Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Нейромережевий підхід до розробки метрики оцінювання точності синхронізації мовлення та руху губ у віртуальних аватарах
(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-23-1
Мирошник Юрій
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)
Керівник проф. Наталія Рябова
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Мирошнику Юрію Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Нейромережевий підхід до розробки метрики оцінювання точності синхронізації мовлення та руху губ у віртуальних аватарах

затверджена наказом університету від 21 квітня 20 25 р. № 295Ст

2. Термін подання студентом роботи до екзаменаційної комісії 6 червня 20 25 р.

3. Вихідні дані до роботи Науково-технічні публікації, документація мови Python, модель DINet, модель Wav2lip, модель SyncNet, набір даних HDTF, набір даних Hallo3, документація до бібліотек Pytorch, Transformers, Diffusions

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі та існуючих рішень
2) Теоретичні дослідження
3) Експериментальні дослідження

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	21.04.2025	виконано
2	Дослідження предметної галузі	24.04.2025	виконано
3	Постановка задачі	27.04.2025	виконано
4	Теоретичні дослідження існуючих методів	01.05.2025	виконано
	вирішення задачі		
5	Розробка архітектури моделі	05.05.2025	виконано
6	Навчання моделі	09.05.2025	виконано
7	Написання пояснювальної записки	18.05.2025	виконано
8	Нормоконтроль	28.05.2025	виконано
9	Перевірка на академічний плагіат	29.05.2025	виконано
10	Підготовка презентації та доповіді	02.06.2025	виконано
11	Захист перед ЕК	06.06.2025	

Дата видачі завдання 21 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Наталія Рябова
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 95 с., 26 рис., 1 табл., 1 дод., 26 джерел.

АУДІО-ВІДЕО СИНХРОНІЗАЦІЯ, ВІРТУАЛЬНІ АВАТАРИ, ГЕНЕРАТИВНО-ЗМАГАЛЬНІ МЕРЕЖІ, ГЛИБОКЕ НАВЧАННЯ, ДИФУЗІЙНІ МОДЕЛІ, СИНХРОНІЗАЦІЯ РУХУ ГУБ.

Об'єкт дослідження – системи автоматичного генерування синхронізованих візуалізацій мовлення для віртуальних аватарів на основі глибоких нейронних мереж.

Предмет дослідження – методи підвищення точності синхронізації мовлення та рухів губ у відео з віртуальними аватарами за допомогою удосконалених метрик аудіо-відео синхронізації.

Мета роботи – розробка та вдосконалення метрики синхронізації на основі моделей аудіо-відео синхронізації для підвищення точності та стабільності роботи систем генерації аудіо-візуальної синхронізації губ.

Методи дослідження – теоретичні (аналіз наукової літератури, порівняння існуючих підходів) та практичні (розробка модифікацій моделі оцінки синхронізації та їх експериментальна перевірка).

Наукова новизна полягає у розробці нової архітектури моделі оцінки аудіо-відео синхронізації AVAlignNet, яка демонструє покращені показники точності та стабільності порівняно з існуючими аналогами. Досліджено вплив різних архітектурних рішень та гіперпараметрів на ефективність навчання AVAlignNet.

Результатом дослідження є розроблена та експериментально перевірена модель оцінки аудіо-відео синхронізації, яка покращує існуючі підходи до оцінки аудіо-відео синхронізації. Проведено детальний аналіз процесу підготовки даних та факторів, що впливають на узагальнювальну здатність моделей на різнорідних датасетах.

ABSTRACT

Master’s thesis contains: 95 pp., 26 fig., 1 tabl., 1 ann., 26 references.

AUDIO-VIDEO SYNCHRONIZATION, DEEP LEARNING, DIFFUSION MODELS, GENERATIVE ADVERSARIAL NETWORKS, LIP-SYNC, VIRTUAL AVATARS.

The object of research – systems for automatic generation of synchronized speech visualizations for virtual avatars based on deep neural networks.

The subject of research – methods for improving the accuracy of speech and lip movement synchronization in videos with virtual avatars using advanced audio-video synchronization metrics.

The purpose of the work – development and improvement of a synchronization metric based on audio-video synchronization models to enhance the accuracy and stability of lip-sync generation systems.

Research methods – theoretical (analysis of scientific literature, comparison of existing approaches) and practical (development of modifications to the synchronization assessment model and their experimental verification).

Scientific novelty lies in the development of a new architecture for the AVAlignNet audio-video synchronization assessment model, which demonstrates improved accuracy and stability compared to existing analogues. The influence of various architectural solutions and hyperparameters on the training effectiveness of AVAlignNet was investigated.

The result of the research is a developed and experimentally verified audio-video synchronization assessment model that improves existing approaches to evaluating audio-video synchronization. A detailed analysis of the data preparation process and factors affecting the generalization ability of models on heterogeneous datasets was conducted.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AVAlignNet – Audio-Video Alignment Network – розроблена в роботі модель оцінки аудіо-відео синхронізації;

DINet – Deformation Inpainting Network – мережа домальовування на основі деформації;

GAN – Generative Adversarial Network – генеративно-змагальна мережа;

HDTF – High-Definition Talking Face – аудіо-візуальний датасет високої чіткості;

LPIPS – Learned Perceptual Image Patch Similarity – навчена перцептивна подібність фрагментів зображень;

LSE-C – Lip Sync Error-Confidence – помилка синхронізації губ-впевненість;

LSE-D – Lip Sync Error-Distance – помилка синхронізації губ-відстань;

MFCC – Mel-Frequency Cepstral Coefficients – мел-частотні кепстральні коефіцієнти;

PSNR – Peak Signal-to-Noise Ratio – пікове відношення сигнал/шум;

SSIM – Structural Similarity Index – індекс структурної подібності;

SyncNet – Synchronization Network – мережа для оцінки синхронності аудіо та відео;

VAE – Variational Autoencoder – варіаційний автоенкодер;

Wav2Lip – Waveform to Lip – модель генерації руху губ на основі аудіосигналу.

ВСТУП

Одним із ключових напрямків розвитку сучасних цифрових технологій є створення реалістичних віртуальних аватарів зі здатністю комунікації. У цьому контексті особливої важливості набуває проблема синхронізації руху губ аватарів з необхідним аудіо. Віртуальні аватари широко використовуються в різноманітних сферах: від ігрової індустрії до телемедицини, освітніх платформ та систем обслуговування клієнтів. Якість взаємодії користувачів із такими системами значною мірою залежить від реалістичності відтворення мовлення.

