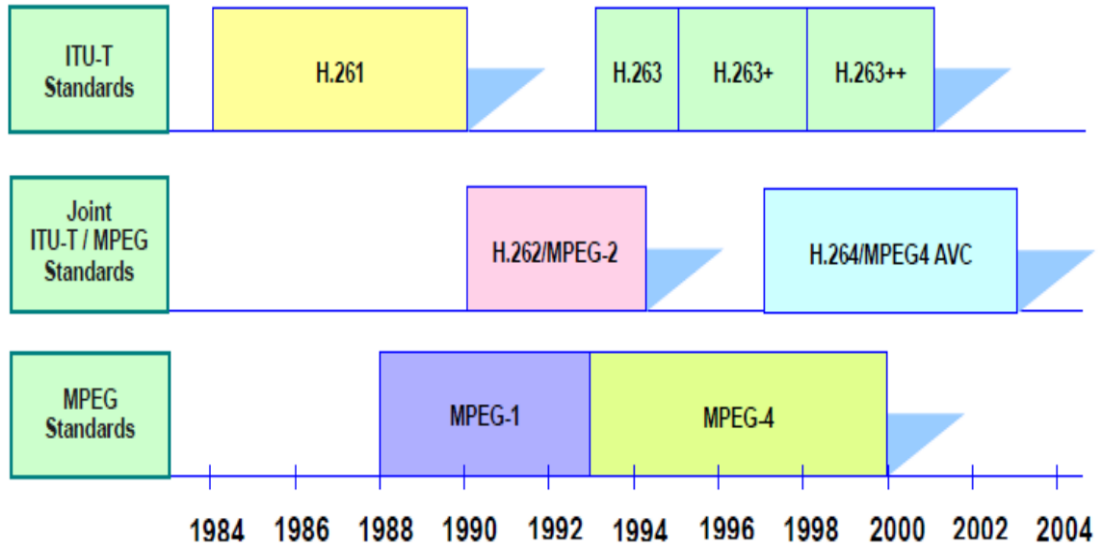


Рисунок 1.1 – Історія розробки стандартів стиснення відео

Він був розроблений для роботи при $p=1, 2, \dots, 30$ кратних базової швидкості передачі даних ISDN, або $p \times 64$ кбіт/с. Алгоритм кодування H.261 був розроблений для роботи з бітрейтами відео між 40 кбіт/с і 2 Мбіт/с. H.261 підтримує два розміри відео CIF і QCIF з використанням схеми дискретизації 4:2:0.

На рисунку 1.2 показано загальний кодера H.261, взятий із довідкової документації ITU, і показує взаємозв'язок між дискретним косинусним перетворенням (DCT), прогнозуванням і логікою оцінки руху [8].

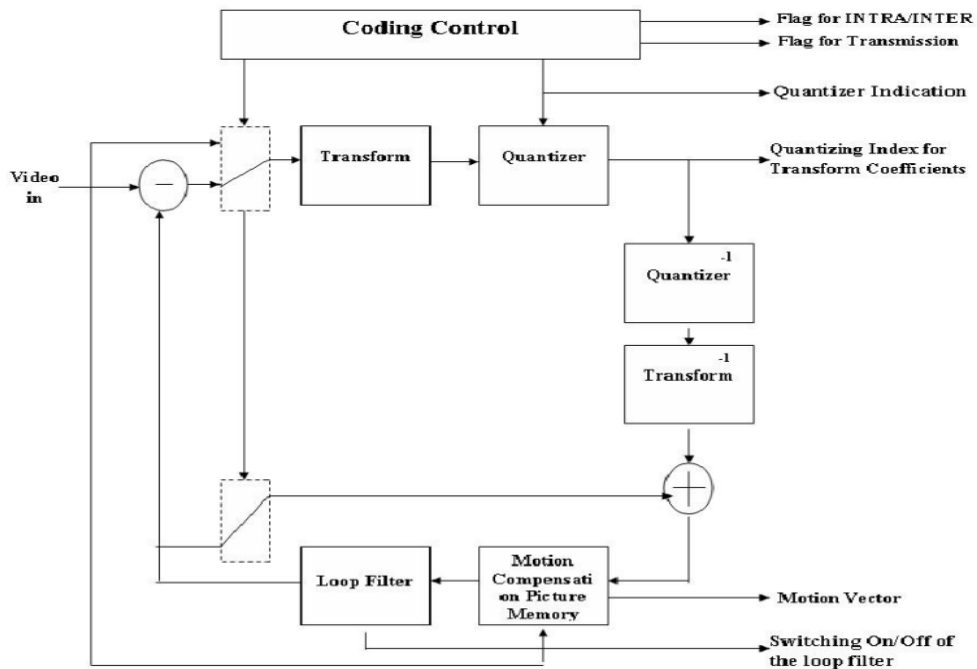


Рисунок 1.2 – Блок-схема вихідного кодера H.261

H.261 визначає два формати зображення: CIF (Common Intermediate Format), який має 288 ліній по 360 пікселів/лінія для інформації про яскравість та 144 x 180 пікселів для інформації про хромінанс; і QCIF (Quarter Common Intermediate Format), який має 144 лінії по 180 пікселів/лінія для яскравості та 72 x 90 пікселів для хромінансу.

Дані для кожного макроблоку складаються з заголовка макроблоку, за яким слідує дані для блоків. Один макроблок складається з чотирьох блоків яскравості та просторово відповідних блоків кольорової різниці, як показано на рис. 1.3.

Зображення кодуються у вигляді компонентів яскравості (Y) та двох компонентів кольорової різниці (CB та CR). Компоненти Y, CB і CR є функціями стандартних хроматичних компонентів (червоний, зелений, синій) і визначені в рекомендації CCIR 601.

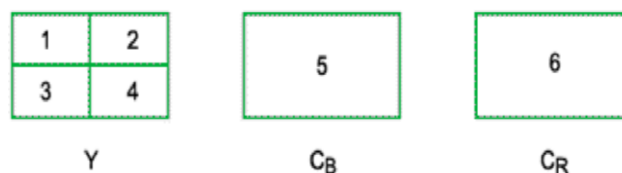


Рисунок 1.3 – Розташування блоків у межах макроблоку за стандартом H.261

У 1993 році ІТУ-Т ініціював стандартизацію з основною метою удосконалення відеотелефонії через телефонну мережу загального користування (ТфОП) (звичайні аналогові телефонні лінії), де загальна доступна швидкість передачі даних становить лише близько 33,6 кбіт/с. Частина стандарту, що відповідає за стиснення відео, називається H.263, і його перша реалізація була прийнята в 1996 році [23]. Вдосконалена версія H.263, H.263 Version 2 (V2), була прийнята в 1997 році, а абсолютно новий алгоритм, який спочатку називався H.26L, в даний час допрацьовується як H.264/AVC.

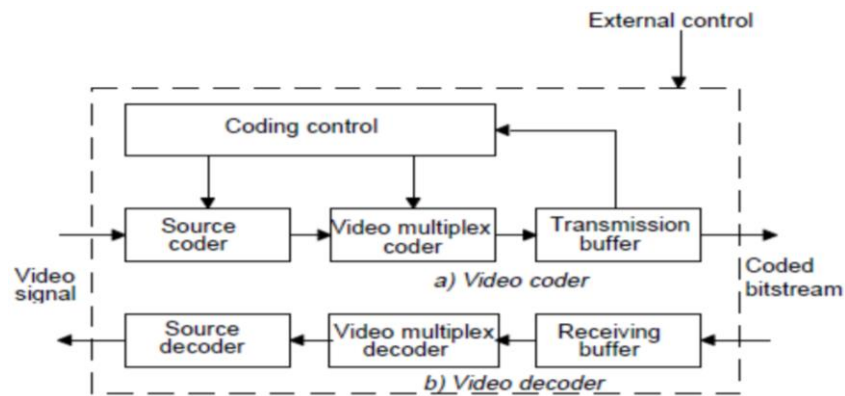


Рисунок 1.4 – Блок-схема відеокодека H.263

H.263 було створено після H.261 з акцентом на забезпечення кращої якості за ще нижчих бітрейтів. Однією з основних цілей при розробці H.263 було забезпечення передачі відео через звичайні телефонні модеми, які на той час працювали на швидкості 28,8 Кбіт/с. Цільовий діапазон роздільної здатності варіювався від SQCIF (128x96) до CIF. Основні алгоритми подібні до H.261, але включають кілька додаткових функцій.

H.263 було оптимізовано для широкого діапазону бітрейтів, а не лише для 64 Кбіт/с, як це було в H.261. Іншим покращенням є підтримка п'яти роздільних здатностей. Окрім CIF (Common Intermediate Format) і QCIF (Quarter Common Intermediate Format), які підтримуються H.261, H.263 також підтримує SQCIF, 4CIF і 16CIF.

- ✓ 4CIF має роздільну здатність у 4 рази більшу за CIF.
- ✓ 16CIF має роздільну здатність у 16 разів більшу за CIF.
- ✓ SQCIF приблизно вдвічі менше за QCIF.

Це означає, що H.263 може ефективно конкурувати зі стандартами MPEG.

До функцій, які роблять його подібним до MPEG, належать:

- ✓ Синтаксично-базоване арифметичне кодування (Syntax-based Arithmetic Coding),
- ✓ Необмежені вектори руху (Unrestricted Motion Vectors),
- ✓ Режим розширеного прогнозування (Advance Prediction Mode),
- ✓ Кадрове прогнозування (Frame Prediction).

Експертна група з динамічних зображень (Moving Pictures Expert Group, MPEG) була створена ISO в 1988 році для розробки стандарту стиснення рухомих зображень (відео) і пов'язаного з ними звуку на цифрових носіях інформації (CD-ROM). Остаточний варіант стандарту, відомий як MPEG-1, був завершений у 1991 році і забезпечує якість відео та аудіо приблизно VHS зі швидкістю близько 1,5 Мбіт/с [19]. Другий етап їхньої роботи, широко відомий як MPEG-2, був розширенням MPEG-1, розробленим для розширення MPEG-1, розробленим для застосування в цифровому телебаченні та для вищих бітрейтів [20].

Третій стандарт, який мав називатися MPEG-3, спочатку передбачався для додатків з високою швидкістю передачі даних, таких як HDTV, але було визнано, що ці додатки також можуть бути використані у MPEG-2. Отже, ці цілі були загорнуті в MPEG-2 (стандарту MPEG-3 не існує). Наразі відеочастина стандартів цифрового телебачення (DTV) і телебачення високої чіткості (HDTV) на значній частині території Північної Америки, Європи та Азії базується на MPEG-2.

Таблиця 1.1 – Характеристика стандартів кодування відео

Стандарти кодування відео	Застосування	Бітрейт
H.261	Відеотелефонія та відеоконференції через ISDN	рх 64 кбіт/с
MPEG-1	Відео на цифрових носіях (CD-ROM)	1,5 Мбіт/с
MPEG-2	Цифрове телебачення	2-20 Мбіт/с
H.263	Відеотелефонія через PSTN	від 33,6 кбіт/с
MPEG-4	Кодування на основі об'єктів, синтетичний контент, інтерактивність, відео-стрімінг	Змінний
H.264/MPEG-4, частина 10 (AVC)	Покращене стиснювання відео	від 10 до 100 кбіт/с

Наступний стандарт відомий як MPEG-4, був розроблений для забезпечення покращеної ефективності стиснення відео і стійкості до помилок, а також розширення функціональних можливостей, включаючи об'єктну обробку, інтеграцію природних об'єктно-орієнтованих зображень, інтеграцію як природного, так і синтетичного (згенерованого комп'ютером) контенту, інтерактивність на основі контенту [21]. У табл. 1.1 наведено сучасні та нові стандарти стиснення відео [15].

1.3. Характеристики систем H.264/AVC та H.265/HEVC

HEVC забезпечує приблизно вдвічі краще стиснення даних, що дозволяє досягти порівняльної якості відео за зниженої швидкості передачі даних або ж суттєво покращити якість відео при аналогічному бітрейті. Крім того, цей

стандарт забезпечує роздільну здатність до 8192×4320, виключно з 8K UHD. У таблиці 1.2 наведено детальне порівняння між H.264 і H.265.

Таблиця 1.2 – Основні відмінності між H.264 та H.265 [15]

Показник	H.264/AVC	H.265/HEVC
Назва	Advanced Video Coding, MPEG 4 Part 10	MPEG-H, High-Efficiency Video Coding, Part 2
Дата затвердження	2003 рік	2013 рік
Еволюція	Наступний MPEG-2 Part, відомого як H.222/H.262	Наступний H.264/AVC або MPEG4-частина 10
Ключові покращення	Зниження бітрейту на 40-50% у порівнянні з MPEG-2 Part. Підтримка передачі джерел високої чіткості для онлайн і трансляції	Зниження бітрейту на 40-50% при збереженні такої ж видимої якості в порівнянні з H.264. Очікується підтримка Ultra High Definition, 2K, 4K для онлайн-трансляцій і мовлення
Підтримка до 8K	Ні. Підтримка лише до 4K	Так
Підтримка до 300 кадрів/с	Ні. Лише до 59.94 кадрів/с	Так

Інструменти забезпечення безпеки цифрового відеопотоку включають криптографію [9], цифрові підписи [43] і водяні знаки у відео [47]. Загальноприйнятно, що жодна технологія окремо не може забезпечити повний захист цифрового відеопотоку, тому криптографія, цифрові підписи та водяні знаки мають його роль у програмних системах безпеки.

Переважним використанням криптографії є те, що зашифрований відеопотік не може бути переглянутим або інтерпретованим, поки отримувач не знає ключа розшифрування. Однак криптографія не є універсальним рішенням, оскільки безпека системи шифрування залежить від захисту ключів

розшифрування. Ця вразливість була продемонстрована, коли система шифрування, що використовувалася для захисту DVD-відео, була зламана [37].

Водяні знаки у відео [47, 18] також пропонуються для захисту цифрових відеопотоків. Водяний знак – це сигнал, який вбудовується у відеопотік таким чином, щоб видимі спотворення від цього були мінімальними.

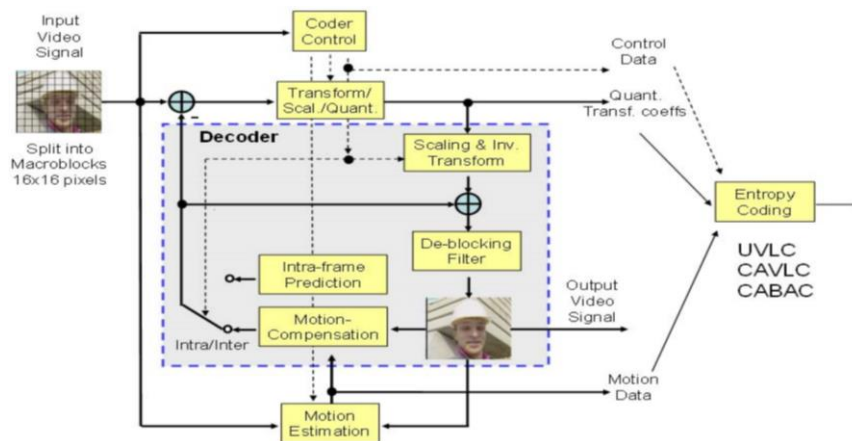


Рисунок 1.5 – Блок-схема типового процесу кодування для VCL H.264/AVC

HEVC/H.265 використовує складний процес кодування відео, розроблений для кількох цілей, таких як полегшення транспортування даних, стійкість до втрати даних і впровадження архітектури паралельної обробки. На рис. 1.6 представлена базова блок-схема кодера.

Опорний кодер HM, побудований на конфігурації випадкового доступу (RA), демонструє вищу ефективність кодування порівняно з конфігурацією низької затримки. Для відеоконтенту з частотою 24, 30, 50 і 60 кадрів/с період внутрішнього оновлення (intra period) встановлювався відповідно на 24, 32, 56 і 64 кадри, а розмір групи кадрів (GOP) задавався рівним 8 [10]. Крім того, використовувалися ієрархічні кадри, де параметр квантування (QP) збільшувався на 1 (що відповідає збільшенню кроку квантування на 12%) між кожним часовим рівнем.

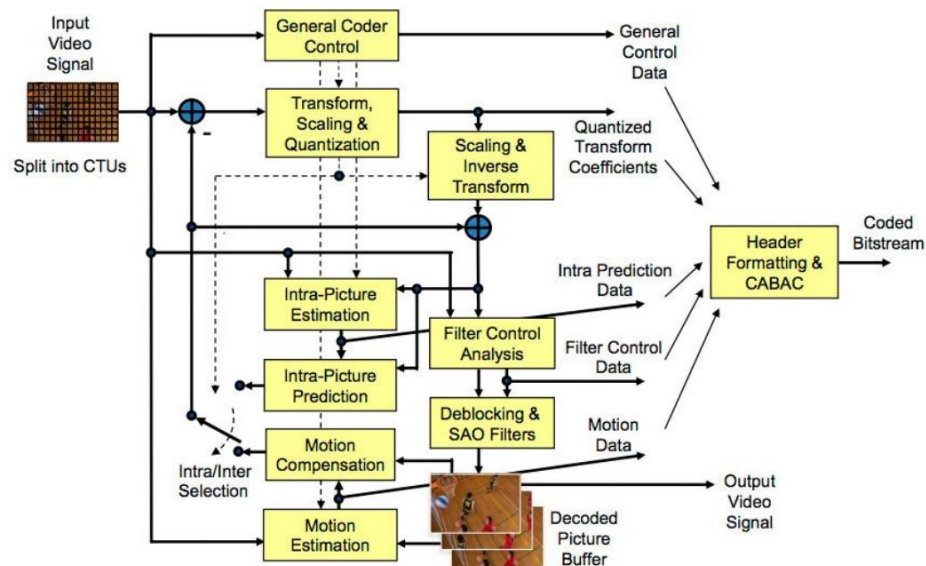


Рисунок 1.6 – Блок-схема кодера HEVC/H.265

Порядок кодування був встановлений як 0, 8, 4, 2, 1, 3, 6, 5, 7, і ці умови тестування були вибрані відповідно до. Для вибору інших параметрів кодера використовувалася конфігурація «CFG 16».