Перші спроби створення систем аудіо-візуальної синхронізації губ базувалися на простих алгоритмічних підходах, які встановлювали відповідність між фонемами та певними позиціями губ. Результати таких систем часто виглядали неприродньо та мали обмежену виразність.

Незважаючи на значний прогрес, існуючі системи синхронізації руху губ із мовленням мають ряд обмежень: нестабільність роботи при різних вхідних даних, недостатня точність синхронізації для складних аудіо, проблема генералізації до нових персон. Ключовою проблемою залишається розробка надійних метрик оцінки якості синхронізації, які можна ефективно використовувати в процесі навчання моделей.

У даній роботі проведено дослідження сучасних підходів до генерації синхронізованих з аудіо рухів губ, вивчено принципи роботи моделей оцінки аудіо-візуальної синхронізації. Запропоновано та реалізовано модифіковану архітектуру моделі оцінки синхронізації з метою підвищення її точності та стабільності. Особливу увагу приділено експериментальному дослідженню впливу різних гіперпараметрів та архітектурних рішень на ефективність, а також інтеграції вдосконаленої моделі у процес навчання генеративної моделі для оцінки впливу на кінцеву якість синхронізації та візуальну реалістичність згенерованих відео.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної галузі

Автоматичне генерування синхронізованих візуалізацій мовлення для віртуальних аватарів полягає у створенні реалістичних відео, де рухи губ аватара точно відповідають поданому мовленню. Ця задача відома як генерація «говорячих облич» (audio-driven facial animation) – синтез обличчя, що говорить, на основі аудіосигналу. Модель отримує на вході звук мовлення (та, за потреби, зображення обличчя або певну позу) і продукує послідовність зображень обличчя, на яких губи рухаються відповідно до аудіо. Сучасні нейромережеві підходи до цієї задачі дозволяють отримувати відео, де синхронізація мовлення з рухом губ наближається до реальних відеозаписів.

Ступінь складності та візуального представлення аватарів варіюється від простих двовимірних (2D) чи тривимірних (3D) стилізованих моделей до високодеталізованих фотореалістичних репрезентацій. Ключовими характеристиками аватарів є:

– зовнішній вигляд: стиль, рівень деталізації, можливість кастомізації;

– рух та навігація: здатність переміщатися у віртуальному просторі;

– інтерактивність: можливість взаємодії з віртуальними об’єктами та іншими аватарами;

– комунікаційні можливості: здатність до відтворення мовлення та супутніх невербальних сигналів (жести, міміка, рухи губ). Саме аспект відтворення рухів губ, синхронізованих з аудіорядом, є важливим елементом функціональності сучасних аватарів [1].

Історично аватари еволюціонували від текстових репрезентацій у MUD (Multi-User Dungeons) та простих графічних зображень у ранніх онлайн-іграх до складних 3D-моделей у сучасних віртуальних світах та системах віртуальної реальності (Virtual Reality, VR) та доповненої

реальності (Augmented Reality, AR). Цей розвиток був зумовлений прогресом у комп’ютерній графіці, мережевих технологіях та алгоритмах штучного інтелекту.

На сучасному етапі віртуальні аватари є невід’ємною складовою широкого спектра цифрових платформ та прикладних галузей. В ігровій індустрії вони виступають центральними елементами, представляючи гравців та неігрових персонажів (Non-Playable Characters, NPC). У соціальних віртуальних світах та метавсесвіті, таких як VRChat чи Meta Horizon Worlds, аватари слугують основою для соціальної взаємодії, самовираження та формування віртуальних спільнот, дозволяючи користувачам взаємодіяти у спільних тривимірних просторах.

Аватари також знаходять широке застосування у сферах комунікації та спільної роботи. Платформи на кшталт Microsoft Mesh використовують їх для створення ефекту присутності під час віддалених зустрічей, тренінгів та колективної роботи над проектами, що сприяє відтворенню невербальних сигналів та покращенню якості комунікації. У віртуальній та доповненій реальності аватари є ключовими для забезпечення занурення та інтерактивності, що є важливим для тренажерів (медичних, промислових), освітніх програм та терапевтичних застосунків, наприклад, для лікування фобій.

Крім того, сфера освіти та тренінгів активно використовує аватари у ролі віртуальних викладачів, тренерів чи симуляційних персонажів для вивчення мов або відпрацювання навичок. В охороні здоров'я вони застосовуються для телемедичних консультацій, віртуальної психотерапії, навчання пацієнтів та медичного персоналу, а також як віртуальні компаньйони [2]. Електронна комерція та маркетинг використовують аватари для віртуальних примірочних, демонстрації товарів та у ролі віртуальних консультантів чи представників бренду.

Таким чином, предметна галузь охоплює теоретичні основи, методи створення, анімації, керування та застосування віртуальних аватарів у

різноманітних цифрових системах. Технології, що забезпечують реалістичне відтворення поведінки та комунікації аватарів, включно з синхронізацією рухів губ з мовленням, є складовою цієї галузі.

1.2 Актуальність дослідження

Тематика синхронного «озвучення» віртуальних аватарів є надзвичайно актуальною в умовах стрімкого розвитку технологій штучного інтелекту (Artificial Intelligence, AI). Зокрема, швидке зростання можливостей великих мовних моделей (Large Language Models, LLM) та систем генерації тексту висуває на передній план питання мультимодальної взаємодії з користувачем. Сучасні користувачі очікують, що цифрові асистенти будуть не лише «розумними» у плані діалогу, але й матимуть людську подобу для більш природної комунікації. Показовим є приклад ChatGPT від OpenAI – цей LLM-сервіс менш ніж за два місяці набрав понад 100 млн користувачів (січень 2023 року), встановивши рекорд найшвидшого зростання аудиторії серед споживчих застосунків [3]. Така масова популярність мовних AI-систем стимулює попит на їх інтеграцію в різні платформи, зокрема у вигляді реалістичних розмовних аватарів (віртуальних співрозмовників) для сервісів підтримки, навчальних програм, персональних помічників тощо.