Нами проаналізовані застосовано основні технології стиснення відео та нові стандарти (H.264/AVC і H.265/HEVC). Також обговорювалися деякі питання безпеки під час трансляції відео через Інтернет. Встановлено, що запропоновані рішення для вирішення питань безпеки стануть найбільшим технічним бар'єром для передачі високоякісного цифрового відео через Інтернет.

1.4. Завдання кваліфікаційної роботи

У кваліфікаційній роботі проведено дослідження, що стосується розробки та впровадження ефективних методів створення відео для сервісів відео-хостингу із застосуванням сучасних стандартів і технологій, таких як нейронні мережі. Основною роботи є розробка модулів кодування відео, які забезпечують зменшення обсягу даних без значної втрати якості, а також підвищують економічну ефективність процесу.

Для досягнення мети визначено такі основні завдання:

- 1) аналіз існуючого стану та стандартів стиснення відео;
- 2) дослідження методів стиснювання відео з використанням нейронних мереж;
- 3) розробка модулів кодування відео;
- 4) аналіз навчальних і тестових баз даних;
- 5) проведення експериментів і оцінка результатів;
- 6) розробка заходів із охорони праці та безпеки в надзвичайних ситуаціях;
- 7) розрахувати економічний ефект від застосування запропонованих методів для послуг відео-хостингу.

Виконання цих завдань дозволяє створити ефективну систему стиснення відео, що відповідає сучасним вимогам якості та економічної доцільності, а також забезпечити безпеку процесів обробки даних.

РОЗДІЛ 2.

ОСНОВИ ТЕОРІЇ СТИСНЕННЯ ВІДЕО З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ ТА АНАЛІЗ ДАНИХ ДЛЯ НАВЧАННЯ МОДЕЛЕЙ

2.1. Особливості відеокодування та використання глибокого навчання для обробки зображень

Найновіший стандарт відеокодування HEVC, він передбачає використання сусідніх реконструйованих пікселів для прогнозування поточного блоку кодування з 33 кутовими режимами внутрішнього прогнозування, режимом постійного струму та планерним режимом, як показано на рис. 2.1

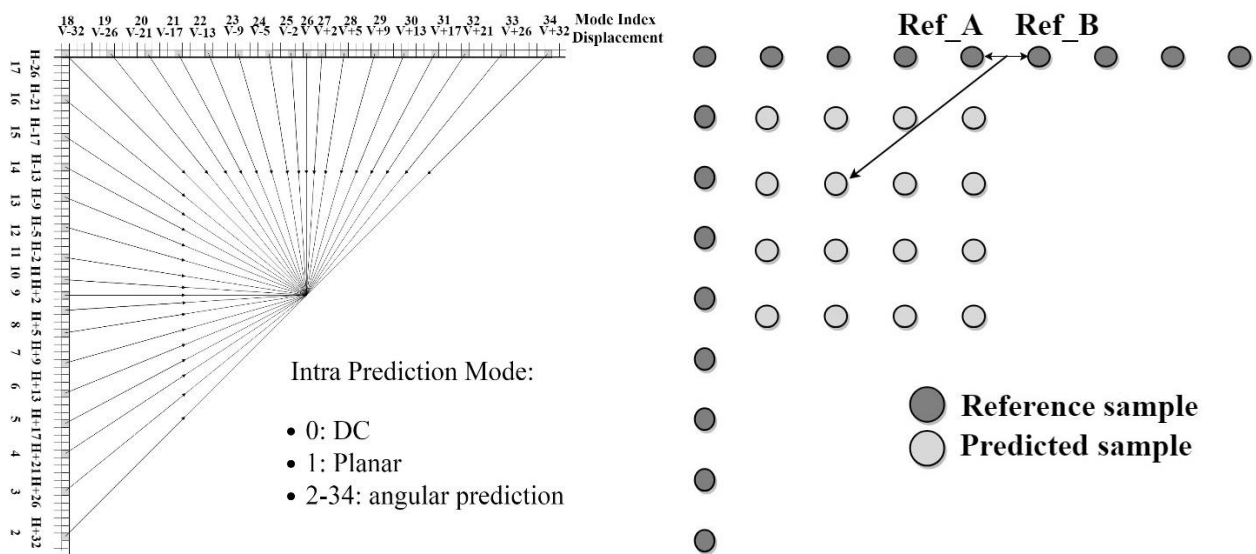
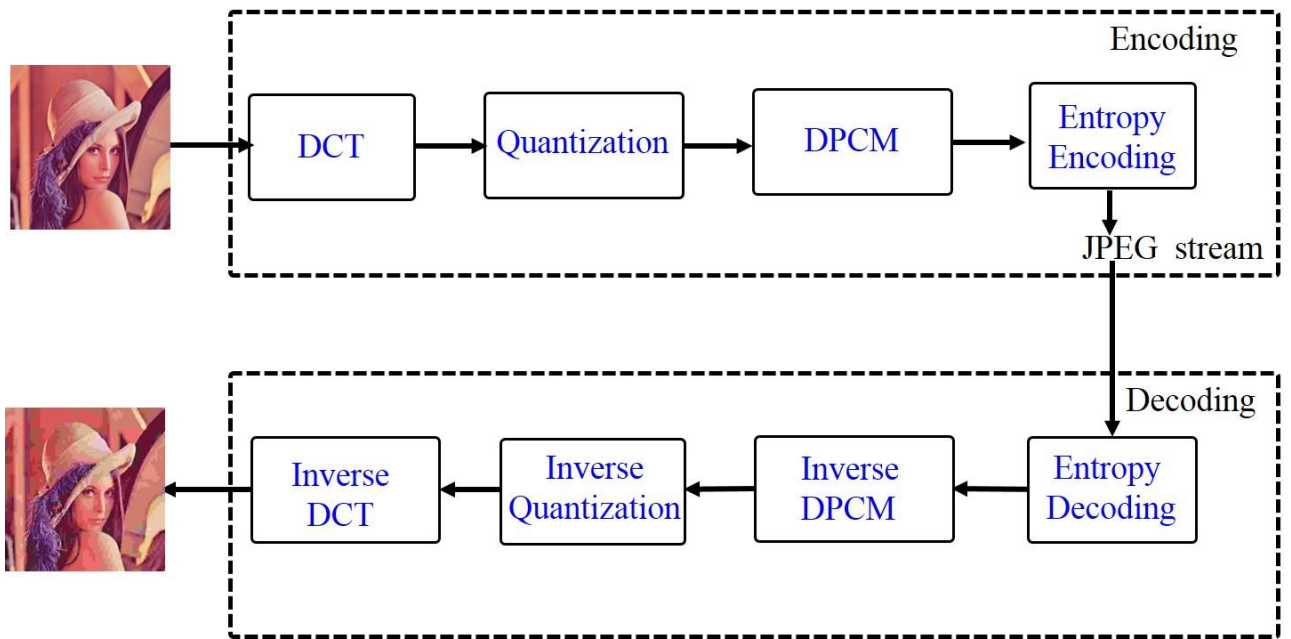


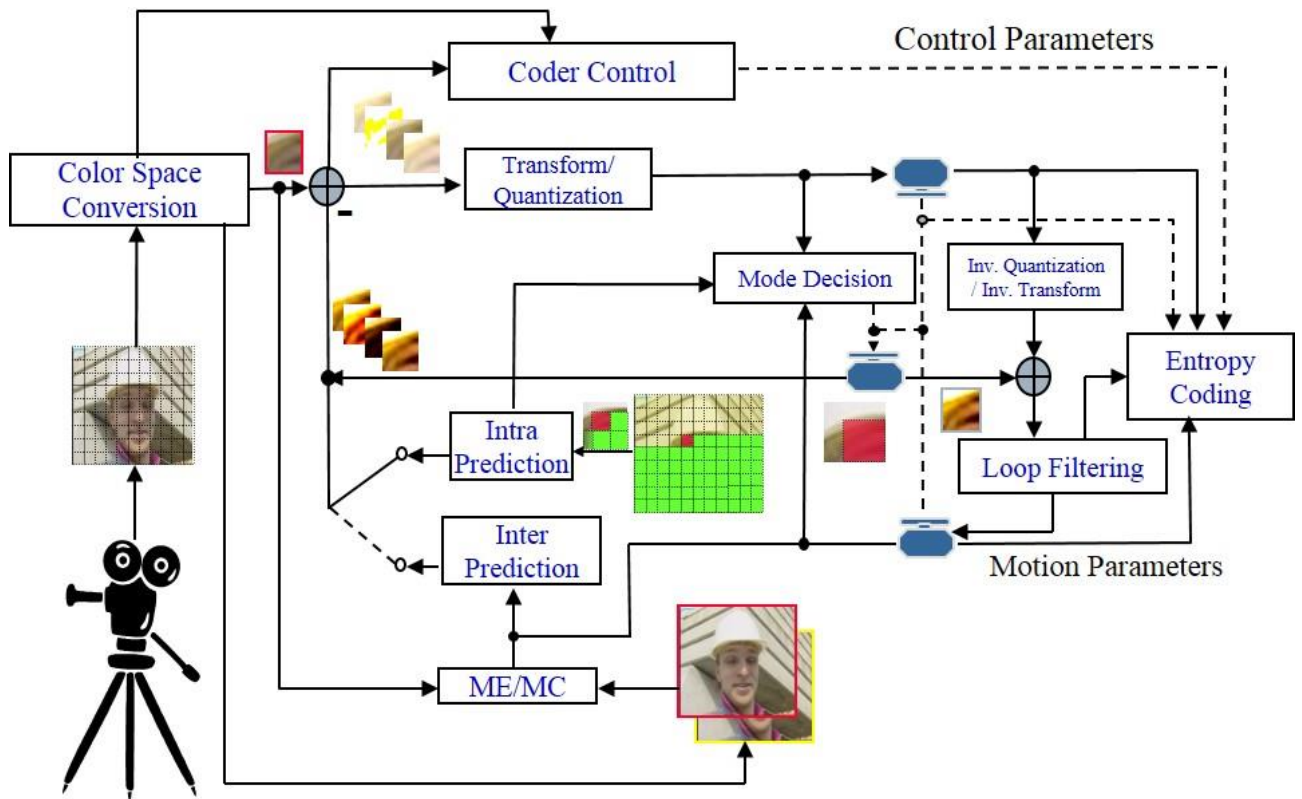
Рисунок 2.1 – Представлення внутрішнього стиснення HEVC:

а) різні внутрішні режими; б) екземпляр кутового стиснення

За міжкадрового стиснення HEVC покращується продуктивність кодування шляхом подальшого вдосконалення свого попередника, H.264/AVC, з різних точок зору, наприклад, збільшення різноманітності розділення PU, використання більшої кількості відводів інтерполяційного фільтра для компенсації руху підвибірки [10].



a)



б)

Рисунок 2.2 – Структура процесу стиснення зображень і відео:

а) стиснення JPEG; б) гібридне стиснення відео

При цьому передбачається вдосконалення кодування додаткової інформації, включаючи більш ймовірні режими (MPM) для кодування

внутрішнього режиму [49], розширене стиснення вектора руху (AMVP) і режим злиття для кодування предиктора вектора руху [12].

Останнім часом нейронні мережі, особливо згорточні нейронні мережі (CNN), досягли значного успіху в багатьох сферах, включаючи класифікацію зображень/відео, обробку та стиснення тощо (рис. 2.3).

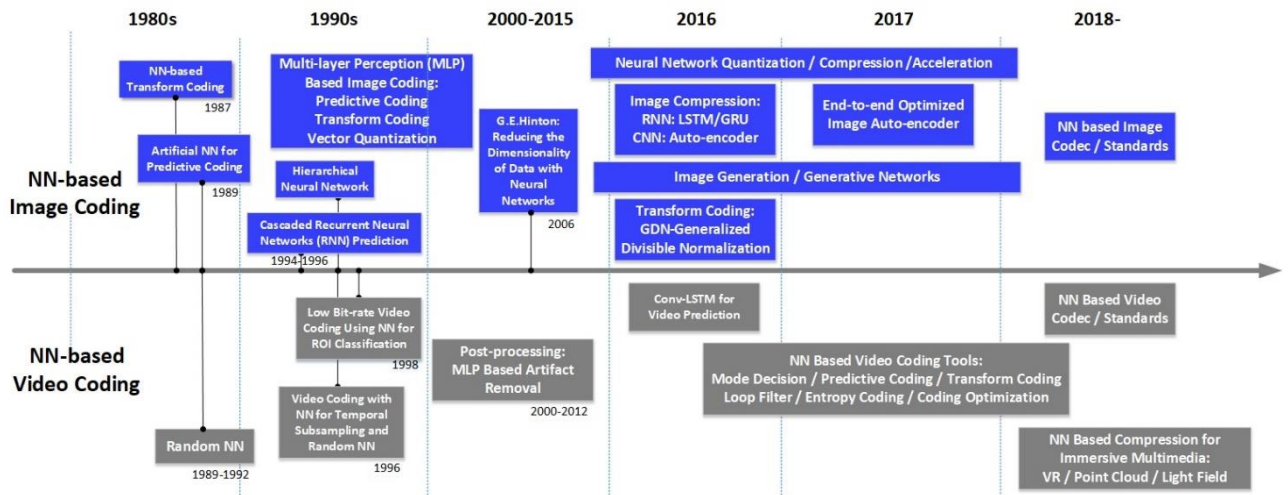


Рисунок 2.3 – Використовувані алгоритми стиснення на основі нейронних мереж

Заслуговує на увагу CNN, який зазвичай складається з одного або кількох згорткових шарів (рис. 2.4).

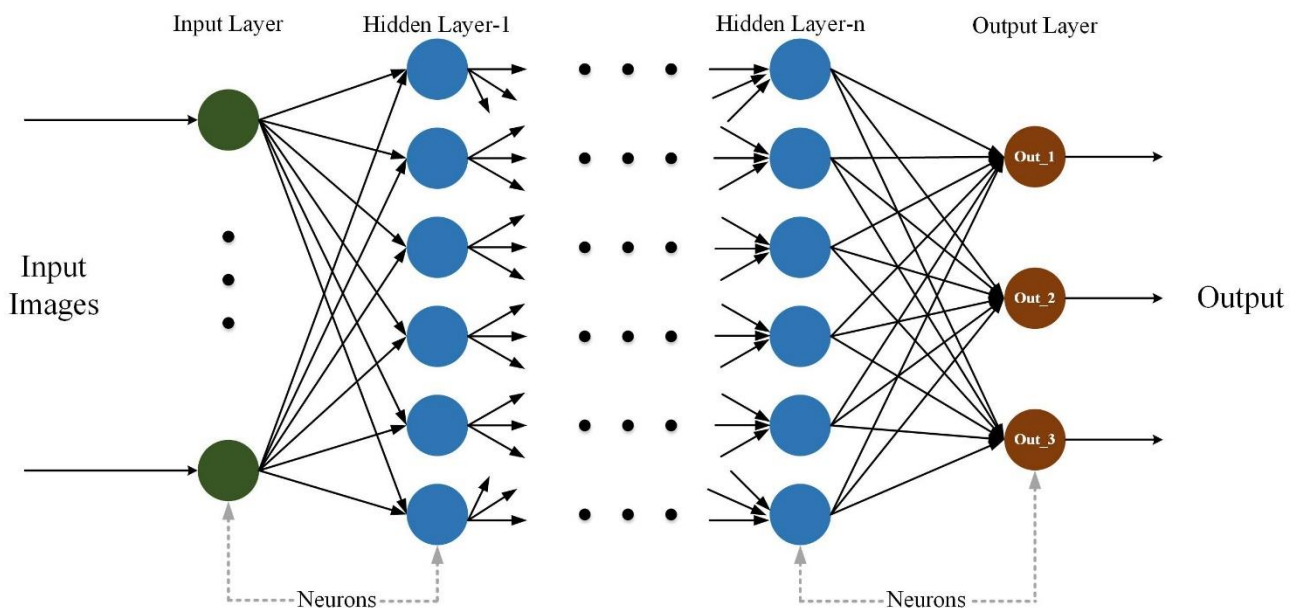


Рисунок 2.4 – Архітектура нейронної мережі

Зокрема, деякі завдання потребують кілька повністю пов'язаних шарів після шарів згортки. Параметри в цих шарах можна добре навчити на основі масиву даних із зразками зображень і відео, позначених для конкретних завдань. Навчена CNN може бути добре застосована для вирішення завдань класифікації, розпізнавання та прогнозування на тестових даних з високоефективною адаптивністю.

Якість сигналів передбачення, створених CNN, перевершила якість прогнозів на основі правил. Крім того, CNN можна інтерпретувати як екстрактори функцій для перетворення зображення та відео в простір функцій із компактним представленням, що є корисним для стиснення зображень та відео. Завдяки цим відмінним характеристикам CNN вважається перспективним рішенням для завдання стиснення. Тому, щоб добре зрозуміти існуючий розвиток CNN щодо стиснення зображень і відео, нами надається детальний огляд стиснення зображень і відео за допомогою нейронної мережі.

За останнє десятиліття методи машинного навчання (ML), особливо методи глибокого навчання, засновані на передових глибоких нейронних мережах (DNN), забезпечили революційний прогрес у різних сферах застосування комп'ютерного зору, зокрема для обробки і розуміння зображень/відео [27]. Структури DNN були призначені для обробки інформації із використанням структури нейронів у біологічних системах. Їх продуктивність поступово покращується за допомогою ітеративного процесу навчання з використанням великих обсягів навчального матеріалу.