Паралельно, ринок технологій цифрових аватарів демонструє вибухове зростання. За галузевими аналізами, глобальний ринок AI-аватарів у 2023 році оцінювався приблизно у 5,9 млрд доларів США і прогнозовано зростатиме на ~30% щорічно до 2032 року [4]. Це обумовлено інтересом бізнесу до віддаленого спілкування та онлайн-сервісів: зокрема, все більше компаній впроваджують віртуальних агентів і асистентів для взаємодії з клієнтами. Кількісно, число пристроїв з голосовими асистентами (Siri, Alexa тощо), які потенційно можуть бути доповнені візуальними аватарами, вже перевищило кількість населення планети – станом на 2024 рік у світі

використовуються понад 8,4 млрд цифрових голосових помічників [5]. Попит на реалістичну анімацію обличчя під мовлення також підживлюється індустрією розваг та освіти: від персоналізованих віртуальних ведучих і тренерів до ігор – скрізь потрібні персонажі, що природно говорять і виражають емоції синхронно зі звуком. Тому дослідження, націлені на підвищення якості таких технологій, є своєчасними і затребуваними.

Критичною вимогою для успішного використання «говорючих» аватарів є точна синхронізація руху губ з промовою. Від цього залежить переконливість і зрозумілість спілкування. Люди дуже чутливі до невідповідності між звуком та зображенням: навіть незначна розсинхронізація (десятки мілісекунд) помітний і спричиняє відчуття штучності. Некоректна артикуляція аватара може відволікати, знижувати довіру користувача до системи або навіть призводити до непорозумінь. Наприклад, в аудіовізуальній психології описано ефект МакГурка (McGurk effect) – коли звук не збігається з рухом губ, у глядача виникає ілюзія іншого почутого звуку [6]. Для людей з порушеннями слуху неточна робота аватара зробить неможливим читання з губ, що зведе користь такого помічника нанівець. Натомість ідеально синхронний та реалістичний аватар здатен суттєво підвищити якість користувацького досвіду: забезпечити ефект «живої» присутності, кращу емоційну залученість і довіру до цифрового співрозмовника.

З огляду на це, проблема підвищення точності синхронізації мовлення та рухів губ є надзвичайно актуальною для розробників систем віртуальної комунікації, телемедицини, навчальних платформ і багатьох інших галузей, де застосовуються інтерактивні віртуальні персонажі. А головне стабільність роботи моделі та можливості її масштабування до нових персонажів.

1.3 Методи двовимірної генерації руху губ

Генерація руху губ з використанням аудіо у двовимірному просторі передбачає синтез послідовності зображень обличчя, або відповідного регіону губ, згідно з вхідним звуковим сигналом мовлення. Іншими словами, на основі аудіо потрібно згенерувати реалістичні рухи губ на цільовому обличчі, забезпечуючи синхронність звуку та відео. Для розв'язання цього завдання в літературі запропоновано різноманітні підходи. Нижче наведено огляд цих підходів, їх принципів роботи, характерні формули і прикладів реалізацій.

1.3.1 Моделі на основі GAN

Змагальна генеративна мережа (Generative Adversarial Network, GAN) – це підхід глибинного навчання, де дві нейронні мережі, генератор (G) і дискримінатор (D), тренуються змагально: генератор намагається створити дані, що не відрізняються від справжніх, а дискримінатор намагається відрізнити згенеровані дані від реальних [7]. Задача генератора полягає в «обмані» дискримінатора, максимізуючи ймовірність прийняття штучних зразків за справжні. У класичній формулюванні GAN цільова функція має вигляд:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{real}}[\ln D(x)] + E_{z\sim p(z)}\left[\ln\left(1 - D\big(G(z)\big)\right)\right], \quad (1.1)$$

де x – реальні дані;

z – випадковий шум;

$D(x)$ – ймовірність, що дискримінатор класифікує зразок як справжній;

$G(z)$ – згенерований зразок на основі шуму z;

p_{real} – розподіл реальних даних;

$p(z)$ – розподіл випадкового шуму.

У випадку синтезу руху губ застосовуються умовні GAN-моделі, де генератору подається не просто шум, а й додаткова умова – наприклад, звук мовлення або зображення обличчя, яке слід анімувати. Дискримінатор, у свою чергу, теж отримує на вхід відповідні умови, наприклад, той самий звук, і намагається визначити, чи узгоджуються між собою аудіо та відео. Таким чином, мережа навчається генерувати реалістичні кадри обличчя, що відповідають поданому аудіо.

Одним із найвідоміших представників цього підходу є модель Wav2Lip [8], яка вважається основою подальших моделей синхронізації губ. Архітектура Wav2Lip складається з енкодер-декодерної генеративної мережі, що приймає на вхід поточний кадр обличчя з маскою на області рота та аудіопослідовність, представлену мел-спектрограмою. Відео-енкодер вилучає ознаки обличчя, враховуючи позу голови та освітлення, з неповного кадру та довільного опорного кадру того ж обличчя, а аудіо-енкодер отримує ознаки мовлення з аудіосигналу. Далі декодер об’єднує ці ознаки та генерує зображення нижньої частини обличчя, що синхронізоване з мовленням. Схематично роботу моделі представлена на рисунку 1.1.

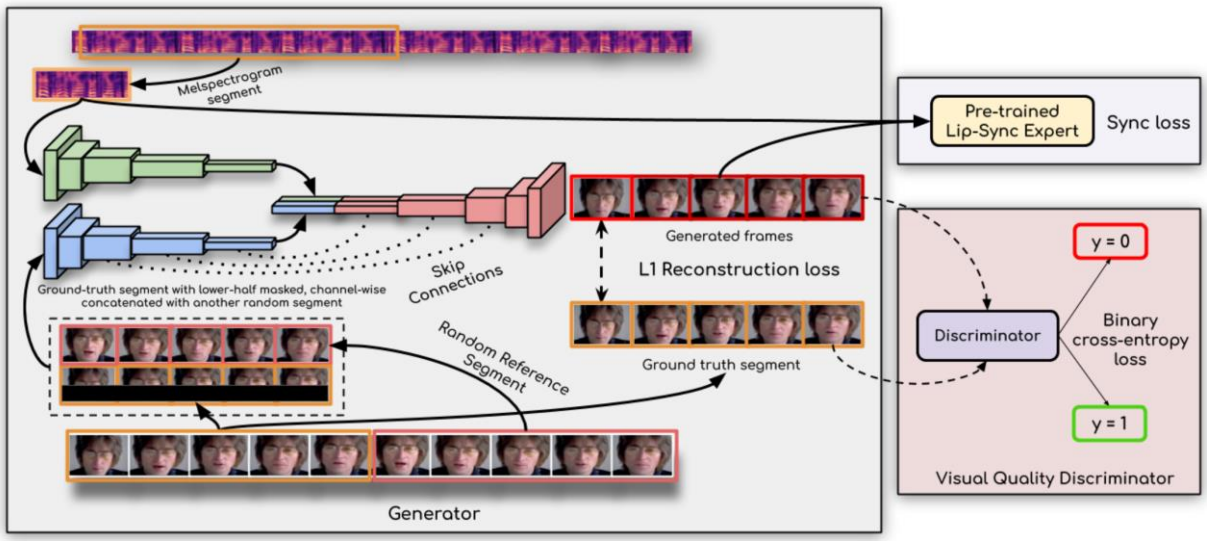

Рисунок 1.1 – Схема роботи моделі «Wav2Lip»