ШНМ мають потужні можливості нелінійного моделювання та представлення даних і можуть вирішувати багато складних задач комп'ютерного зору, включаючи класифікацію, розпізнавання тощо [29]. Завдяки швидкому розвитку високопродуктивних обчислювальних технологій і значному збільшенню обсягів зберігання даних, ці методи глибокого навчання на основі даних також все частіше застосовуються в інших сферах, включаючи інтелектуальних асистентів, маркетинг і фінанси, а також процес прийняття рішень [11].

Насамперед представимо основні теорії DNN. Потім коротко розглядаються дві типові ШНМ – згорткові нейронні мережі (CNN) та генеративні змагальні мережі (GAN), та також розглянемо кілька популярних архітектур CNN і GAN, які застосовуються для глибокої обробки зображень і відео.

2.2. Основи теорії стиснення відео з використанням нейронних мереж

Сучасні нейронні мережі (deep neural networks, DNNs) є одним із ключових інструментів для вирішення складних задач, таких як розпізнавання об'єктів і класифікація. Ці мережі складаються з великої кількості штучних нейронів, що утворюють багат шарову структуру. Кожен нейрон містить налаштовувані параметри, які адаптивно оновлюються під час навчання моделі на основі вхідних даних (тренувальних наборів даних) [29].

Для навчання моделі використовується функція втрат, яка обчислює різницю між прогнозами моделі \hat{y} та фактичними даними y . Найпоширенішими функціями втрат є l_1 та l_2 -норми. Наприклад, l_2 -норма розраховується за формулою:

$$L(\theta) = \frac{1}{n} \sum_{i=1}^n \| y_i - \hat{y}_i \|^2, \quad (2.1)$$

де n – кількість вибірок у тренувальному наборі; y_i – реальні значення; \hat{y}_i – прогнозовані значення.

Одним із найефективніших методів оптимізації для навчання CNN-моделей є алгоритм Adam. Він поєднує переваги двох попередніх алгоритмів – Adaptive Gradient Algorithm (AdaGrad) та Root Mean Square Propagation (RMSProp).

Алгоритм Adam забезпечує швидку збіжність і стабільність тренувального процесу. Його оновлення параметрів виконується за формулами:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta} L(\theta_t), \quad (2.2)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_{\theta} L(\theta_t))^2, \quad (2.3)$$

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{v_t} + \varepsilon}. \quad (2.4)$$

де β_1, β_2 – параметри згладжування; η – швидкість навчання; ε – мале число для уникнення ділення на нуль.

Порівняння алгоритмів оптимізації наведено у таблиці 2.1.

Таблиця 2.1 – Порівняння алгоритмів оптимізації нейронних мереж

Алгоритм	Швидкість збіжності	Стабільність	Використання у відеосервісах
AdaGrad	Середня	Низька	Рідко
RMSProp	Висока	Середня	Часто
Adam	Дуже висока	Висока	Дуже часто

Розглянемо застосування Adam-оптимізатора для навчання CNN-моделей у задачах покращення стиснення відео. Основна ідея полягає в тому, щоб мінімізувати втрати, забезпечуючи при цьому високу якість відновлення кадрів.

На рисунку 2.5 представлено схему процесу навчання нейронної мережі для стиснення відео.

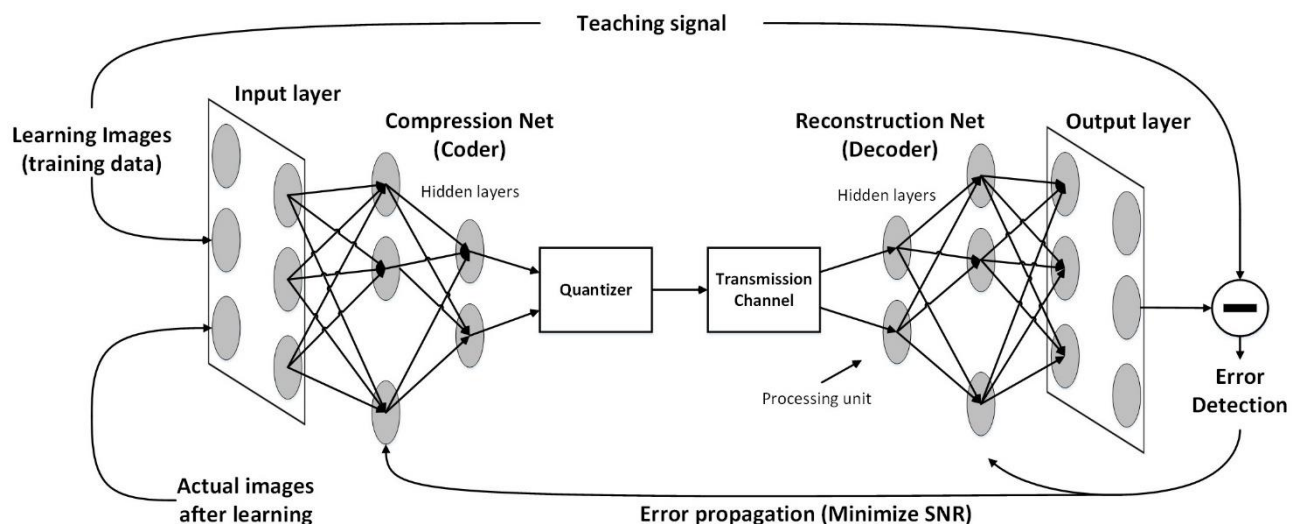


Рисунок 2.5 – Схема навчання нейронної мережі для задачі стиснення відео

Алгоритми оптимізації, такі як Adam, забезпечують швидке і стабільне навчання нейронних мереж, що робить їх ідеальним вибором для задач стиснення відео. Використання цих методів дозволяє досягти високої якості стисненого відео при збереженні низьких обчислювальних витрат.

2.3. Згорткова нейронна мережа (CNN) для стиснення відео

Згорткові нейронні мережі (Convolutional Neural Networks, CNNs) є однією з найпоширеніших архітектур у глибокому навчанні. Вони були вперше запропоновані для розпізнавання написаних поштових індексів [28], а згодом стали основою для аналізу зображень і відео. CNN імітують роботу зорової кори мозку, де кожен нейрон обробляє стимул у межах своєї рецептивної області [16].

CNN виконують згорткові операції над вхідними даними (наприклад, блоками зображень або кадрами відео), використовуючи ядра згортки. Кожне ядро має ваги, які налаштовуються під час навчання для оптимального виділення просторових або просторово-часових ознак. Архітектура CNN зазвичай включає:

1. Шари згортки (Convolutional Layers) – витягують ознаки на різних рівнях абстракції;
2. Шари підвибірки (Pooling Layers) – зменшують розмір ознак, щоб зменшити складність моделі та уникнути перенавчання;
3. Функції активації – вводять нелінійність для навчання складних патернів;
4. Нормалізація пакетів (Batch Normalization) – стабілізує навчання.

Основна операція в CNN – згортка, яка обчислюється за формулою:

$$S(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n) \cdot K(m, n), \quad (2.5)$$

де $S(i, j)$ – результат згортки для пікселя (i, j) ; I – вхідний блок зображення; K – ядро згортки; M, N – розміри ядра згортки.

На рисунку 2.6 показано основну структуру CNN для обробки зображень і відео.

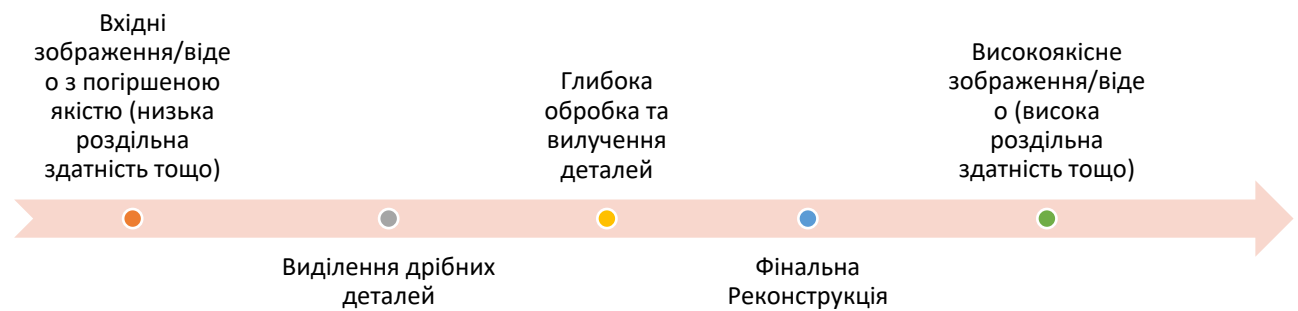


Рисунок 2.6 – Основна структура CNN для обробки зображень і відео

Щодо етапів у CNN під час стиснення відео, то вони подані у табл. 2.2.

Таблиця 2.2 – Типові етапи у CNN

Етап	Опис
Швидкий витяг ознак	Використання кількох шарів згортки для виділення низькорівневих ознак.
Глибока обробка ознак	Використання додаткових шарів, таких як залишкові блоки чи каскадні з'єднання.
Фінальна реконструкція	Реконструкція високоякісного зображення або відео на основі глибоких ознак.

У дослідженнях відео-стиснення та відновлення використовуються різні архітектури CNN. У таблиці 2.3 представлено їх ключові особливості.

Таблиця 2.3 – Основні особливості архітектур CNN

Архітектура	Основні особливості	Джерело
SRCNN	Перша CNN для надроздільної здатності зображень, 3 згорткові шари.	[14]
FSRCNN	Використовує 8 шарів із різними розмірами ядер.	[25]
VDSR	20 шарів із глобальним залишковим навчанням.	[30]
SRResNet	Включає залишкові блоки для покращення стабільності.	[50]

RDN	Поєднує залишкові блоки та щільні з'єднання.	[17]
-----	--	------

Ефективність CNN залежить від:

1. Глибини (кількість шарів);
2. Ширини (кількість карт ознак);
3. Кардинальності (розмір множини трансформацій).

Формула розрахунку кардинальності має вигляд:

$$C = \sum_{i=1}^N f(w_i, h_i), \quad (2.6)$$

де C – кардинальність; (w_i, h_i) – функція трансформації параметрів ядра.

Застосування CNN у задачах стиснення відео дозволяє значно покращити якість стиснення завдяки багатошаровій обробці ознак. Подальші дослідження слід спрямувати на оптимізацію архітектур із використанням сучасних механізмів та нових типів з'єднань.

2.4. Генеративні змагальні мережі (GANs)

Генеративні змагальні мережі (Generative Adversarial Networks, GANs) були вперше запропоновані Гудфеллоу та співавторами у 2014 році [17]. Ці мережі складаються з двох основних модулів – генератора та дискримінатора, що працюють за змагальним принципом. Основна структура GAN представлена на рисунку 2.7.

Генератор створює синтетичні дані, імітуючи реальні. Дискримінатор оцінює, чи є вхідні дані справжніми або синтетичними. У процесі навчання обидва модулі навчаються одночасно. Генератор намагається обдурити дискримінатор, створюючи більш реалістичні дані. Дискримінатор вдосконалюється, щоб краще розрізнити реальні та створені дані.

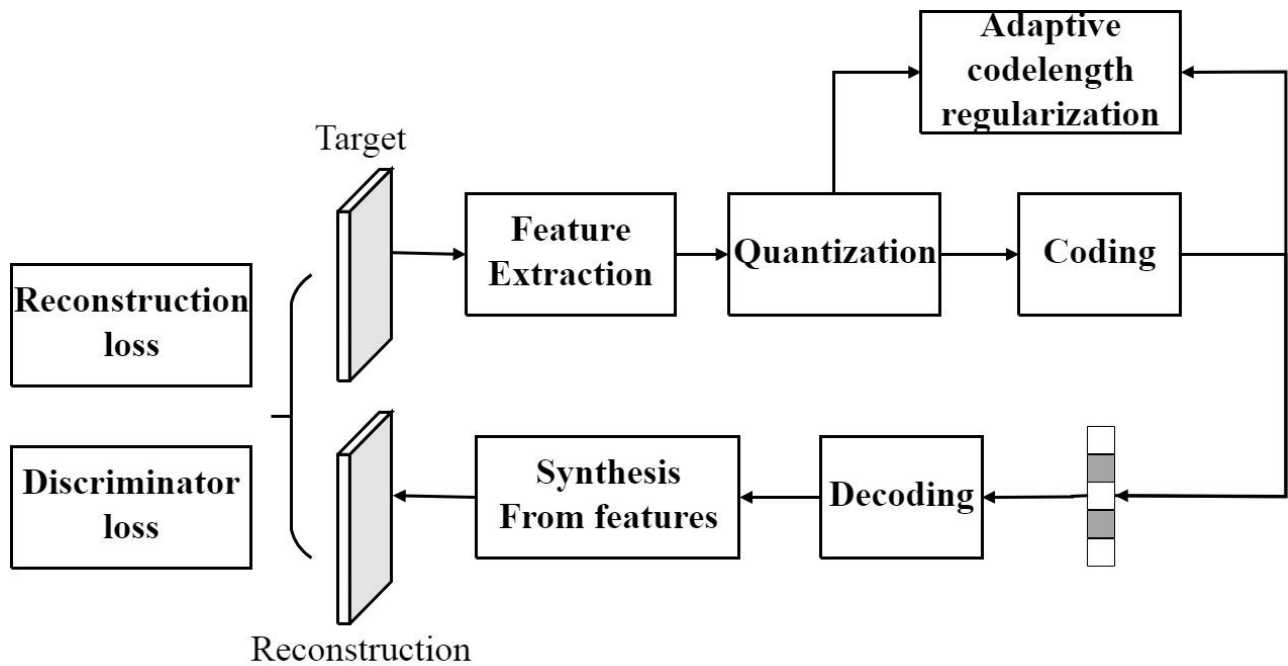


Рисунок 2.7 – Основна структура GAN

Формула функції втрат для генератора L_G та дискримінатора L_D записується як:

$$L_G = -\mathbb{E}_{z \sim p_z(z)} [\log(D(G(z)))], \quad (2.7)$$

$$L_D = -\mathbb{E}_{x \sim p_{\text{real}}} [\log(D(x))] - \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))], \quad (2.8)$$

де $G(z)$ – вихід генератора для випадкового шуму z ; $D(x)$ – ймовірність, що дискримінатор вважає вхід x реальним; p_{real} – розподіл реальних даних; $p_z(z)$ – розподіл випадкового шуму.

Розглянемо використання GAN у задачах відновлення відео. GAN дозволяють досягати більш високої візуальної якості та деталізації текстур у порівнянні з традиційними CNN [41]. Їх використання охоплює:

1. Суперроздільність зображень (SISR);
2. Видалення шумів;
3. Відновлення втрачених фрагментів відео.

Таблиця 2.4 – Поширені архітектури GAN для відновлення зображень і відео

Архітектура	Основні особливості
ADGAN	Використовує U-подібну структуру та стандартну методологію GAN.
SRResCGAN	Базується на залишкових блоках та використовує методологію RaGAN.
SRGAN	Поєднує стандартний GAN із перцептивною функцією втрат (VGG19).
PCARNGAN	Каскадні з'єднання та залишкові блоки, тренування стандартним GAN.
ESRGAN	Використовує алгоритм Relativistic GAN для тренування ESRResNet.
RCAGAN	Залишкові блоки уваги каналів (CA) та умовний GAN.
RFB-ESRGAN	Замінює щільні залишкові блоки блоками щільного рецептивного поля (RFB).

До переваг GAN належать:

- ✓ Покращення сприйняття якості результатів на основі перцептивних метрик;
- ✓ Генерація більш реалістичних текстур.

До недоліків GAN належать:

- ✓ Висока складність навчання та великі обчислювальні ресурси через спільне навчання генератора і дискримінатора;
- ✓ Нестабільність навчального процесу через змагальну стратегію.

GAN є потужним інструментом для підвищення якості відео та зображень, що робить їх перспективними для використання у відеохостингових сервісах. Подальші дослідження слід спрямовувати на вдосконалення стабільності навчання та оптимізацію обчислювальних витрат.

2.5. Вибір та аналіз навчальних та тестових баз даних

Тренувальні бази даних відіграють критичну роль в оптимізації продуктивності алгоритмів машинного навчання. Ретельно підібраний набір даних для тренування забезпечує надійну генералізацію моделі та мінімізує ризик перенавчання [51]. У відеокодуванні публічно доступні бази даних, спеціально розроблені для цієї галузі, залишаються обмеженими. Дослідники часто адаптують набори даних, спочатку розроблені для задач суперроздільної здатності, інтерполяції кадрів або класифікації. Нижче наведено аналіз основних тренувальних баз даних, що використовуються в дослідженнях відеокодування з використанням машинного навчання (табл. 2.5).

Таблиця 2.5 – Поширені архітектури GAN для відновлення зображень і відео

База даних	Основні характеристики	Застосування у відеокодуванні
1	2	3
BSDS [35]	500 RGB зображень (481×321), розроблених для сегментації зображень.	Тренування CNN-фільтрів циклів
ImageNet [40]	Понад 14 мільйонів RGB зображень, різні роздільності до 2848р, призначені для розпізнавання об'єктів.	Суперроздільність одного зображення
DIV2K [8]	1,000 RGB зображень з різноманітним контентом і високою роздільністю.	Алгоритми суперроздільної здатності

продовження табл. 2.5

1	2	3
---	---	---

UCF101 [42]	13,320 відео (320×240), що охоплюють 101 клас дій людини.	Інтерполяція кадрів і прогнозування руху
Kinetics [24]	650,000 відеокліпів у 700 класах дій з роздільностями до 480×360.	Навчання розпізнаванню дій людини
Vimeo [48]	89,800 послідовностей з роздільностями до 448×256, з акцентом на контрольовані рухи.	Оптичний потік і часовий суперрозділ.
Moments in Time [36]	1,000,000 відеокліпів у 339 класах дій з роздільністю 340×256.	Розпізнавання дій у відео.
YouTube UGC [46]	1,500 користувацьких послідовностей контенту (UGC) з роздільностями від 360р до 1080р.	Дослідження оцінки якості відео.
Tencent Video [44]	86 послідовностей у форматі YCbCr 4:2:0 (3840×2160), з глибиною кольору 8 або 10 біт.	Тренування CNN-засобів відеокодування.