Для покращення якості навчання Wav2Lip застосовує додатковий дискримінатор синхронності – окрему попередньо навчену мережу SyncNet [9]. SyncNet оцінює ступінь відповідності між згенерованим відео та аудіо; генератор оптимізується так, щоб максимізувати цю відповідність. Додатково дискримінатор у Wav2Lip розглядає послідовності кадрів, шляхом подачі кількох згенерованих кадрів, об'єднаних каналами, для забезпечення часової узгодженості руху губ. Таким чином, GAN-модель навчається не лише відтворювати статичну реалістичність кожного окремого кадру, а й плавність руху в динаміці. Завдяки цьому підходу Wav2Lip досягає високої точності синхронізації губ з мовленням навіть для нових даних.

Недоліком класичних 2D GAN-рішень є те, що вони оперують на рівні зображень і часто страждають на певну розмитість деталей рота та артефакти при високій роздільній здатності відео. Наприклад, було зазначено, що Wav2Lip, тренований на кадрах 96×96, в умовах, коли модель не має багато прикладів саме цього обличчя, генерує менш чіткі текстури губ, оскільки мережі важко навчитися відновлювати високочастотні деталі лише зі звуку.

1.3.2 Дифузійні моделі

Дифузійні моделі (diffusion models) – це клас генеративних моделей, які успішно застосовуються для синтезу зображень завдяки здатності відтворювати дрібні деталі та складні розподіли даних. Основна ідея дифузійної моделі полягає у двох процесах: прямому процесі «дифузії» (diffusion), тобто поступовому додаванні шуму до зображення, та зворотному процесі «денойзингу» (denoising), тобто поступовому відновленні зображення із зашумленого стану [10]. Мережа навчається відновлювати оригінальні дані із зашумлених, що дозволяє їй

генерувати нові зразки шляхом проходження зворотного шляху від чистого шуму до фотореалістичного зображення.

Формально, нехай x_0 – реальне зображення, x_t – його стан після додавання шуму протягом t кроків. Прямий процес задається розподілом $q(x_t|x_{t-1})$ так, що x_T для великого T наближається до чистого гаусівського шуму. Зворотний процес моделюється параметризованим розподілом $p_\theta(x_{t-1}|x_t, \text{cond})$, де cond – умовна інформація, в нашому випадку – аудіо. Навчання дифузійної моделі зазвичай зводиться до мінімізації спрощеної метрики відновлення шуму, наприклад:

$$L_{diff} = E_{x_0,\epsilon,t}||\epsilon - \epsilon_\theta(x_t, a, t)||^2, \quad (1.2)$$

де t – крок дифузії;

ϵ – шум, доданий на кроці t;

$\epsilon_\theta(x_t, a, t)$ – оцінка мережі для цього шуму на основі зашумленого зображення x_t та умовного аудіо a.

Під час генерування виконується ітеративний процес: починаючи з випадкового шуму $x_T \sim N(0, I)$, мережа послідовно віднімає шум, керуючись аудіо-контекстом, поступово отримуючи все більш чітке зображення $x_{T-1}, x_{T-2}, \ldots, x_0$. Завдяки такому підходу дифузійні моделі здатні синтезувати фотореалістичні обличчя з високою роздільною здатністю, перевершуючи за візуальною достовірністю традиційні змагальні генеративні мережі. Ці моделі, однак, вимагають значних обчислювальних ресурсів і часу для генерування одного зображення, оскільки типове число кроків T може становити кілька сотень або тисяч.

В останні роки з’явилося багато робіт, які успішно застосували дифузійні моделі. Зокрема, Diff2Lip [11] – одна з перших дифузійних моделей, застосованих до задачі синхронізації руху губ (LipSync), працює у піксельному просторі: модель генерує послідовність кадрів обличчя, керуючись аудіо, і забезпечує кращу чіткість та збереження ідентичності

порівняно з GAN-аналогами. Модель використовує архітектуру, подібну до U-Net, обумовлену аудіоознаками, для поступового зменшення шуму в зображенні рота на кожному кроці дифузійного процесу. Умовна інформація подається в U-Net через механізм вкладення аудіоознак на різних масштабах, а також може застосовуватися навчання як з умовами, так і без них, для гнучкого керування ступенем впливу аудіо на генерацію.

Diff2Lip генерує більш реалістичні деталі у ділянці рота, зокрема зуби та губи, і краще зберігає міміку, ніж системи на основі GAN, такі як Wav2Lip. Це підтверджує, що дифузійні моделі, хоч і потребують більше ресурсів, встановлюють новий рівень якості у задачі дублювання обличчя під аудіо.

1.3.3 Архітектури зі спеціалізованим відображенням ознак

Окрім підходів на основі генеративно-змагальних мереж (Generative Adversarial Networks, GAN) та дифузійних моделей існують методи, які реалізують аудіокеровану генерацію руху губ у двовимірному (2D) відео через використання проміжних або спеціалізованих просторів ознак. Ці підходи спрямовані на підвищення фотореалістичності та точності синхронізації руху губ із аудіосигналом, зосереджуючись на обробці лише ротової зони, що дозволяє зберегти деталі та ідентичність обличчя.

DINet (Deformation Inpainting Network) [12] реалізує підхід «генерації через деформацію». Модель використовує кілька еталонних кадрів із цільовим обличчям, які пропускаються через енкодер для отримання латентних ознак. Потім ці ознаки деформуються відповідно до аудіоінформації, що дозволяє синхронізувати рух губ із вхідним мовленням. На завершальному етапі декодер відновлює області, що зазнали змін, зберігаючи при цьому загальні риси обличчя. Така деформаційна схема дозволяє передавати високочастотні деталі, зокрема зуби та зморшки, оскільки більша частина обличчя залишається незмінною, і лише ротова

зона зазнає аудіозалежної трансформації. У результаті DINet забезпечує реалістичну синхронізацію губ зберігаючи високу якість генерації. Також треба зазначити те, що модель можна оптимізувати до роботи в реальному часі.