Зазначені набори даних підкреслюють різноманітність у типах контенту зображень і відео, просторових роздільностях і призначених застосуваннях. Варто зазначити, що більшість із цих наборів даних не містять контенту з ультрависокою чіткістю (UHD) або послідовностей з динамічними текстурами.

Тестові бази даних є однаково важливими для оцінки продуктивності алгоритмів відеокодування. На відміну від тренувальних наборів даних, вони містять менше, але більш різноманітні відеопослідовності для перевірки генералізації моделі на різних типах контенту і сценаріях. Основні тестові бази даних представлені у таблиці 2.6.

Таблиця 2.6 – Характеристика основних тестових баз даних

Тестові бази даних	Опис	Покриття роздільностей
JVET-CTC SDR	19 послідовностей YCbCr 4:2:0, глибина 10 біт, роздільності: 240р, 480р, 1080р, 2160р.	Різноманітні (статичні та динамічні).
UVG Dataset	16 4K (3840×2160), 10-біт послідовностей із високою просторовою і часовою інформацією.	Контент з високою роздільністю.
Objective-1-fast	30 послідовностей YCbCr 4:2:0 (8-біт), роздільності: 360р, 720р, 1080р.	Природний і екранний контент.

Представлені тестові бази даних надають різноманітні еталони для оцінки ефективності запропонованих алгоритмів кодування за стандартних умов. Комбінація тренувальних і тестових баз даних забезпечує надійну основу для перевірки підходів відеокодування, що базуються на глибокому навчанні.

РОЗДІЛ 3.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ СТИСНЕННЯ ВІДЕО ДЛЯ СЕРВІСІВ ВІДЕО-ХОСТИНГУ

3.1. Розробка модулів кодування на основі CNN

Глибокі нейронні мережі (CNN) активно застосовуються для вдосконалення індивідуальних інструментів відеокодування, інтегрованих у стандартні кодеки. Вони оптимізують такі процеси, як внутрішнє передбачення, оцінка руху, перетворення, квантування, ентропійне кодування, постобробка та фільтрацію в циклі. Нижче наведено характеристики цих підходів.

3.1.1. Модуль кодування 1 (Post Processing - PP)

Під час втратного кодування можуть виникати видимі артефакти, такі як блокування, розмиття чи смуги. Постобробка (Post Processing, PP) зменшує ці артефакти після декодування. На рисунку 3.1 показано робочий процес з використанням CNN для постобробки.

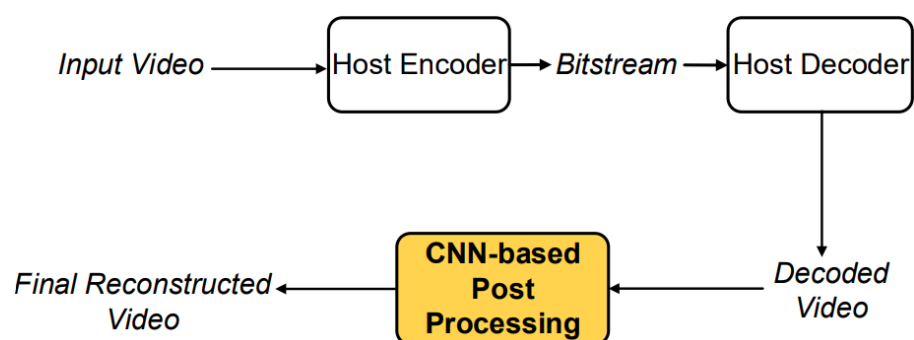


Рисунок 3.1 – Робочий процес з використанням CNN для постобробки

У цьому підході мережа приймає кожен декодований кадр як вхід і генерує фінальний реконструйований кадр.

Модуль постобробки (Post Processing, PP) використовується для зменшення артефактів, що виникають під час втратного кодування відео, таких як блокування, розмиття та смуги. Цей підхід базується на глибоких нейронних мережах (CNN), які дозволяють підвищити якість реконструйованих кадрів.

Нами розроблено модуль кодування 1 (Post Processing, PP), код якого подано у додатку А. Для реалізації постобробки (Post Processing, PP) використовується нейронна мережа, що складається з кількох згорткових шарів. Нижче наведено приклад функції, що визначає архітектуру CNN (рис. 3.2).

```
# Визначення моделі CNN для постобробки
def build_pp_model(input_shape):
    model = Sequential()

    # Перший згортковий шар
    model.add(Conv2D(64, (3, 3), padding='same', input_shape=input_shape))
    model.add(Activation('relu'))
    model.add(BatchNormalization())

    # Додаткові згорткові шари
    for _ in range(3):
        model.add(Conv2D(64, (3, 3), padding='same'))
        model.add(Activation('relu'))
        model.add(BatchNormalization())

    # Вихідний шар
    model.add(Conv2D(3, (3, 3), padding='same')) # RGB вихід
    model.add(Activation('sigmoid'))

    return model
```

Рисунок 3.2 – Фрагмент коду функції, що визначає архітектуру CNN

Ця архітектура включає шари BatchNormalization для стабілізації навчання та активаційні функції ReLU, що забезпечують нелінійність.

У подальшому виконується обробка кадрів. Для обробки декодованих кадрів спочатку завантажується зображення, нормалізується і подається у модель CNN. Нижче наведено код для завантаження та обробки кадру (рис. 3.3).

```
# Завантаження декодованого кадру
def load_frame(filepath):
    frame = cv2.imread(filepath)
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Перетворення в RGB
    frame = frame / 255.0 # Нормалізація до [0, 1]
    return frame

# Постобробка кадру
def post_process_frame(model, frame):
    input_frame = np.expand_dims(frame, axis=0) # Додавання розміру пакету
    output_frame = model.predict(input_frame)
    return np.squeeze(output_frame, axis=0) # Видалення розміру пакету
```

Рисунок 3.3 – Фрагмент коду для завантаження та обробки кадру

Основний робочий процес для модуля постобробки включає завантаження декодованого кадру, передавання кадру через модель CNN та збереження покращеного кадру у форматі зображення (рис. 3.4).

```

if __name__ == "__main__":
    # Визначення розміру вхідних даних
    input_shape = (None, None, 3) # Висота, ширина, канали

    # Створення моделі
    pp_model = build_pp_model(input_shape)

    # Завантаження декодованого кадру
    decoded_frame = load_frame("decoded_frame.jpg")

    # Постобробка кадру
    enhanced_frame = post_process_frame(pp_model, decoded_frame)

    # Збереження покращеного кадру
    enhanced_frame = (enhanced_frame * 255).astype(np.uint8) # Перетворення у [0, 255]
    enhanced_frame = cv2.cvtColor(enhanced_frame, cv2.COLOR_RGB2BGR) # Повернення у BGR
    cv2.imwrite("enhanced_frame.jpg", enhanced_frame)

```

Рисунок 3.4 – Фрагмент коду для передавання кадру через модель CNN та збереження покращеного кадру у форматі зображення

Реалізація модуля постобробки (Post Processing, PP) за допомогою CNN дозволяє значно зменшити артефакти втратного кодування та покращити якість відео. Запропонований підхід є гнучким і може бути адаптований до різних умов та типів відеоконтенту.

3.1.2. Модуль кодування 2 (Фільтрація в циклі - ILF)

Фільтрація в циклі (ILF) застосовується як на стороні енкодера, так і декодера, покращуючи якість реконструйованих кадрів. Рисунок 3.4 демонструє архітектуру енкодера з CNN-фільтром в циклі. Фільтр дозволяє використовувати покращені кадри як опорні для подальшого прогнозування руху, зменшуючи залишкові сигнали.

Фільтрація в циклі (In-loop Filtering, ILF) є важливим етапом відеокодування, спрямованим на покращення якості реконструйованих кадрів та їх подальше використання як опорних кадрів. Цей підхід базується на

використанні глибоких нейронних мереж (CNN), які дозволяють усувати артефакти та підвищувати точність прогнозування руху.

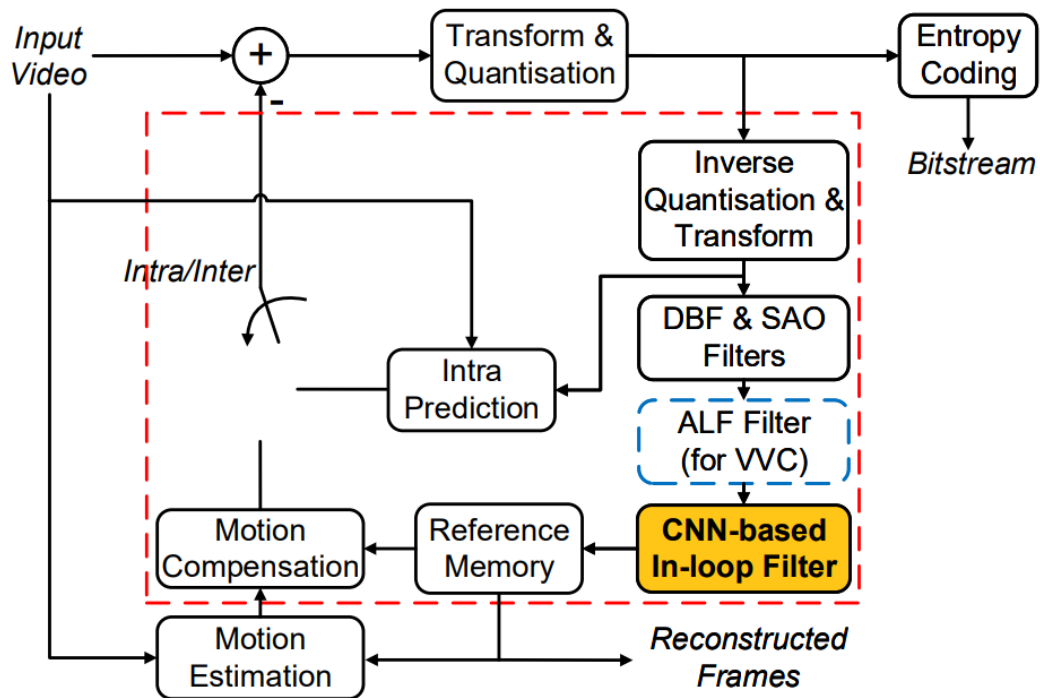


Рисунок 3.5 – Архітектуру енкодера з CNN-фільтром в циклі

Нами розроблено модуль кодування 2 (Фільтрація в циклі, ILF), код якого подано у додатку Б. Для фільтрації в циклі використовується модель CNN, яка складається з кількох згорткових шарів. Ця модель включає згорткові шари, функції активації ReLU та шари нормалізації BatchNormalization для забезпечення стабільності навчання.

Процес фільтрації забезпечується наступним чином. Реконструйовані кадри обробляються через модель CNN для покращення їх якості. Нижче наведено код завантаження та обробки кадру (рис. 3.6).

```
# Завантаження реконструйованого кадру
def load_frame(filepath):
    frame = cv2.imread(filepath)
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Перетворення в RGB
    frame = frame / 255.0 # Нормалізація до [0, 1]
    return frame

# Фільтрація в циклі
def in_loop_filter(model, frame):
    input_frame = np.expand_dims(frame, axis=0) # Додавання розміру пакету
    filtered_frame = model.predict(input_frame)
    return np.squeeze(filtered_frame, axis=0) # Видалення розміру пакету
```

Рисунок 3.6 – Фрагмент коду для завантаження та обробки кадру

Основний робочий процес для фільтрації в циклі включає (рис. 3.7):

- ✓ Завантаження реконструйованого кадру;
- ✓ Передавання кадру через модель CNN;
- ✓ Збереження покращеного кадру у форматі зображення.

```

if __name__ == "__main__":
    # Визначення розміру вхідних даних
    input_shape = (None, None, 3) # Висота, ширина, канали

    # Створення моделі
    ilf_model = build_ilf_model(input_shape)

    # Завантаження реконструйованого кадру
    reconstructed_frame = load_frame("reconstructed_frame.jpg")

    # Фільтрація в циклі
    enhanced_frame = in_loop_filter(ilf_model, reconstructed_frame)

    # Збереження покращеного кадру
    enhanced_frame = (enhanced_frame * 255).astype(np.uint8) # Перетворення у [0, 255]
    enhanced_frame = cv2.cvtColor(enhanced_frame, cv2.COLOR_RGB2BGR) # Повернення у BGR
    cv2.imwrite("enhanced_frame_ilf.jpg", enhanced_frame)

```

Рисунок 3.7 – Фрагмент коду для фільтрації в циклі

Реалізація модуля фільтрації в циклі за допомогою CNN забезпечує значне покращення якості реконструйованих кадрів. Це, в свою чергу, підвищує ефективність відеокодування за рахунок покращення прогнозування руху та зменшення залишкових сигналів.

3.1.3. Модуль кодування 3 (Адаптація просторової роздільності - SRA)

Адаптація просторової роздільності (SRA) передбачає зменшення роздільності оригінальних кадрів під час кодування та їх відновлення через CNN-суперроздільність при декодуванні. На рисунку 3.8 зображено робочий процес SRA. Відеокадри просторово зменшуються фільтром Lanczos3 перед компресією, а потім реконструюються до повної роздільності.

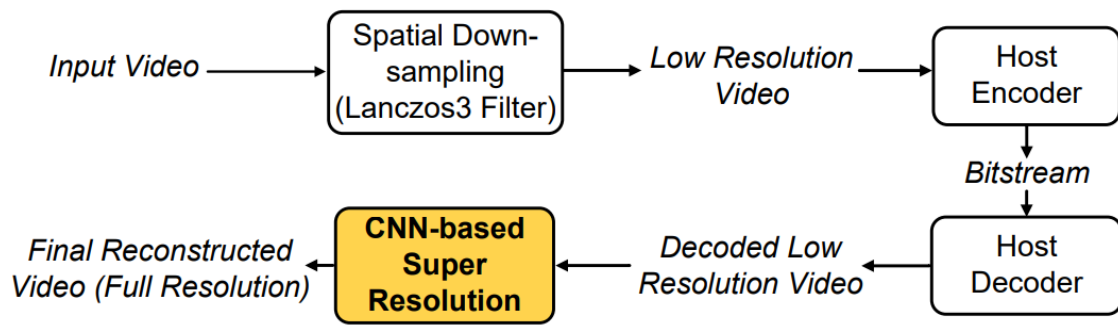


Рисунок 3.8 – Робочий процес кодування з модулем SRA на базі CNN

Модуль адаптації просторової роздільності (Spatial Resolution Adaptation, SRA) використовується для зменшення роздільності відеокадрів перед кодуванням і відновлення їх до початкової роздільності після декодування. Цей підхід базується на глибоких нейронних мережах (CNN), що забезпечують високу якість реконструйованих кадрів навіть при значному зменшенні роздільності.

Нами розроблено модуль кодування з (Spatial Resolution Adaptation, SRA), код якого подано у додатку В. Для адаптації просторової роздільності використовується CNN-модель, що включає кілька згорткових шарів та шарів підвищення роздільності (Upsampling). Нижче наведено код функції для створення такої моделі SRA (рис. 3.9).

```

# Визначення моделі CNN для SRA
def build_sra_model(input_shape):
    model = Sequential()

    # Перший згортковий шар
    model.add(Conv2D(64, (3, 3), padding='same', input_shape=input_shape))
    model.add(Activation('relu'))
    model.add(BatchNormalization())

    # Додаткові згорткові шари
    for _ in range(3):
        model.add(Conv2D(64, (3, 3), padding='same'))
        model.add(Activation('relu'))
        model.add(BatchNormalization())

    # Шар підвищення роздільності
    model.add(UpSampling2D(size=(2, 2), interpolation='nearest'))

    # Вихідний згортковий шар
    model.add(Conv2D(3, (3, 3), padding='same')) # RGB вихід
    model.add(Activation('sigmoid'))

    return model
  
```

Рисунок 3.9 – Фрагмент коду для функції створення моделі SRA

Адаптація просторової роздільності складається з двох основних етапів – зменшення роздільності та її відновлення за допомогою CNN. Нижче наведено код функцій для реалізації цих етапів (рис. 3.10).

```
# Зменшення роздільності кадру
def downsample_frame(frame, scale=2):
    height, width = frame.shape[:2]
    new_height, new_width = height // scale, width // scale
    downsampled_frame = cv2.resize(frame, (new_width, new_height), interpolation=cv2.INTER_AREA)
    return downsampled_frame

# Відновлення роздільності за допомогою CNN
def upscale_frame(model, frame):
    input_frame = np.expand_dims(frame, axis=0) # Додавання розміру пакету
    upscaled_frame = model.predict(input_frame)
    return np.squeeze(upscaled_frame, axis=0) # Видалення розміру пакету
```

Рисунок 3.10 – Фрагмент коду для функції адаптація просторової роздільності

Основний робочий процес для SRA включає (рис. 3.11):

- ✓ Завантаження вихідного кадру;
- ✓ Зменшення роздільності кадру;
- ✓ Відновлення роздільності за допомогою CNN;
- ✓ Збереження відновленого кадру.

```
if __name__ == "__main__":
    # Визначення розміру вхідних даних
    input_shape = (None, None, 3) # Висота, ширина, канали

    # Створення моделі
    sra_model = build_sra_model(input_shape)

    # Завантаження вихідного кадру
    original_frame = cv2.imread("original_frame.jpg")
    original_frame = cv2.cvtColor(original_frame, cv2.COLOR_BGR2RGB) # Перетворення в RGB
    original_frame = original_frame / 255.0 # Нормалізація до [0, 1]

    # Зменшення роздільності
    downsampled_frame = downsample_frame(original_frame)

    # Відновлення роздільності
    upscaled_frame = upscale_frame(sra_model, downsampled_frame)

    # Збереження відновленого кадру
    upscaled_frame = (upscaled_frame * 255).astype(np.uint8) # Перетворення у [0, 255]
    upscaled_frame = cv2.cvtColor(upscaled_frame, cv2.COLOR_RGB2BGR) # Повернення у BGR
    cv2.imwrite("upscaled_frame.jpg", upscaled_frame)
```

Рисунок 3.11 – Фрагмент коду робочого процесу для SRA

Модуль адаптації просторової роздільності забезпечує ефективне зменшення обсягу даних для кодування та високу якість реконструкції

відеокадрів. Використання CNN дозволяє зберігати деталі навіть при значному зменшенні роздільності.