Іншим прикладом можна вважати IP-LAP (Identity-Preserving Landmark-Appearance Network) [13], що використовує траєкторію двовимірних (2D) лендмарків нижньої частини обличчя як проміжне представлення. Модель спочатку перетворює аудіо на послідовність координат губ, а потім, маючи цільове обличчя із замаскованою ротовою ділянкою, декодує це обличчя з урахуванням лендмарків, що задають форму рота для кожного кадру. Завдяки такому розділенню задача генерації зводиться до якісного відтворення потрібної конфігурації рота, а не до повної обробки всіх пікселів обличчя. У результаті IP-LAP досягає високої візуальної узгодженості та чіткої синхронізації звуку і руху губ .

Нарешті, MuseTalk [14] демонструє механізм «латентного домальовування». Мережа проектує кадр обличчя, з прихованою ротовою зоною, у латентний простір варіаційного автоенкодера (Variational Autoencoder, VAE), де на основі аудіо створює потрібну анімацію рота. Такий підхід дозволяє зберегти глобальний вигляд обличчя та пов’язані з ним деталі, тоді як лише обмежена зона рота «домальовується» під конкретний звуковий контент. Завдяки інтеграції аудіоознак у декількох масштабах MuseTalk забезпечує реалістичне відтворення текстур.

Усі ці системи демонструють, що у 2D-генерації мовлення успішно застосовуються гібридні та багатокомпонентні архітектури, в яких анімація обличчя досягається деформацією або доповненням лише ротової зони. Це дозволяє забезпечити високу фотореалістичність і якісну синхронізацію зі звуком без повної генерації всіх пікселів кадру.

1.4 Методи тривимірної генерації мовлення

Методи тривимірної (3D) генерації мовлення використовують просторове представлення обличчя для моделювання геометрії голови та рухів губ, що дозволяє візуалізувати результати з різних кутів огляду. На відміну від двовимірних підходів, ці методи явно враховують просторову структуру обличчя, форму, рельєф та динаміку – що підвищує реалістичність рухів і дозволяє змінювати ракурс камери. Сучасні 3D-методи можна класифікувати за двома основними підходами: використання параметричних 3D-моделей обличчя та застосування нейронних полів випромінювання (Neural Radiance Fields, NeRF).

1.4.1 Методи на основі 3DMM

Підходи, що базуються на тривимірних морфованих моделях обличчя (3D Morphable Models, 3DMM), представляють обличчя як набір параметрів, які керують його формою, виразом та позою. У контексті аудіокерованої генерації мовлення нейронні мережі прогнозують ці параметри на основі вхідного аудіосигналу для кожного кадру. Потім ці параметри застосовуються до 3D-моделі обличчя, після чого виконується рендеринг для створення реалістичної анімації. Перевага цього підходу полягає в точному контролі атрибутів обличчя, таких як збереження ідентичності, керування виразами та поворотами голови, що сприяє отриманню детальних та реалістичних анімацій.

Одним із прикладів є система EmoTalker [15], яка генерує відео обличчя, що говорить, на основі єдиного зображення та аудіозапису мовлення. Ця модель особливу увагу приділяє виразності: вона не потребує додаткових міток емоцій чи еталонних відеокадрів. Параметри 3DMM розділяються на дві групи: рухи губ та решта (міміка й повороти голови). Спочатку аудіоенкодером виділяються ознаки, що відповідають рухам губ,

а потім окремим блоком генеруються параметри виразу обличчя. Для моделювання немовних рухів (блимання, міміка, нахили голови, пов’язані з інтонацією) EmoTalker застосовує двоетапний підхід:

– відображає послідовність згенерованих коефіцієнтів міміки у простір словника кодів, навченого представляти емоційно забарвлені вирази;

– використовує архітектуру трансформера, що явно моделює зв’язки між аудіо і різними типами рухів обличчя.

Така архітектура дозволяє врахувати емоційні флуктуації голосу: в результаті модель синхронізує губи з мовленням і одночасно генерує природні мімічні реакції та рухи голови, узгоджені з аудіо. У дослідженні показано, що EmoTalker досягає високого рівня реалізму: на емоційно забарвлених аудіоданих вона забезпечує точну синхронізацію губ та яскраві вирази обличчя при генерації обличь, що говорять, для раніше не бачених осіб.

1.4.2 Методи на основі NeRF

На відміну від підходів, що явно моделюють геометрію через тривимірну сіткову модель, методи на основі нейронних полів випромінювання (NeRF) використовують ці поля для безпосереднього відтворення обличчя у вигляді щільного 3D-представлення. NeRF дозволяє за заданим станом (наприклад, виразом обличчя) генерувати зображення з будь-якої точки огляду, забезпечуючи високий рівень фотореалізму та узгодженість у тривимірному просторі. Це привабило дослідників до використання NeRF для генерації облич, що говорять, оскільки такий підхід природно гарантує 3D-узгоджені результати та можливість зміни ракурсу обличчя.

GeneFace++ [16] є удосконаленим підходом на основі NeRF, спрямованим на узагальнення моделі на різні аудіовходи та значне

підвищення швидкодії рендерингу – понад 45 кадрів на секунду. Попередник, GeneFace, поєднав NeRF з окремим генератором рухів. GeneFace++ розвиває цю ідею, покращуючи довгострокову узгодженість руху губ та стійкість до нестандартного аудіо. Для цього використовується окремий генератор «аудіо-в-міміку» з додатковими аудіоознаками та часовою функцією втрат, а також згладжування параметрів. Спеціально оптимізований NeRF-рендерер (Instant Motion-to-Video) забезпечує генерацію в реальному часі. Це перша NeRF-система, що досягає стабільної, реалістичної генерації без потреби тренування під конкретну особу.

1.5 Метрики аудіо-візуальної синхронізації

Оцінювання якості згенерованих відео з говорючими обличчями базується на кількох стандартизованих метриках, які дозволяють кількісно вимірювати синхронність між аудіо та відео, а також фотореалістичність кадрів. Найбільш поширеними є SSIM, PSNR, LPIPS, LSE-D, LSE-C, FID та FVD. Кожна з них виконує свою функцію в комплексному аналізі якості моделі.