3.1.4. Модуль кодування 4 (Адаптація ефективної глибини бітів - EBDA)

EBDA дозволяє адаптувати глибину бітів під час кодування для підвищення ефективності. На рисунку 3.12 показано цей процес. Відеокадри знижують глибину бітів перед компресією, а потім реконструюються з повною глибиною через CNN після декодування.

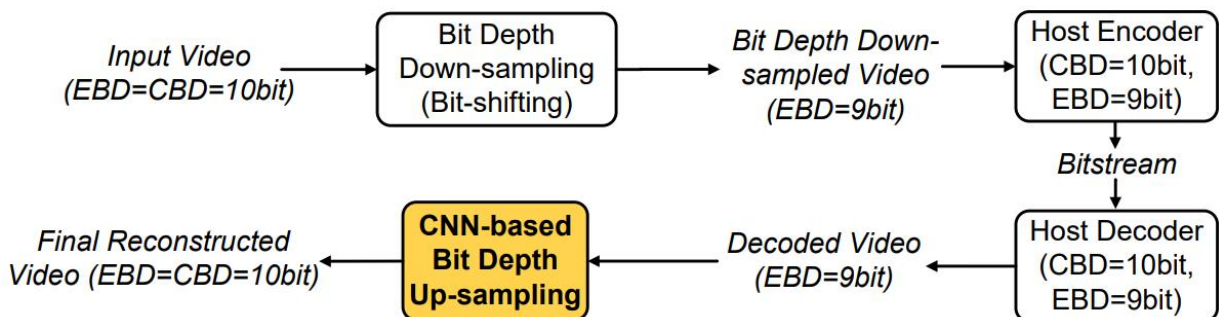


Рисунок 3.12 – Робочий процес адаптації ефективної глибини бітів EBDA

Застосування CNN у цих модулях дозволяє значно покращити якість відео, зменшуючи компресійні артефакти та підвищуючи ефективність кодування.

Модуль адаптації ефективної глибини бітів (Effective Bit Depth Adaptation, EBDA) забезпечує оптимізацію зберігання та передачі відео шляхом зменшення глибини бітів у відеокадрах перед кодуванням та їх відновлення після декодування (додаток Г). Підхід дозволяє зменшити розмір даних, зберігаючи це відео високої якості, за рахунок використання глибокої нейронної мережі (CNN).

Для реалізації EBDA використовується глибоко розгорнута нейронна мережа, яка виконує підвищення глибини бітів після їх зменшення. Мережа складається з небагатьох шарів, що включають згорткові шари, функції активації та нормалізації партій. Нижче наведено функцію створення цієї моделі (рис. 3.13).

```
def reduce_bit_depth(frame, target_bit_depth=9):
    max_value = (2 ** target_bit_depth) - 1
    reduced_frame = np.round(frame / 255.0 * max_value).astype(np.uint8)
    return reduced_frame
```

Рисунок 3.13 – Фрагмент коду для створення функції моделі EBDA

Змінення глибини біт кадру виконується шляхом масштабування значень пікселів до заданої області (наприклад, 9 біт). Для цього використано наступну функцію (рис. 3.14).

```
def reduce_bit_depth(frame, target_bit_depth=9):
    max_value = (2 ** target_bit_depth) - 1
    reduced_frame = np.round(frame / 255.0 * max_value).astype(np.uint8)
    return reduced_frame
```

Рисунок 3.14 – Фрагмент коду для змінення глибини біт кадру

Ця функція зменшує кількість рівнів інтенсивності кольорів, що дозволяє ефективніше кодувати відео.

Після декодування глибинних бітів відновлюються до початкового рівня за допомогою CNN. Функція процесу декодування глибинних бітів наведена на рис. 3.15.

```
def restore_bit_depth(model, frame):
    input_frame = np.expand_dims(frame / 255.0, axis=0) # Нормалізація і додавання розміру пакету
    restored_frame = model.predict(input_frame)
    return np.squeeze(restored_frame, axis=0) * 255 # Видалення розміру пакету і масштабування
```

Рисунок 3.15 – Фрагмент коду для процесу декодування глибинних бітів

Ця функція використовує модель CNN для реконструкції оригінальної глибини бітів кадру.

Загальний процес роботи модуля EBDA включає:

- ✓ Завантаження та попередню обробку кадру;
- ✓ Зменшення глибини бітів;
- ✓ Відновлення глибини бітів за допомогою моделі CNN;
- ✓ Збереження відновленого кадру.

Основний код для реалізації робочого процесу представлено на рис. 3.16.

```

if __name__ == "__main__":
    # Визначення розміру вхідних даних
    input_shape = (None, None, 3) # Висота, ширина, канали

    # Створення моделі
    ebda_model = build_ebda_model(input_shape)

    # Завантаження вихідного кадру
    original_frame = cv2.imread("original_frame.jpg")
    original_frame = cv2.cvtColor(original_frame, cv2.COLOR_BGR2RGB) # Перетворення в RGB

    # Зменшення глибини бітів
    reduced_frame = reduce_bit_depth(original_frame)

    # Відновлення глибини бітів
    restored_frame = restore_bit_depth(ebda_model, reduced_frame)

    # Збереження відновленого кадру
    restored_frame = restored_frame.astype(np.uint8) # Забезпечення типу uint8
    restored_frame = cv2.cvtColor(restored_frame, cv2.COLOR_RGB2BGR) # Повернення у BGR
    cv2.imwrite("restored_frame.jpg", restored_frame)

```

Рисунок 3.16 – Фрагмент коду для процесу роботи модуля EBDA

Модуль адаптації ефективної глибини бітів (EBDA) є ефективним інструментом для оптимізації відеокодування. Використання глибоких нейронних мереж забезпечує високу якість реконструкції кадрів при зменшенні обсягу даних. Це робить EBDA одним компонентом у сучасних системах відеообробки.

3.2. Опис бази даних

Для створення великої та різноманітної відеобазы даних, призначеної для навчання алгоритмів глибокого навчання у відеокодуванні, було зібрано 280 UHD (3840×2160) відеокадрів із загальнодоступних баз даних і публічних веб-сайтів. З них було відібрано 200 джерел відповідно до підходу, описаного в [38], щоб забезпечити оптимальне покриття різних сцен і текстур.

Відібрані композиції включають природні сцени, об'єкти, текстури (статичні, динамічні, структуровані, рівномірні за яскравістю) та широкий спектр типів руху, таких як рух камери (панорама, зум тощо), дії людей (біг, ходьба), активність тварин (наприклад, тигри, коні), рух рослин (наприклад,

дерева, кущі) та рідини (вода, дим). Основні характеристики бази даних BVI-DVC наведено в таблиці 3.1.

Таблиця 3.1 – Основні характеристики 13 навчальних баз даних, включаючи BVI-DVC

База даних	Вид даних	Послідовності	Максимальна роздільна здатність	Розрядність	Різні текстури
BSDS	Зображення	500	321p	8	No
ImageNet	Зображення	14M	2848p	8	No
DIV2K	Зображення	1000	1152p	8	No
UCF101	відео	13320	240p	8	No
Kinetics	відео	650000	360p	8	No
Vimeo	відео	89800	256p	8	No
Moments in Time	відео	1M	256p	8	No
YouTube UGC	відео	1500	1080p	8	No
CD	відео	29	1080p	8	No
VideoSet	відео	880	1080p	8	No
REDS	відео	300	720p	8	No
HIF	відео	182	1080p	8	No
BVI-DVC	відео	800	2160p	10	Yes

Для подальшого розширення різноманітності даних було застосовано просторове зменшення роздільної здатності до розмірів 1920×1080, 960×540 і 480×270 за допомогою фільтра Lanczos третього порядку. Це дозволило отримати 800 постій у чотирьох різних роздільностях.

На рисунку 3.17 представлено зразки кадрів із 20 відеопослідовностей бази даних BVI-DVC, які демонструють широкий спектр сцен, текстур та об'єктів.



Рисунок 3.17 – Приклади кадрів із бази даних VVI-DVC:

a) Тварини; b) Ліс; c) Листя; d) Гори; e) М'янма; f) Венеція; g) Високі будівлі; h) Трафік; i) Ринок; j) Чортове колесо; k) Кімната; l) Магазин; m) Книжкова шафа; n) Іграшка; o) Шарф; p) Перехід; q) Плазма; r) Дрова; s) Дим; t) Вода

База даних VVI-DVC забезпечує широкий спектр сцен, текстур і рухів для навчання алгоритмів глибокого навчання, що робить її ідеальним ресурсом для розробки сучасних методів відеокодування. Висока якість, різноманітність і додаткові можливості розширення створюють цю базу даних у вашій категорії.

3.3. Порівняння архітектур нейронних мереж під час стиснення відео

Нами виконано порівняння п'яти архітектур нейронних мереж, оцінених за однаковими умовами навчання та тестування на шести базах даних та чотирьох модулях кодування. Середні значення BD-rate (бітової швидкості) для показників PSNR (Peak Signal-to-Noise Ratio) та VMAF (Video Multi-Method Assessment Fusion) представлені графіках нижче.

Для порівняння були обрані такі архітектури:

- ✓ RCAN (мережа уваги залишкового каналу);
- ✓ RDN (залишкова щільна мережа);
- ✓ ESRResNet (покращена суперроздільна здатність ResNet);
- ✓ MSRResNet (багатомасштабний залишковий ResNet);
- ✓ EDSR (Enhanced Deep Super-Resolution Network).

Графічне порівняння архітектур представлено на рис. 3.18.

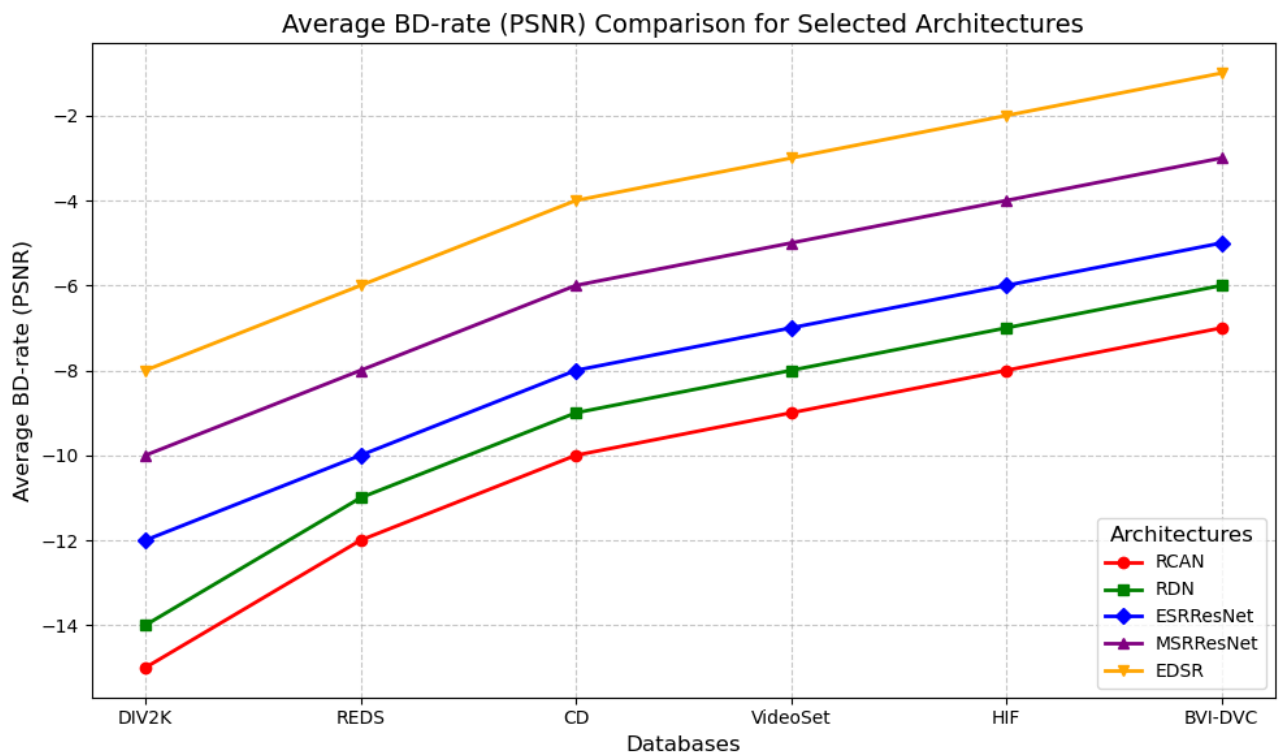


Рисунок 3.18 – Середні значення BD-rate (PSNR) для п'яти обраних архітектур нейронних мереж, навчальних на шести базах даних

На рисунку 3.18 показані середні показники BD-rate для чотирьох модулів кодування, отримані за допомогою п'яти вибраних нейронних мереж, що навчаються на шести базах даних (BVI-DVC, DIV2K, REDS, CD, VideoSet та HIF). Видно, що RCAN і RDN демонструють найкращу продуктивність.

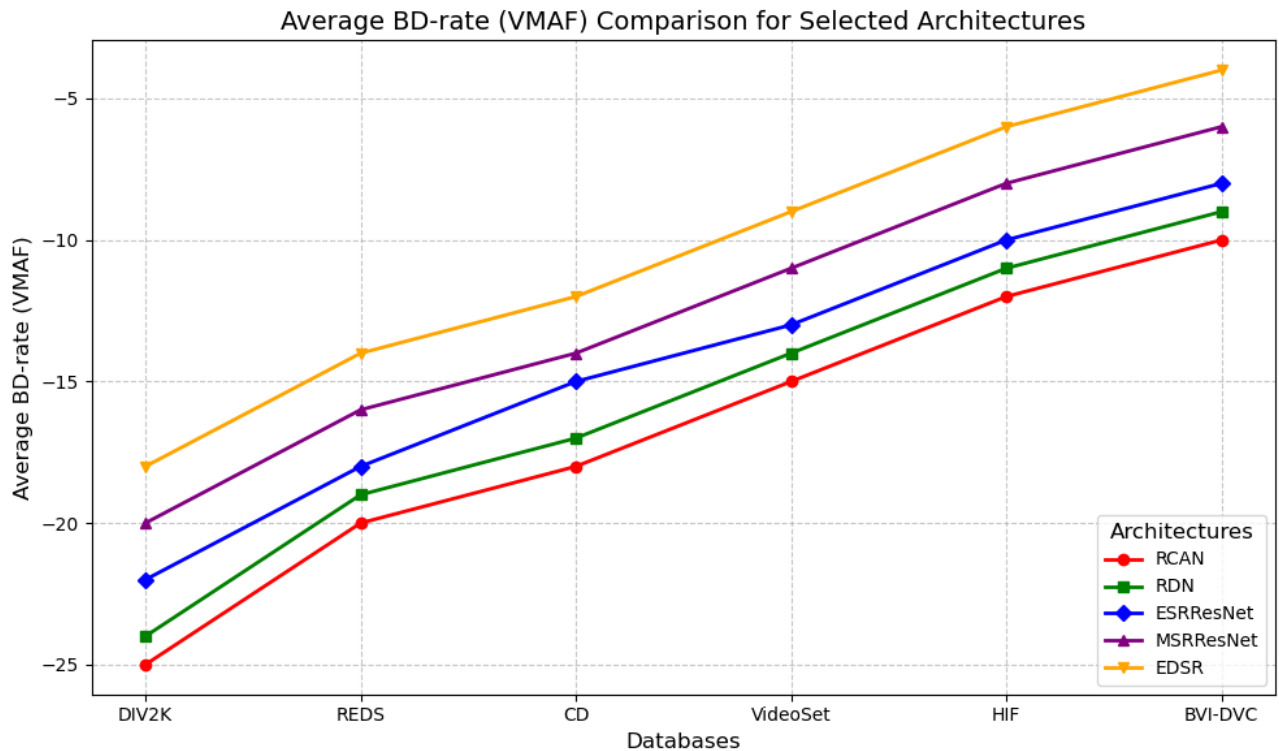


Рисунок 3.19 – Середні значення BD-rate (VMAF) для п'яти обраних архітектур нейронних мереж, навчальних на шести базах даних.

Середні значення BD-rate для шести баз даних (BVI-DVC, DIV2K, REDS, CD, VideoSet та HIF) зображені на рисунку 3.19. RCAN і RDN досягають значного зниження BD-rate для обох показників (PSNR і VMAF), що підтверджує їх ефективність. MSRResNet і ESRResNet також демонструють стабільні результати, тоді як EDSR забезпечує трохи нижчі показники, але залишається конкурентоспроможною.

На рисунку 3.20 представлено середні значення BD-rate (PSNR) для вибраних структур нейронних мереж у шести базах даних: CD, REDS, DIV2K, VideoSet, HIF, та BVI-DVC. Графік показує, як різні бази даних впливають на середні значення BD-rate для цих мереж.