Індекс структурної подібності (Structural Similarity Index, SSIM) [14] – метрика, яка оцінює схожість між згенерованим кадром і еталоном (ground truth) на основі локальних патернів яскравості, контрасту та структури. Формула має вигляд:

$$\mathrm{SSIM}(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)}, \tag{1.3}$$

де μ_X, μ_Y – середні значення;

σ_X^2, σ_Y^2 – дисперсії;

σ_{XY} – коваріація;

C_1, C_2 – малі сталі.

Значення SSIM ∈ [0, 1], де 1 означає повну відповідність. У задачах синхронізації SSIM використовується як показник візуальної стабільності та структурної узгодженості між згенерованим та еталонним відео.

Пікове відношення сигнал/шум (Peak Signal-to-Noise Ratio, PSNR) [14] – класична метрика якості зображення, яка вимірює співвідношення сигнал/шум між згенерованим кадром та еталонним. Вона базується на середньоквадратичній помилці (MSE) та обчислюється за формулою:

$$\text{PSNR} = 10 \log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}} \right), \tag{1.4}$$

де MAX – максимальне можливе значення пікселя в зображенні;

MSE – середньоквадратична помилка між згенерованим та еталонним зображеннями.

Більше значення PSNR свідчить про меншу піксельну помилку, тобто кращу реконструкцію.

Навчена перцептивна подібність фрагментів зображень (Learned Perceptual Image Patch Similarity, LPIPS) [13] – метрика, яка вимірює візуальну відстань між зображеннями в просторі ознак глибоких нейронних мереж (наприклад, VGG). На відміну від SSIM та PSNR, LPIPS краще корелює з людським сприйняттям. Менше значення LPIPS означає вищу перцептивну подібність.

Помилка синхронізації губ-відстань (Lip Sync Error-Distance, LSE-D) [13] – метрика, що оцінює ступінь синхронізації між звуком і рухами губ. Вона базується на моделі аудіо-відео синхронізації, яка витягує ознаки з аудіо та зображення, після чого обчислюється середня евклідова відстань між ними:

$$\text{LSE-D} = \frac{1}{T}\sum_{t=1}^{T} |\, f_a(a_t) - f_v(v_t) \,|_2 , \qquad (1.5)$$

де $f_a(a_t)$, $f_v(v_t)$ – ознаки аудіо і відео, витягнуті мережею аудіо-відео синхронізації;

T – кількість кадрів.

Менше значення LSE-D вказує на кращу синхронізацію між аудіо та відео.

Помилка синхронізації губ-впевненість (Lip Sync Error-Confidence, LSE-C) [13] – показник упевненості моделі аудіо-відео синхронізації у тому, що аудіо та відео синхронізовані. Він розраховується як різниця між найменшою та медіанною відстанню між витягнутими ознаками, що відображає наскільки однозначно мережа «бачить» правильну синхронізацію. Високе LSE-C свідчить про гарну синхронізацію.

FID (Frechet Inception Distance) [11] – одна з найвідоміших метрик для оцінки реалістичності згенерованих зображень. Вона обчислює відстань Фреше між статистиками ознак реальних та синтетичних кадрів:

$$\text{FID} = |\mu_r - \mu_g|_2^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r \Sigma_g\right)^{1/2}\right), \qquad (1.6)$$

де (μ_r, Σ_r) та (μ_g, Σ_g) – середні та коваріації для реальних і згенерованих зображень відповідно.

Низьке FID означає, що розподіл згенерованих кадрів схожий на реальний, тобто відео виглядає природно.

FVD (Frechet Video Distance) [11] – розширення метрики FID для відео. Вона базується на ознаках, витягнутих з відео-розпізнавальних моделей (наприклад, I3D), і оцінює динамічну узгодженість та плавність руху у відеоряді. Так само, як і FID, вона порівнює статистики ознак реальних та згенерованих відео. FVD враховує як візуальні, так і часові

характеристики: чим нижче значення, тим ближчий згенерований відеоряд до реального за динамікою.

Зазначені метрики часто використовуються в комбінації: SSIM, PSNR, LPIPS – для оцінки якості кадрів, LSE-D, LSE-C – для перевірки аудіо-візуальної синхронізації, FID, FVD – для оцінки візуальної реалістичності і динаміки. Такий комплекс дозволяє всебічно порівнювати моделі за ключовими аспектами якості генерації говорючих обличь.

1.6 Постановка задачі

Метою кваліфікаційної роботи є дослідження підходів до підвищення точності синхронізації мовлення та рухів губ у відео з віртуальними аватарами. Це передбачає розробку та вдосконалення метрики синхронізації на основі моделей аудіо-відео синхронізації, яка використовується у функції втрат під час навчання більшості моделей типу LipSync. Досягнення цієї мети сприятиме забезпеченню більш реалістичної аудіо-візуальної взаємодії, та головне стабільності роботи моделі, що має важливе значення для цифрових асистентів, телемедицини, освітніх платформ та інших сучасних сервісів.

В процесі аналізу предметної області було виділено декілька задач, які мають бути виконані в ході кваліфікаційної роботи:

– провести аналіз сучасних моделей для генерації синхронізованих з аудіо рухів губ;

– дослідити принципи роботи моделей, які забезпечують аудіо-візуальну синхронізацію, якщо така використовується;

– запропонувати модифікації до існуючої моделі оцінки синхронізації з метою підвищення її стабільності та точності;

– виконати аналіз результатів навчання та роботи отриманої моделі аудіо-відео синхронізації, порівняти її з не модифікованим варіантом;

– впровадити розроблену удосконалену модель оцінки у процес навчання моделі LipSync;

– виконати аналіз отриманих результатів за допомогою стандартних показників для визначення впливу удосконаленої метрики на точність синхронізації.

Реалізація зазначених завдань сприятиме підвищенню якості аудіо-візуальної синхронізації в системах генерації 2D аватарів.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1 Архітектурні основи моделі SyncNet

Модель SyncNet призначена для автоматичного визначення синхронності між рухами губ на відео та звуком мовлення. В основі SyncNet – дві паралельні нейронні мережі (енкодери): візуальна і аудіальна, які спільно навчаються порівнювати область губ та відповідний голосовий сигнал. Кінцева мета – оцінити ступінь синхронності: високу для правильного поєднання відеоряду і аудіодоріжки та низьку для невідповідних пар.

2.1.1 Оригінальна архітектура SyncNet

Оригінальна модель SyncNet [9] реалізує два окремі енкодери – візуальний та аудіальний. Візуальний енкодер обробляє послідовність із 5 кадрів відео (близько 0,2 секунди) розміром 111×111 пікселів у відтінках сірого, які містять лише область рота. Аудіоенкодер отримує відповідний фрагмент мовлення у вигляді 13 мел-частотних кепстральних коефіцієнтів (MFCC) з 20 часовими кроками, що відповідає 0,2 секунди аудіо з частотою дискретизації 100 Гц. На рисунку 2.1 можна побачити вхідні дані для моделі.

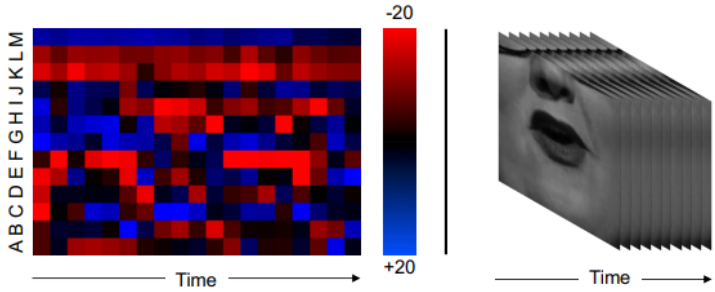

Рисунок 2.1 – Вхідні дані для SyncNet

Обидва енкодери представлені згортковими мережами: візуальний енкодер базується на архітектурі раннього злиття (Early Fusion), а аудіоенкодер – на модифікованій VGG-M зі зміненими розмірами фільтрів для прийому входів нестандартних розмірів. На виході кожен енкодер формує вектор фіксованої розмірності (вкладення) розміром 256 елементів. Мережа навчається за допомогою контрастивної функції втрат, яка мінімізує евклідову відстань між векторами аудіо та відео для синхронізованих пар і максимізує її для несинхронізованих. Загальну модель можна побачити на рисунку 2.2.

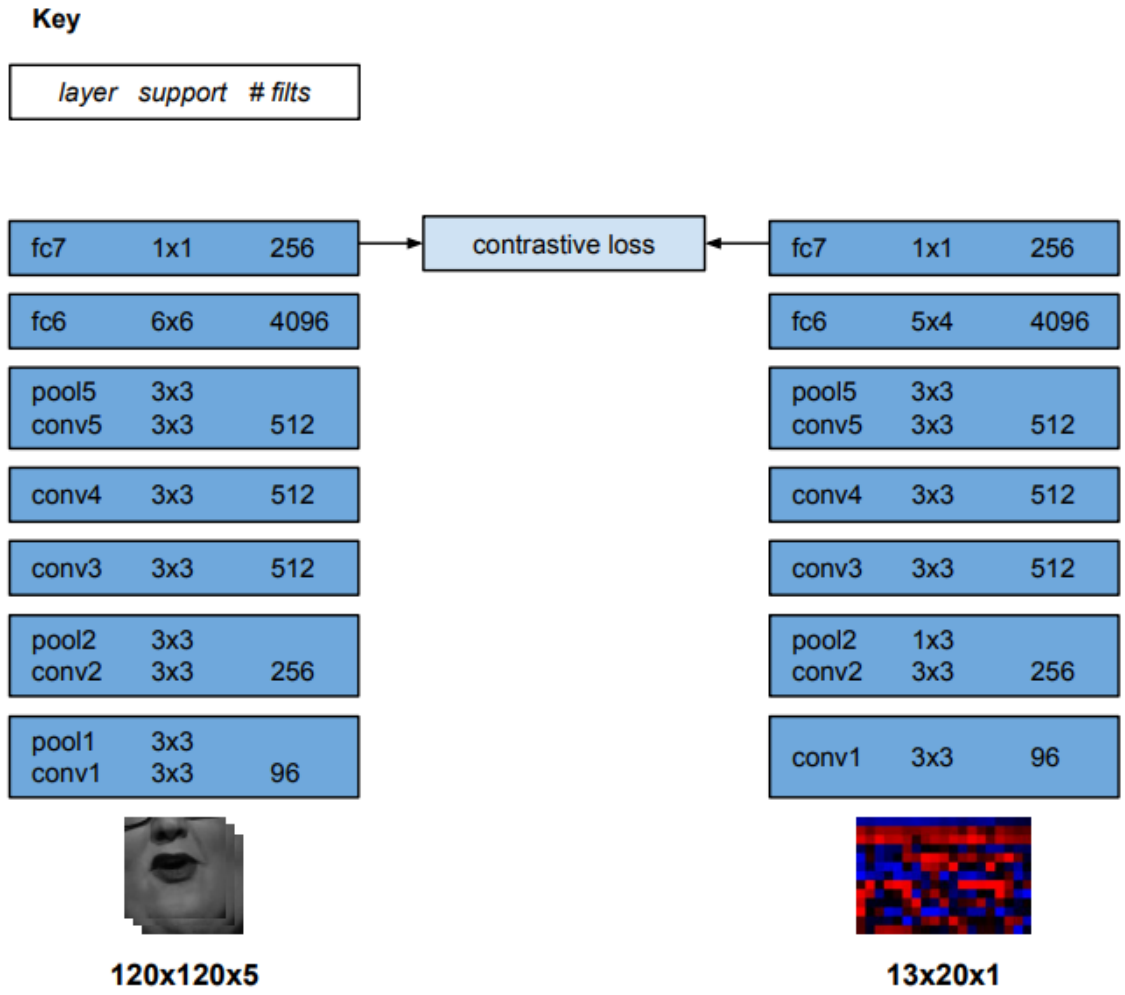

Рисунок 2.2 – Загальна архітектура моделі SyncNet

2.1.2 Використання SyncNet у Wav2Lip

У моделі Wav2Lip [8] SyncNet функціонує як експертний дискримінатор синхронності губ. Візуальний вхід модифіковано з чорно-білих зображень 96×96 пікселів на кольорові кадри більшої роздільності, а архітектура енкодерів стала глибшою з використанням блоків ResNet. Функція втрат змінена з контрастивної на бінарну крос-ентропію з косинусною подібністю. Візуальний енкодер складається з приблизно 17 згорткових шарів і продукує 1024-вимірний вектор ознак, тоді як аудіоенкодер працює зі спектрограмою розміром 80 частот × 16 часових кроків і також видає 1024-вимірне представлення.