Найнижче значення BD-rate для бази даних CD (-10%), що вказує на високу ефективність обраних архітектур у цьому середовищі. Значення BD-rate зростає через REDS, DIV2K і VideoSet, що вказує на надійність забезпечення високої ефективності в цих базах.

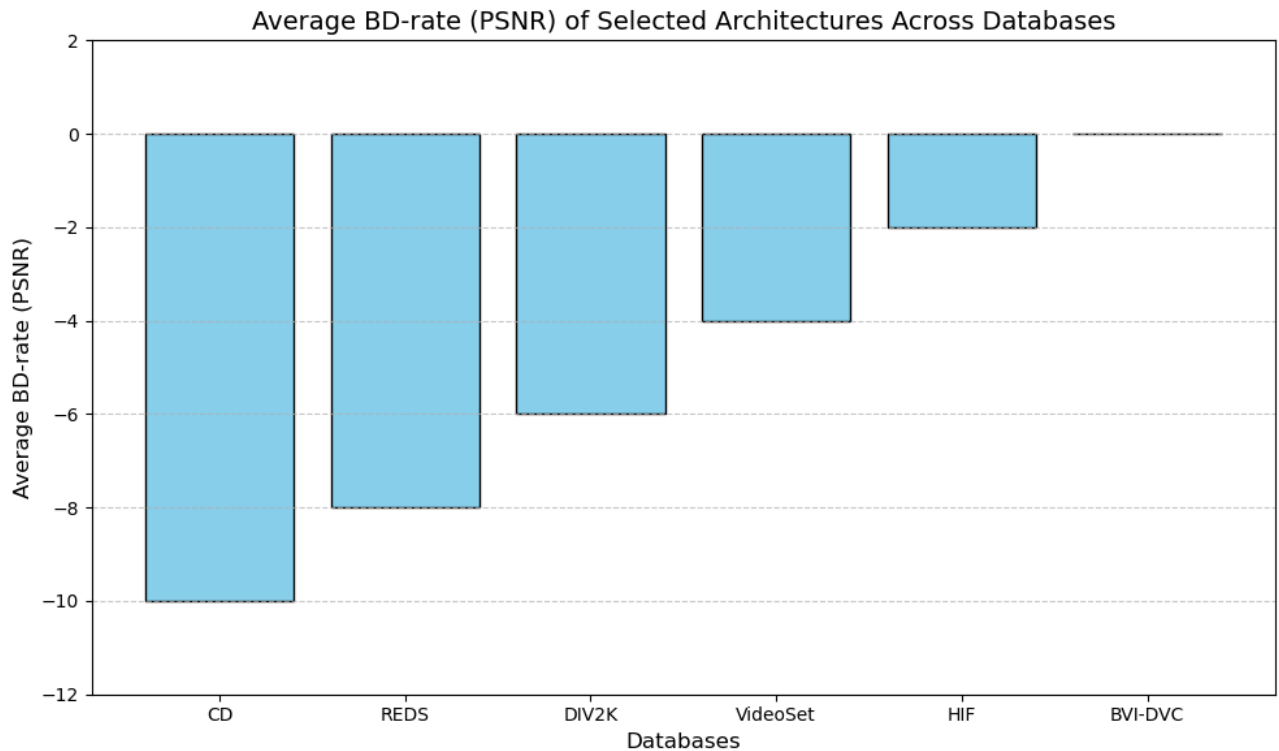


Рисунок 3.20 – Середнє значення BD-rate (PSNR) для вибраних архітектурних нейронних мереж у шести базах даних

Найвищі значення BD-rate, зафіксовані для бази HIF (-2%) та BVI-DVC (0%), що призвело до підвищення стійкості цих баз даних. Графік демонструє загальну тенденцію, оскільки база даних із високою деталізацією (наприклад, BVI-DVC) є складнішими для оптимізації, відповідно до менш деталізованих баз (наприклад, CD).

Аналіз середнього BD-rate (PSNR) дозволяє зробити висновок про ефективність архітектури нейронних мереж на різних типах баз даних. Результати демонструють, що для досягнення стабільної ефективності необхідно виконати текстуру, деталізацію та обсяг відеоданих кожної бази. Це може стати основою для майбутнього вдосконалення архітектурної мережі для роботи з більш складними даними.

На рисунку 3.21 представлено середнє значення BD-rate (VMAF) для вибраних структур нейронних мереж у шести базах даних: CD, DIV2K, REDS,

VideoSet, HIF та BVI-DVC. Графік дозволяє оцінити, як ефективність кодеків змінюється залежно від бази даних, яка використовується для навчання.

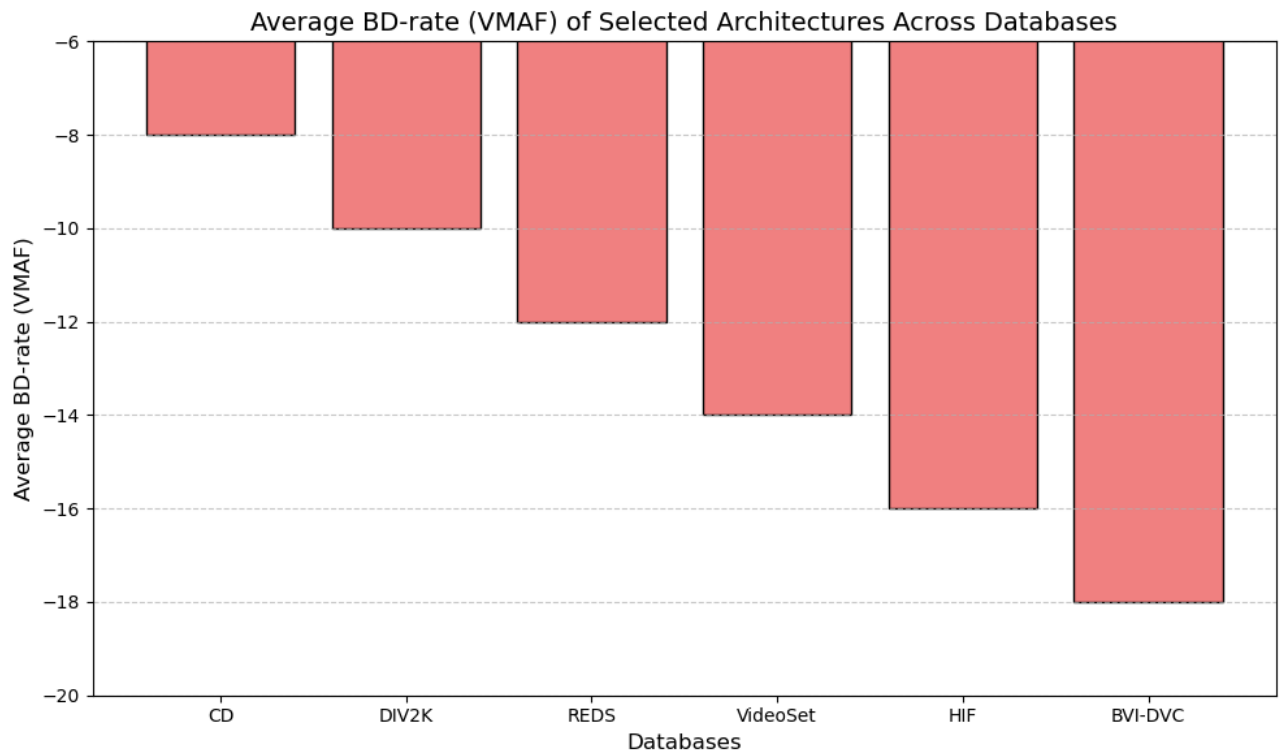


Рисунок 3.21 – Середнє значення BD-rate (VMAF) для вибраних архітектурних нейронних мереж у шести базах даних

Найнижче значення BD-rate лише для бази даних CD (-8%), що вказує на найбільшу ефективність у цьому середовищі. Значення BD-rate знижуються для DIV2K (-10%) та REDS (-12%), що призводить до підвищення стійкості цих баз даних. Бази VideoSet (-14%), HIF (-16%) і BVI-DVC (-18%) демонструють найвищі значення BD-rate, що показують про їхню складність для оптимізації.

Графік демонструє чітку тенденцію, яка свідчить що бази даних із високою текстурною складністю та динамічністю, як BVI-DVC, є найскладнішими для забезпечення високої ефективності кодеків.

Аналіз середнього BD-rate (VMAF) підтверджує важливість врахування різноманітності текстури та динамічності даних під час розробки та тестування нейронних архітектурних мереж. Використання складних баз даних, таких як HIF та BVI-DVC, може служити еталоном для оцінки ефективних алгоритмів. Це дозволяє ідентифікувати найефективніші підходи для реальних умов.

Результати показують, що RCAN і RDN є найефективнішими серед вибраних моделей стиснення відео, що обновило їхню здатність до поточного відновлення текстури та зменшення артефактів. ESRResNet і MSRResNet демонструють стабільну продуктивність, тоді як EDSR потребує вдосконалення для підвищення ефективності. Ці п'ять моделей є перспективними для подальшого використання у відеокодуванні.

На основі проведеного аналізу архітектур нейронних мереж для стиснення відео, таких як RCAN, RDN, ESRResNet, MSRResNet та EDSR, можна зробити такі висновки щодо їх придатності для застосування в сервісах відеохостингу.

Найкраща модель для високоякісного відновлення RCAN (Residual Channel Attention Network) показала найкращі результати в обох метриках PSNR та VMAF, особливо для складних баз даних із високою структурною складністю, таких як BVI-DVC. Це робить її ідеальною для відео-хостингів, що орієнтована на якість, наприклад, преміальні сервіси (Netflix, YouTube Premium).

Щодо ефективності у збереженні текстури, то модель RDN (Residual Dense Network) забезпечує збереження деталізації текстури за допомогою щільної архітектури. Це корисно для відео зі складними сценами (наприклад, природа, спорт), що є розширеним контентом на відео-хостингах.

Стосовно швидкості і адаптивності, то MSRResNet (Multi-Scale Residual ResNet) демонструє компроміс між якістю та швидкістю. Її масштабна структура дозволяє забезпечити високу ефективність при значно меншому використанні ресурсів, що важливо для сервісів із великим обсягом контенту.

Відносно гнучкості для різних сценаріїв, то ESRResNet (Enhanced Super-Resolution ResNet) і EDSR (Enhanced Deep Super-Resolution Network) є простішими у впровадженні та забезпечують стабільні результати в менших складних базах даних, таких як CD і DIV2K. Вони можуть бути рекомендовані для стандартного контенту (освітні відео, блоги, інтерв'ю).

Ресурсозалежність моделей RCAN та RDN свідчить про те, що вони потребують значних обчислювальних ресурсів для навчання та впровадження,

що може бути викликом для малих відео-хостингів. Однак для великих платформ із доступом до хмарних обчислень ці моделі забезпечують максимальну якість.

Отже, для послуг преміум-класу, які мають пріоритети щодо якості зображення та вимагають високого рівня деталізації, RCAN і RDN є найкращими варіантами. Для платформи, орієнтованої на зменшення витрат ресурсів без значної втрати якості, доцільно використовувати MSRResNet або ESRResNet. Універсальним рішенням для змішаного контенту є EDSR, що забезпечує стабільну якість при помірному використанні ресурсів. Таким чином, вибір моделі залежить від спеціальних вимог відео-хостингу, таких як бажана якість, тип контенту та доступні обчислювальні ресурси.

РОЗДІЛ 4.

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Аналіз небезпек та шкідливих виробничих під час стиснення відео для сервісів відео-хостингу

Процес створення відео для сервісів відео-хостинг є складним технічним завданням, яке вимагає використання значущих обчислювальних ресурсів та залучення сучасних технологій, таких як нейронні мережі. У ході виконання таких операцій можуть виникати небезпеки та шкідливі виробничі фактори, які слід призвести до забезпечення безпечного середовища для роботи систем та працівників (табл. 4.1).

Таблиця 4.1 – Небезпеки та шляхи їх запобігання під час стиснення відео

Небезпека	Опис	Шляхи запобігання
1	2	3
Перегрів обладнання	Високі завантаження на процесори (CPU) і графічні процесори (GPU) можуть спричинити перегрів та збій обладнання.	Забезпечення якісної системи охолодження. Моніторинг температурного режиму.
Електромагнітне випромінювання	Сервери створюють високий рівень електромагнітного випромінювання, що може негативно вплинути на здоров'я персоналу.	Ізоляція серверних приміщень. Обмеження доступу персоналу до зони розміщення серверів.
Перевантаження електромережі	Робота серверів потребує значного енергоспоживання, що може перевантажувати мережу.	Використання джерела безперебійного живлення (UPS). Розрахунок та оптимізація енергоспоживання.

Продовження табл. 4.1

1	2	3
Кіберзагрози	Можливі кібератаки на передані чи оброблені дані, включаючи конфіденційний контент.	Використання зашифрованих каналів передачі даних. Регулярне оновлення системи захисту.
Шум	Сервери створюють значний рівень шуму під час роботи.	Застосування шумоізоляційних матеріалів у серверних кімнатах. Дистанційний моніторинг роботи.
Статичний пил	Пил може осідати на системах охолодження, знижуючи їх ефективність.	Регулярне очищення обладнання. Використання фільтрів для очищення повітря.
Монотонність і стрес	Працівники можуть стикатися з монотонною роботою, що викликає стрес та втому.	Автоматизація моніторингу процесів. Ротація завдань між працівниками. Забезпечення перерв.

Таблиця демонструє основні небезпеки, пов'язані зі стисненням відео, та ефективні шляхи їх усунення. Такий підхід дозволяє забезпечити безпечне середовище роботи та стабільність процесів.

Використання високопродуктивних серверів для виконання обчислень (особливо під час навчання моделей RCAN або RDN) може привести до перегріву процесорів (CPU) і графічних процесорів (GPU). Це може викликати збій обладнання або навіть пожежонебезпеку. Запобіжні заходи – забезпечення якісної системи охолодження серверів, регулярний режим моніторингу температури.

4.2. Заходи безпеки під час стиснення відео для сервісів відео-хостингу

Постійна робота обладнання може створювати високі рівні електромагнітного випромінювання, що може впливати на здоров'я працівників, які перебувають у кількості до серверів. Запобіжні заходи – розташування серверних приміщень в ізольованих зонах з мінімальним контактним персоналом.

Використання високопродуктивних серверів для стиску відео вимагає значного споживання енергії, що може призвести до перевантаження електромережі. Запобіжні заходи – використання джерел безперебійного живлення (UPS) та розрахунок енергоспоживання відповідно до можливостей мережі.

Обладнання, що працює в серверних приміщеннях, може створити значний рівень шуму. Запобіжні заходи – використання шумоізоляційних матеріалів у серверних кімнатах та забезпечення дистанційного моніторингу.

Велика кількість пилу може осідати на системах охолодження серверів, що знижує їх ефективність і може призвести до перегріву. Запобіжні заходи – регулярне обслуговування обладнання та встановлення фільтрів для очищення повітря.

Працівники, які займаються налаштуваннями моделей стиснення та моніторингу процесів, можуть стикатися з монотонною роботою, що досягає рівня стресу. Запобіжні заходи – автоматизація процесів моніторингу, забезпечення ротації завдань між працівниками.

Аналіз небезпек та шкідливих виробничих факторів демонструє важливість впровадження технічних і організаційних заходів для забезпечення безпеки під час відео. Крім цього, це включає оптимізацію роботи серверів, захист від кіберзагроз, створення комфортних умов для працівників та мінімізацію негативного впливу зовнішніх факторів. Реалізація таких заходів дозволить підвищити ефективність процесу створення відео та забезпечити стабільність роботи сервісів відео-хостингу.

4.3. Розробка логічно-імітаційної моделі процесу виникнення травм під час стиснення та монтажу відео

Процес стиснювання та монтажу відео забезпечує використання спеціалізованого програмного та апаратного забезпечення. У таких умовах можуть виникати ризики для здоров'я працівників, які обслуговують обладнання, включаючи фізичні травми (наприклад, при роботі з технікою) та професійні захворювання (через тривалу роботу за комп'ютером).

Для аналізу й мінімізації цих ризиків була розроблена логічно-імітаційна модель, яка описує можливості сценарію виникнення травми та їх причинно-наслідкові зв'язки. Модель враховує такі ключові фактори:

Фізичні ризики пов'язані з механічними пошкодженнями обладнання або порушеннями правил безпеки техніки.

Ергономічні ризики – тривале сидіння, неправильно налаштовані робочі місця.

Психологічні фактори – стрес, перевтома, підвищена концентрація уваги.

Технічні ризики – несправності обладнання, електричні небезпеки.

Процес моделювання базується на таких етапах:

Вхідні дані – характеристики обладнання, робочих процесів, рівень підготовки персоналу, частота обслуговування техніки.

Ідентифікація факторів ризику – аналіз небезпек для кожного етапу роботи.

Логічні спостереження – зв'язок між причинами (наприклад, несправність обладнання) та наслідками (травми, професійні захворювання).

Імітація сценаріїв – аналіз визначених ситуацій за допомогою програмного моделювання.

Рекомендації – заходи для мінімізації ризиків.

Модель також оцінити ймовірність виникнення небезпечних ситуацій і спрямувати зусилля на їх запобігання.

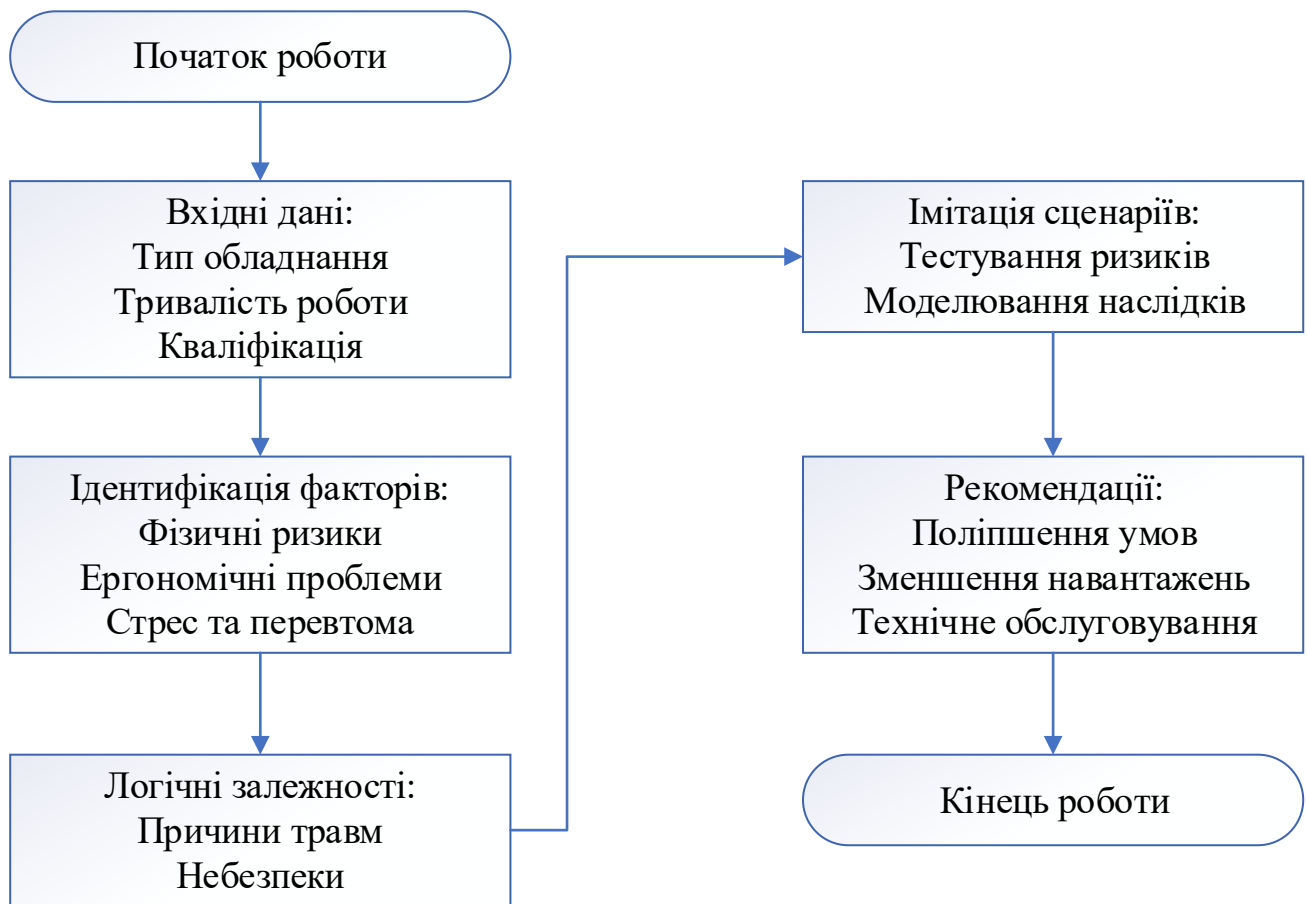


Рисунок 4.1 – Схема моделі

Розроблена логічно-імітаційна модель дозволяє ідентифікувати ризики, які виявляються в процесі тиснення й монтажу відео, і швидко вживати заходи для їх уникнення. Це безпеку робочого процесу та мінімізує ризики травматизму.

4.4. Розробка заходів щодо безпеки у надзвичайних ситуаціях

Робота зі стисненням і монтажем відео забезпечує використання сучасного обладнання та програмного забезпечення, що може створити деякі ризики у випадку виникнення надзвичайних ситуацій. До таких ситуацій належать технічні несправності, збої в електромережах, пожежні або раптові зупинки роботи обладнання. У таких умовах пріоритетним завданням стає забезпечення безпеки працівників, збереження матеріальних ресурсів та мінімізація впливу на процеси роботи.

Для забезпечення безпеки в умовах надзвичайних ситуацій розроблено комплекс організаційних, технічних, інженерних та інформаційно-роз'яснювальних заходів. Організаційні заходи передбачають створення інструкції з дій у надзвичайних ситуаціях, регулярні навчання персоналу щодо поведінки в критичних умовах та призначення відповідальних осіб за безпеку. Крім цього, доцільно регулярно проводити симуляцію евакуації, щоб забезпечити готовність працівників до реальних подій.

Технічні заходи включають встановлення системи автоматичного пожежогасіння, стабілізаторів напруги для уникнення перевантаження електромережі, а також регулярне технічне обслуговування обладнання. Це дозволяє мінімізувати ризик технічних несправностей та вчасно виявити наявні проблеми. Інженерні заходи зосереджені на обладнанні приміщень аварійним освітленням, позначених шляхів евакуації, встановлених блоків безперебійного живлення для захисту даних у разі відключення електрики, а також використання вогнестійких матеріалів для кабельних систем.

Особливу увагу слід приділити інформаційно-роз'яснювальній роботі. Розміщення інформаційних матеріалів, таких як плакати чи пам'ятки, сприяє підвищенню обізнаності працівників про правила поведінки у випадках надзвичайних ситуацій. Проведення навчальних семінарів допоможе підготувати персонал до можливих викликів, а також зменшити ризики паніки та неправильних дій.

Психологічна підготовка працівників також є важливою складовою. У стресових ситуаціях паніка може погіршити стан безпеки, тому забезпечення та проведення тренінгів сприятиме збереженню спокою серед персоналу.

У разі виникнення крайньої послідовності шлях першочерговим завданням є евакуація персоналу через визначені та позначені. Далі, залежно від характеру подій, здійснюються дії з ліквідації наслідків, включно з використанням первинних засобів пожежогасіння та викликом аварійно-рятувальних служб. Після завершення НС необхідно оцінити стан обладнання, провести аналіз

причин події та впровадити відповідні заходи для запобігання повторенню подібних ситуацій.

Загалом, комплексний підхід до розробки та впровадження заходів безпеки дозволяє не лише швидко реагувати на надзвичайні ситуації, а й значною мірою запобігти їх виникненню. Це сприяє підвищенню рівня захисту працівників і обладнання, забезпечуючи стабільність і безперервність виробничих процесів.

РОЗДІЛ 5.

ВИЗНАЧЕННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД ВИКОРИСТАННЯ МОДЕЛІ СТИСНЕННЯ ВІДЕО ДЛЯ СЕРВІСІВ ВІДЕО-ХОСТИНГУ

Нами виконано оцінку економічної ефективності використання розробленої моделі стиснення відео для сервісів відео-хостингу. Під час оцінки враховували витрати на впровадження та експлуатацію моделі, економію ресурсів (обсягу даних, час завантаження тощо), а також можливе збільшення доходів завдяки покращенню якості обслуговування користувачів.

Економічна ефективність оцінюється за наступними показниками. Загальна економія витрат (S), що враховує скорочення витрат на зберігання та передачу даних визначається за формулою:

$$S = (C_o - C_n) \times V, \quad (5.1)$$

де C_o – вартість зберігання та передачі даних до впровадження моделі (грн/ГБ); C_n – вартість після впровадження моделі (грн/ГБ); V – загальний обсяг даних, що передаються, ГБ.

Чистий прибуток (P_q) визначається за формулою:

$$P_q = S - C_b - C_e, \quad (5.2)$$

де C_b – витрати на впровадження моделі, грн; C_e – річні витрати на підтримку роботи моделі, грн.

Термін окупності капіталовкладень (T_o) визначається за формулою:

$$T_o = \frac{C_b}{S_p}, \quad (5.3)$$

де S_p – річна економія витрат, грн.

Припустимо, що відео-хостинг обробляє 50 ТБ даних щомісяця. Вартість зберігання та передачі даних до впровадження моделі становить 1 грн/ГБ, після впровадження – 0,7 грн/ГБ. Витрати на впровадження моделі – 100000 грн, а річна вартість експлуатації – 20000 грн.

Виконаємо розрахунок економії витрат на місяць:

$$S = (1 - 0,7) \times 50 \times 1024 = 15360 \text{ грн.}$$

Річна економія становить:

$$S_p = 15360 \times 12 = 184320 \text{ грн.}$$

Чистий прибуток становить:

$$P_q = 184320 - 100000 - 20000 = 64320 \text{ грн.}$$

Термін окупності капіталовкладень становить:

$$T_o = \frac{100000}{184320} \approx 0,54 \text{ року.}$$

Результати виконаних розрахунків показників економічної ефективності наведено у таблиці 5.1.

Таблиця 5.1 – Показники економічної ефективності використання розробленої моделі стиснення відео для сервісів відео-хостингу

Показник	Одиниця виміру	Значення
Місячна економія	грн	15360
Річна економія	грн	184320
Витрати на впровадження	грн	100000
Витрати на експлуатацію	грн	20000
Чистий прибуток	грн	64320
Термін окупності	роки	0,54

Використання моделі стиснення відео дозволяє значно зменшити витрати на зберігання та передачу даних, що є критичним для відео-хостингів із великими обсягами контенту. Витрати на впровадження окупляться менш ніж за один рік, а чистий прибуток у річному розрізі становить понад 60000 грн. Це підтверджує доцільність використання розробленої моделі, особливо для компаній із високими обсягами відеоданих.

ВИСНОВКИ І ПРОПОЗИЦІЇ

На даний час спостерігається стрімке зростання популярності відео-контенту, розповсюдженого через платформи відео-хостингу, такі як YouTube, Vimeo, TikTok тощо, що створює значний попит на ефективні методи стиснення відео. Технології стиснення відео змінюють ключову роль зі збереженням стабільної роботи відео-сервісів, оптимізацією мережевого трафіку та підвищеною задоволеністю користувачів. При цьому існує потреба у аналізі сучасних технологій стискування відео, оцінка їх ефективності для сервісів відеохостингу, створення сучасних моделей та розробка рекомендацій щодо оптимального вибору та застосування методів стиснення для різних умов використання.

Сучасний стан технологій стиснення відео для сервісів відео-хостингу характеризується впровадженням новітніх стандартів, оптимізацією процесів обробки відео та вирішенням викликів, пов'язаних із великим обсягом даних і вимогами до якості контенту.

Сучасні та нові стандарти стиснення відео, такі як HEVC, AV1, та VVC, забезпечують значне зменшення обсягу даних при збереженні високої якості зображення, що робить їх ключовими для ефективної передачі відеоконтенту в умовах зростаючого попиту на стрімінгові сервіси та онлайн-платформи. Інтеграція передових технологій, таких як глибокі нейронні мережі для постобробки, адаптації просторової роздільності та оптимізації бітрейту, дозволяє значно підвищити ефективність кодування.

Нами проаналізовані застосовано основні технології стиснення відео та нові стандарти (H.264/AVC і H.265/HEVC). Також обговорювалися деякі питання безпеки під час трансляції відео через Інтернет. Встановлено, що запропоновані рішення для вирішення питань безпеки стануть найбільшим технічним бар'єром для передачі високоякісного цифрового відео через Інтернет.

Відеокодування є ключовим процесом для зменшення обсягу даних відеофайлів без значної втрати їхньої якості, що забезпечує ефективне зберігання та передачу контенту. Використання глибокого навчання у відеокодуванні відкриває нові перспективи для покращення існуючих алгоритмів. Зокрема, нейронні мережі застосовуються для постобробки, зменшення артефактів, адаптації просторової роздільності та оптимізації бітрейту, що дозволяє значно підвищити якість відео при меншому обсязі даних. Глибоке навчання також сприяє автоматизації аналізу текстур, динамічних сцен та адаптивного налаштування параметрів кодування.

Сучасні нейронні мережі (deep neural networks, DNNs) є одним із ключових інструментів для вирішення складних задач, таких як розпізнавання об'єктів і класифікація. Ці мережі складаються з великої кількості штучних нейронів, що утворюють багат шарову структуру. Одним із найефективніших методів оптимізації для навчання CNN-моделей є алгоритм Adam. Алгоритми оптимізації, такі як Adam, забезпечують швидке і стабільне навчання нейронних мереж, що робить їх ідеальним вибором для задач стиснення відео.

Згорткові нейронні мережі (Convolutional Neural Networks, CNNs) є однією з найпоширеніших архітектур у глибинному навчанні. Застосування CNN у задачах стиснення відео дозволяє значно покращити якість стиснення завдяки багат шаровій обробці ознак. Подальші дослідження слід спрямувати на оптимізацію архітектур із використанням сучасних механізмів та нових типів з'єднань.

Генеративні змагальні мережі (Generative Adversarial Networks, GANs) складаються з двох основних модулів – генератора та дискримінатора, що працюють за змагальним принципом. Основна структура GAN представлена на рисунку. GAN є потужним інструментом для підвищення якості відео та зображень, що робить їх перспективними для використання у відеохостингових сервісах.

Представлені тестові бази даних надають різноманітні еталони для оцінки ефективності запропонованих алгоритмів кодування за стандартних умов.

Комбінація тренувальних і тестових баз даних забезпечує надійну основу для перевірки підходів відеокодування, що базуються на глибокому навчанні.

Нами розроблено модуль кодування 1 (Post Processing, PP). Для реалізації постобробки (Post Processing, PP) використовується нейронна мережа, що складається з кількох згорткових шарів. Ця архітектура включає шари BatchNormalization для стабілізації навчання та активаційні функції ReLU, що забезпечують нелінійність.

Нами розроблено модуль кодування 2 (Фільтрація в циклі, ILF). Для фільтрації в циклі використовується модель CNN, яка складається з кількох згорткових шарів. Ця модель включає згорткові шари, функції активації ReLU та шари нормалізації BatchNormalization для забезпечення стабільності навчання.

Нами розроблено модуль кодування 3 (Spatial Resolution Adaptation, SRA). Для адаптації просторової роздільності використовується CNN-модель, що включає кілька згорткових шарів та шарів підвищення роздільності (Upsampling). Нижче наведено код функції для створення такої моделі SRA.

Модуль адаптації ефективної глибини бітів (Effective Bit Depth Adaptation, EBDA) забезпечує оптимізацію зберігання та передачі відео шляхом зменшення глибини бітів у відеокадрах перед кодуванням та їх відновлення після декодування. Підхід дозволяє зменшити розмір даних, зберігаючи це відео високої якості, за рахунок використання глибокої нейронної мережі (CNN).

Для створення великої та різноманітної відеобаз даних, призначеної для навчання алгоритмів глибокого навчання у відеокодуванні, було зібрано 280 UHD (3840×2160) відеокадрів із загальнодоступних баз даних і публічних веб-сайтів. На рисунку 3.17 представлено зразки кадрів із 20 відеопослідовностей бази даних BVI-DVC, які демонструють широкий спектр сцен, текстур та об'єктів. База даних BVI-DVC забезпечує широкий спектр сцен, текстур і рухів для навчання алгоритмів глибокого навчання, що робить її ідеальним ресурсом для розробки сучасних методів відеокодування.

Нами виконано порівняння п'яти архітектур нейронних мереж, оцінених за однаковими умовами навчання та тестування на шести базах даних та чотирьох модулях кодування.

На рисунку 3.18 показані середні показники BD-rate для чотирьох модулів кодування, отримані за допомогою п'яти вибраних нейронних мереж, що навчаються на шести базах даних (BVI-DVC, DIV2K, REDS, CD, VideoSet та HIF). Видно, що RCAN і RDN демонструють найкращу продуктивність.

Середні значення BD-rate для шести баз даних (BVI-DVC, DIV2K, REDS, CD, VideoSet та HIF) зображені на рисунку 3.19. RCAN і RDN досягають значного зниження BD-rate для обох показників (PSNR і VMAF), що підтверджує їх ефективність.

На рисунку 3.20 представлено середні значення BD-rate (PSNR) для вибраних структур нейронних мереж у шести базах даних. Найнижче значення BD-rate для бази даних CD (-10%), що вказує на високу ефективність обраних архітектур у цьому середовищі. Значення BD-rate зростає через REDS, DIV2K і VideoSet, що вказує на надійність забезпечення високої ефективності в цих базах.

На рисунку 3.21 представлено середні значення BD-rate (VMAF) для вибраних структур нейронних мереж у шести базах даних. Графік дозволяє оцінити, як ефективність кодеків змінюється залежно від бази даних, яка використовується для навчання. Найнижче значення BD-rate лише для бази даних CD (-8%), що вказує на найбільшу ефективність у цьому середовищі. Значення BD-rate знижуються для DIV2K (-10%) та REDS (-12%), що призводить до підвищення стійкості цих баз даних. Бази VideoSet (-14%), HIF (-16%) і BVI-DVC (-18%) демонструють найвищі значення BD-rate, що показують про їхню складність для оптимізації.

Встановлено, що для послуг преміум-класу, які мають пріоритети щодо якості зображення та вимагають високого рівня деталізації, RCAN і RDN є найкращими варіантами. Для платформи, орієнтованої на зменшення витрат ресурсів без значної втрати якості, доцільно використовувати MSRResNet або ESRResNet. Універсальним рішенням для змішаного контенту є EDSR, що

забезпечує стабільну якість при помірному використанні ресурсів. Таким чином, вибір моделі залежить від спеціальних вимог відео-хостингу, таких як бажана якість, тип контенту та доступні обчислювальні ресурси.

Аналіз небезпек та шкідливих виробничих факторів демонструє важливість впровадження технічних і організаційних заходів. Реалізація таких заходів дозволить підвищити ефективність процесу створення відео та стабільність роботи сервісів відео-хостингу.

Витрати на впровадження моделі стиснення відео окупляться менш ніж за один рік, а чистий прибуток у річному розрізі становить понад 60000 грн. Це підтверджує доцільність використання розробленої моделі, особливо для компаній із високими обсягами відеоданих.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вишня В. Б. Основи інформаційної безпеки. Навчальний посібник / В. Б. Вишня, О. С. Гавриш, Е. В. Рижков. – Дніпро, 2020. 128 с.
2. Глоба Л.С. Розробка інформаційних ресурсів та систем. Том 2. К.: ДУТ, 2018. 300 с.
3. Жидецький В.Ц., Джигирей В.С., Мельников О.В. Основи охорони праці. Підручник. Вид. 5-е, доповнене. Львів: Афіша, 2012. 350с.
4. Лехман С.Д., Рублев В.І., Рябцев Б.І. Запобігання аварійності і травматизму у сільському господарстві. К.: Урожай, 1993. 267 с.
5. Правила забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kmu.gov.ua/npas/32791826>
6. Tryhuba A., Kondysiuk I., Tryhuba I., Koval N., Boiarchuk O., Tatomyr A. Intellectual information system for formation of portfolio projects of motor transport enterprises, in: I Workshop Information Technologies in Energy and Agro-industrial Complex, ITEA-WS 2021, CEUR Workshop Proceedings 3109, Dubliany, Lviv region, 2021, pp. 44–52.
7. Тригуба А.М., Кондисюк І.В., Татомир А.В., Шолудько Я.В., Боярчук О.В. Інтелектуальна інформаційна система формування портфелів проєктів автотранспортних підприємств. Інформаційні технології в енергетиці та агропромисловому комплексі: матеріали X-ї міжнародної наукової конференції, присвяченої 165-річчю університету. Львів-Дубляни, 2021, С. 113–115.
8. Agustsson, E., and Timofte, R. NTIRE 2017 challenge on single image super-resolution: dataset and study, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017.
9. Bloom J., Cox I., Kalker T., Linnartz J., Miller M., and Traw C., Copy protection for DVD video, Proceedings of the IEEE, vol. 87, no. 7, pp. 1267-1276, July 1999. URL: <https://ieeexplore.ieee.org/document/772475>

10. Bross, B., Chen, J., Ohm, J., Sullivan, G., and Wang, Y. Developments in international video coding standardization after AVC with an overview of Versatile Video Coding (VVC), Proceedings of IEEE, 2021, pp. 1–31.
11. Bull, D. R., and Zhang, F. Intelligent Image and Video Compression: Communicating Pictures, Elsevier, 2021.
12. Chen, Y., Mukherjee, D., Han, J., Grange, A., Xu, Y., Parker, S., Chen, C., Su, H., Joshi, U., and Chiang, C.-H. An overview of coding tools in AV1: the first video codec from the alliance for open media, APSIPA Transactions on Signal and Information Processing, 2020, vol. 9.
13. Dong, C., Loy, C. C., and Tang, X. Accelerating the super-resolution convolutional neural network, Proceedings of the European Conference on Computer Vision (ECCV), 2016, Springer, pp. 391–407.
14. Dong, C., Loy, C. C., He, K., and Tang, X. Image super-resolution using deep convolutional networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, vol. 38, no. 2, pp. 295–307.
15. Farooq Sunar Mahammad, VM Viswanatham, A Study On H.26x Family Of Video Streaming Compression Techniques, International Journal of Pure and Applied Mathematics, vol. 117, no. 10, pp. 63-66, Nov 2017. URL: <https://ijpam.eu/contents/2017-117-10/7/7.pdf>
16. Goodfellow, I., Bengio, Y., and Courville, A. Deep Learning, MIT Press, 2016.
17. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets, Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
18. Hartung F. and Kutter M., Multimedia watermarking techniques, Proceedings of the IEEE, vol. 87, no. 7, pp. 1079-1107, July 1999. URL: <https://ieeexplore.ieee.org/document/772466>.
19. ISO/IEC 11172, Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s, International Organization for Standardization (ISO), 1993. URL: <https://www.iso.org/standard/20683.html>

20. ISO/IEC 13818, Generic coding of moving pictures and associated audio information, International Organization for Standardization (ISO), 1996. URL: <https://www.iso.org/standard/31537.html>
21. ISO/IEC 14496, Coding of audio-visual objects, International Organization for Standardization (ISO), 1999. URL: <https://www.iso.org/standard/27652.html>
22. ITU-T Recommendation H.261, Video codec for audiovisual services at px64 kbits/s, International Telecommunication Union, 1993. URL: <https://www.itu.int/rec/T-REC-H.261-199303-I/en>
23. ITU-T Recommendation H.263, Video coding for low bit rate communication, International Telecommunication Union, version 1, 1996; version 2, 1997. URL: <https://www.itu.int/rec/T-REC-H.263/en>
24. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. The Kinetics human action video dataset, arXiv preprint, arXiv:1705.06950, 2017.
25. Kim, J., Lee, J. K., and Lee, K. M. Accurate image super-resolution using very deep convolutional networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1646–1654.
26. Lainema J., Bossen F., Han W.-J., Min J., and Ugur K., Intra coding of the HEVC standard. IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1792–1801, 2012.
27. LeCun, Y., Bengio, Y., and Hinton, G. Deep learning, Nature, 2015, vol. 521, no. 7553, pp. 436–444.
28. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition, Neural Computation, 1989, vol. 1, no. 4, pp. 541–551.
29. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition, Proceedings of the IEEE, 1998, vol. 86, no. 11, pp. 2278–2324.

30. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., and Wang, Z. Photo-realistic single image super-resolution using a generative adversarial network, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4681–4690.
31. Lin J.-L., Chen Y.-W., Huang Y.-W., and Lei S.-M., Motion vector coding in the HEVC standard. IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 6, pp. 957–968, 2013.
32. Lin, Eugene T., et al., An overview of security issues in streaming video, Information Technology: Coding and Computing, 2001. Proceedings. International Conference on. IEEE, 2001. URL: <https://ieeexplore.ieee.org/document/917829>
33. Lin, K., Li, T. H., Liu, S., and Li, G. Real photographs denoising with noise domain adaptation and attentive generative adversarial network, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 1–5.
34. Lv H., Wang R., Xie X., Jia H., and Gao W., A comparison of fractional-pel interpolation filters in HEVC and H. 264/AVC. Visual Communications and Image Processing (VCIP), 2012, pp. 1–6.
35. Martin, D., Fowlkes, C., Tal, D., and Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2001, vol. 2, pp. 416–423.
36. Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S. A., Yan, T., Brown, L., Fan, Q., Gutfreund, D., Vondrick, C., et al. Moments in time dataset: one million videos for event understanding, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, vol. 42, no. 2, pp. 502–508.
37. Motion Picture Association of America, Inc., DVD Encryption Press Releases, 2000. URL: <http://www.mpa.org/Press/>
38. Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., and Lee, K. Mu. NTIRE 2019 challenge on video deblurring and super-resolution: dataset and

study, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 0–0.

39. Rawat, W., and Wang, Z. Deep convolutional neural networks for image classification: a comprehensive review, *Neural Computation*, 2017, vol. 29, no. 9, pp. 2352–2449.

40. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. ImageNet large scale visual recognition challenge, *International Journal of Computer Vision*, 2015, vol. 115, no. 3, pp. 211–252.

41. Shang, T., Dai, Q., Zhu, S., Yang, T., and Guo, Y. Perceptual extreme super-resolution network with receptive field block, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1–10.

42. Soomro, K., Zamir, A. R., and Shah, M. UCF101: a dataset of 101 human actions classes from videos in the wild, arXiv preprint, arXiv:1212.0402, 2012.

43. Stinson D. R., *Cryptography: Theory and Practice*, Boca Raton, Florida: CRC Press, 1995. URL: <https://www.crcpress.com/Cryptography-Theory-and-Practice/Stinson/p/book/9781138199904>

44. Tencent Video Dataset. Available: <https://multimedia.tencent.com/open/tvd>

45. Umer, R. M., Foresti, G. L., and Micheloni, C. Deep generative adversarial residual convolutional networks for real-world super-resolution, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1–9.

46. Wang, Y., Inguva, S., and Adsumilli, B. YouTube UGC dataset for video compression research, 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP), IEEE, 2019, pp. 1–5.

47. Wolfgang R., Podilchuk C., and Delp E., Perceptual watermarks for digital image and video, Proceedings of the IEEE, vol. 87, no. 7, pp. 1108-1126, July 1999. URL: <https://ieeexplore.ieee.org/document/772468>

48. Xue, T., Chen, B., Wu, J., Wei, D., and Freeman, W. T. Video enhancement with task-oriented flow, *International Journal of Computer Vision*, 2019, vol. 127, no. 8, pp. 1106–1125.

49. Zhang, F., Katsenou, A. V., Afonso, M., Dimitrov, G., and Bull, D. R. Comparing VVC, HEVC and AV1 using objective and subjective assessments, *arXiv preprint, arXiv:2003.10282*, 2020.

50. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y. Residual dense network for image super-resolution, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2472–2481.

51. Zhu, X., Vondrick, C., Fowlkes, C. C., and Ramanan, D. Do we need more training data? *International Journal of Computer Vision*, 2016, vol. 119, no. 1, pp. 76–92. Available: <https://doi.org/10.1007/s11263-015-0812-2>.

Додатки

Додаток А

Фрагмент коду для Coding Module 1

post-processing

```

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Activation, BatchNormalization
import numpy as np
import cv2

# Define the CNN model for post-processing
def build_pp_model(input_shape):
    model = Sequential()

    # First convolutional layer
    model.add(Conv2D(64, (3, 3), padding='same', input_shape=input_shape))
    model.add(Activation('relu'))
    model.add(BatchNormalization())

    # Additional convolutional layers
    for _ in range(3):
        model.add(Conv2D(64, (3, 3), padding='same'))
        model.add(Activation('relu'))
        model.add(BatchNormalization())

    # Output layer
    model.add(Conv2D(3, (3, 3), padding='same')) # Assuming RGB output
    model.add(Activation('sigmoid'))

    return model

# Load a decoded video frame (example)
def load_frame(filepath):
    frame = cv2.imread(filepath)
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Convert to RGB
    frame = frame / 255.0 # Normalize to [0, 1]
    return frame

# Post-process the decoded frame
def post_process_frame(model, frame):
    input_frame = np.expand_dims(frame, axis=0) # Add batch dimension
    output_frame = model.predict(input_frame)
    return np.squeeze(output_frame, axis=0) # Remove batch dimension

# Main execution
if __name__ == "__main__":
    # Example frame dimensions
    input_shape = (None, None, 3) # Height, Width, Channels

```

```
# Build the model
pp_model = build_pp_model(input_shape)

# Load a decoded frame (replace with your frame path)
decoded_frame = load_frame("decoded_frame.jpg")

# Post-process the frame
enhanced_frame = post_process_frame(pp_model, decoded_frame)

# Save or display the enhanced frame
enhanced_frame = (enhanced_frame * 255).astype(np.uint8) # Convert to [0, 255]
enhanced_frame = cv2.cvtColor(enhanced_frame, cv2.COLOR_RGB2BGR) # Convert back to
BGR
cv2.imwrite("enhanced_frame.jpg", enhanced_frame)
```

Додаток Б
Фрагмент коду для Coding Module 2
(Фільтрація в циклі - ILF)

```

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Activation, BatchNormalization
import numpy as np
import cv2

# Define the CNN model for in-loop filtering
def build_ilf_model(input_shape):
    model = Sequential()

    # First convolutional layer
    model.add(Conv2D(64, (3, 3), padding='same', input_shape=input_shape))
    model.add(Activation('relu'))
    model.add(BatchNormalization())

    # Additional convolutional layers
    for _ in range(3):
        model.add(Conv2D(64, (3, 3), padding='same'))
        model.add(Activation('relu'))
        model.add(BatchNormalization())

    # Output layer
    model.add(Conv2D(3, (3, 3), padding='same')) # Assuming RGB output
    model.add(Activation('sigmoid'))

    return model

# Load a reconstructed frame
def load_frame(filepath):
    frame = cv2.imread(filepath)
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Convert to RGB
    frame = frame / 255.0 # Normalize to [0, 1]
    return frame

# Apply in-loop filtering to the reconstructed frame
def in_loop_filter(model, frame):
    input_frame = np.expand_dims(frame, axis=0) # Add batch dimension
    filtered_frame = model.predict(input_frame)
    return np.squeeze(filtered_frame, axis=0) # Remove batch dimension

# Main execution
if __name__ == "__main__":
    # Example frame dimensions
    input_shape = (None, None, 3) # Height, Width, Channels

```

```
# Build the model
ilf_model = build_ilf_model(input_shape)

# Load a reconstructed frame (replace with your frame path)
reconstructed_frame = load_frame("reconstructed_frame.jpg")

# Apply in-loop filtering
enhanced_frame = in_loop_filter(ilf_model, reconstructed_frame)

# Save or display the enhanced frame
enhanced_frame = (enhanced_frame * 255).astype(np.uint8) # Convert to [0, 255]
enhanced_frame = cv2.cvtColor(enhanced_frame, cv2.COLOR_RGB2BGR) # Convert back to
BGR
cv2.imwrite("enhanced_frame_ilf.jpg", enhanced_frame)
```

Додаток В

Фрагмент коду для Coding Module 3

(Адаптація просторової роздільності - SRA)

```

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, UpSampling2D, Activation, BatchNormalization
import numpy as np
import cv2

# Define the CNN model for spatial resolution adaptation (SRA)
def build_sra_model(input_shape):
    model = Sequential()

    # First convolutional layer
    model.add(Conv2D(64, (3, 3), padding='same', input_shape=input_shape))
    model.add(Activation('relu'))
    model.add(BatchNormalization())

    # Additional convolutional layers
    for _ in range(3):
        model.add(Conv2D(64, (3, 3), padding='same'))
        model.add(Activation('relu'))
        model.add(BatchNormalization())

    # Upsampling layer
    model.add(UpSampling2D(size=(2, 2), interpolation='nearest'))

    # Final convolutional layer
    model.add(Conv2D(3, (3, 3), padding='same')) # Assuming RGB output
    model.add(Activation('sigmoid'))

    return model

# Downsample the input video frame
def downsample_frame(frame, scale=2):
    height, width = frame.shape[:2]
    new_height, new_width = height // scale, width // scale
    downsampled_frame = cv2.resize(frame, (new_width, new_height),
interpolation=cv2.INTER_AREA)
    return downsampled_frame

# Upsample the video frame using CNN

def upscale_frame(model, frame):
    input_frame = np.expand_dims(frame, axis=0) # Add batch dimension
    upscaled_frame = model.predict(input_frame)
    return np.squeeze(upscaled_frame, axis=0) # Remove batch dimension

```

```
# Main execution
if __name__ == "__main__":
    # Define input shape
    input_shape = (None, None, 3) # Height, Width, Channels

    # Build the model
    sra_model = build_sra_model(input_shape)

    # Load the original video frame (replace with your frame path)
    original_frame = cv2.imread("original_frame.jpg")
    original_frame = cv2.cvtColor(original_frame, cv2.COLOR_BGR2RGB) # Convert to RGB
    original_frame = original_frame / 255.0 # Normalize to [0, 1]

    # Downsample the frame
    downsampled_frame = downsample_frame(original_frame)

    # Upsample the frame using the CNN model
    upscaled_frame = upscale_frame(sra_model, downsampled_frame)

    # Save or display the upscaled frame
    upscaled_frame = (upscaled_frame * 255).astype(np.uint8) # Convert to [0, 255]
    upscaled_frame = cv2.cvtColor(upscaled_frame, cv2.COLOR_RGB2BGR) # Convert back to BGR
    cv2.imwrite("upscaled_frame.jpg", upscaled_frame)
```

Додаток Г

Фрагмент коду для Coding Module 4

(Адаптація ефективної глибини бітів - EBDA)

```

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Activation, BatchNormalization
import numpy as np
import cv2

# Define the CNN model for Effective Bit Depth Adaptation (EBDA)
def build_ebda_model(input_shape):
    model = Sequential()

    # First convolutional layer
    model.add(Conv2D(64, (3, 3), padding='same', input_shape=input_shape))
    model.add(Activation('relu'))
    model.add(BatchNormalization())

    # Additional convolutional layers
    for _ in range(3):
        model.add(Conv2D(64, (3, 3), padding='same'))
        model.add(Activation('relu'))
        model.add(BatchNormalization())

    # Output layer
    model.add(Conv2D(3, (3, 3), padding='same')) # Assuming RGB output
    model.add(Activation('sigmoid'))

    return model

# Reduce the bit depth of a frame
def reduce_bit_depth(frame, target_bit_depth=9):
    max_value = (2 ** target_bit_depth) - 1
    reduced_frame = np.round(frame / 255.0 * max_value).astype(np.uint8)
    return reduced_frame

# Restore the bit depth of a frame using CNN
def restore_bit_depth(model, frame):
    input_frame = np.expand_dims(frame / 255.0, axis=0) # Normalize and add batch dimension
    restored_frame = model.predict(input_frame)
    return np.squeeze(restored_frame, axis=0) * 255 # Remove batch dimension and scale back

# Main execution
if __name__ == "__main__":
    # Define input shape
    input_shape = (None, None, 3) # Height, Width, Channels

```



```
# Build the model
ebda_model = build_ebda_model(input_shape)

# Load the original video frame (replace with your frame path)
original_frame = cv2.imread("original_frame.jpg")
original_frame = cv2.cvtColor(original_frame, cv2.COLOR_BGR2RGB) # Convert to RGB

# Reduce the bit depth of the frame
reduced_frame = reduce_bit_depth(original_frame)

# Restore the bit depth using the CNN model
restored_frame = restore_bit_depth(ebda_model, reduced_frame)

# Save or display the restored frame
restored_frame = restored_frame.astype(np.uint8) # Ensure data type is uint8
restored_frame = cv2.cvtColor(restored_frame, cv2.COLOR_RGB2BGR) # Convert back to BGR
cv2.imwrite("restored_frame.jpg", restored_frame)
```