Принципова особливість імплементації SyncNet у Wav2Lip полягає в тому, що параметри дискримінатора не змінюються під час навчання генератора, зберігаючи таким чином об'єктивну оцінку синхронності. Це рішення дозволяє Wav2Lip генерувати дуже точно синхронізовані рухи губ, оскільки генератор отримує стабільний і неупереджений сигнал зворотного зв'язку.

2.1.3 Використання SyncNet у DINet

SyncNet у моделі DINet [12] відзначається суттєвими архітектурними відмінностями в підході до обробки вхідних даних. Аудіовхід представлений не традиційними MFCC, а високорівневими логітами мовлення, отриманими від моделей розпізнавання мовлення DeepSpeech. Ці логіти мають розмірність 29 і представляють ймовірнісні оцінки фонем або символів для кожного часового кроку, що надає моделі структуровану лінгвістичну інформацію.

Візуальний вхід у DINet обробляється з високою роздільністю – 256×256 пікселів для області рота, що забезпечує збереження детальних текстурних особливостей губ. Найбільш суттєва

інновація полягає в методі обробки отриманих ознак: замість безпосереднього обчислення косинусної подібності між векторами аудіо та відео, DINet використовує підхід просторового перетворення аудіовектора та подальшої згорткової обробки. Аудіовектор розмірності 128 перетворюється у тензор з просторовими розмірами, аналогічними карті ознак обличчя, після чого аудіо та відео тензори конкатенуються по вимірності каналів.

Отримане об'єднане представлення обробляється за допомогою спеціалізованого згорткового модуля (об'єднувального енкодера), який складається з двох згорткових шарів. Перший шар згортає 256 каналів до 128, а другий трансформує їх в один канал, формуючи просторову карту оцінки синхронності. Ця карта може бути усереднена або з неї може бути вибране максимальне значення для отримання скалярного показника впевненості системи у синхронності пари аудіо-відео.

Такий архітектурний підхід забезпечує DINet кращу стійкість до варіацій мови та акцентів, можливість просторової локалізації областей з різною якістю синхронізації та загальне підвищення якості візуальних деталей у синтезованих рухах губ. Також використання логітів дозволяє моделі досягти кращих результатів на англійській мові, ніж MFCC.

2.2 Проблеми навчання аудіо-візуальних моделей синхронізації губ

Навчання моделей, що зв'язують аудіо та відео для синхронізації губ, характеризується низкою специфічних проблем, які необхідно враховувати під час розробки та тренування таких систем. Ці проблеми зумовлені як особливостями мультимодальних даних, так і складністю встановлення відповідностей між рухами губ та звуками мовлення.

Першою суттєвою перешкодою є стабілізація функції втрат на плато. Типовим проявом цієї проблеми є стабілізація значення функції втрат близько 0,693, що відповідає значенню $-\ln(0{,}5)$ у випадку бінарної крос-

ентропії (binary cross-entropy). Таке значення свідчить, що модель не здатна розрізняти класи і фактично виконує випадкове вгадування з імовірністю 50%. За такої ситуації нейронна мережа постійно продукує ймовірність синхронності близько 0,5, не надаючи корисної інформації про узгодженість аудіо та відео. Це може виникати через невдалу ініціалізацію початкових параметрів або надмірну складність завдання.

Неточна синхронізація навчальних даних є другою критичною проблемою. Для ефективного навчання необхідні коректно синхронізовані зразки. Якщо набір даних містить випадки, де звук випереджає або відстає від відео, але при цьому помилково позначені як синхронні, модель отримує суперечливі сигнали навчання. Це призводить до вивчення хибних закономірностей або неспроможності виявити справжні залежності. Аналогічна ситуація виникає при генерації негативних прикладів – якщо «несинхронне» аудіо надто подібне до правильного, модель втрачає здатність адекватно розрізняти класи.

Варіації положення голови та ракурсу суттєво впливають на якість навчання. Моделі типу SyncNet чутливі до змін візуального представлення обличчя, оскільки працюють безпосередньо з піксельними даними. При поворотах голови губи видно під іншим кутом, їхня форма геометрично змінюється порівняно з фронтальним виглядом, тоді як звукова складова залишається незмінною. Експериментальні дослідження підтверджують, що оригінальна модель SyncNet демонструвала нестабільність при трансформаціях зображення через неспроможність врахувати тривимірну природу обличчя та компенсувати різні ракурси.

Обмежена якість візуальних даних значно ускладнює процес навчання. Якщо область рота вирізана неточно або кадри мають низьку роздільність, модель отримує недостатньо інформації про артикуляцію. Дрібні рухи язика та тонкі артикуляційні відмінності губ стають невиразними. Внаслідок цього встановлення аудіо-візуальних відповідностей ускладнюється, а нейронна мережа може плутати різні звуки

через їхню візуальну подібність у розмитому стані. Саме тому в сучасних реалізаціях використовується збільшений масштаб зображення обличчя.

Недостатня різноманітність навчальних даних призводить до слабкої здатності узагальнення. Мультимодальні моделі, що працюють з мовленням і артикуляцією, повинні узагальнювати через численні виміри варіативності: різні диктори мають індивідуальні особливості мовлення, різні мови характеризуються відмінними фонетичними наборами, різноманітні умови освітлення та фони впливають на якість візуальної інформації. За умови обмеженої різноманітності даних модель ризикує перенавчитися на конкретному акценті чи диктора або взагалі не виявити справжніх закономірностей. Наприклад, система, навчена виключно на англомовних відеозаписах, часто демонструє погіршену працездатність з іншими мовами через відмінності артикуляційних патернів.

Технічні обмеження процесу навчання також створюють суттєві перешкоди. Через високу розмірність відео та аудіо, моделі синхронізації губ часто обмежені малим розміром навчального пакету (batch). Це означає, що на кожній ітерації градієнт обчислюється за невеликою підвибіркою даних, яка може не відображати всю різноманітність набору. Як наслідок, шум градієнта стає значним, а процес оновлення вагових коефіцієнтів – нестабільним. У результаті навчання може сходитися надзвичайно повільно або зупинятися на плато.

Складність мультимодального узагальнення є фундаментальною проблемою. Модель повинна виявити узагальнені ознаки, інваріантні щодо диктора та мови, які відображають взаємозв'язок між мовленням і рухами артикуляційних органів. Це значно складніше, ніж навчання в межах однієї модальності. З позиції теорії інформації, аудіо та відео є двома окремими джерелами даних, і їхнє об'єднання підвищує ентропію вхідної інформації, що ускладнює виділення спільних компонентів та встановлення взаємозв'язків між ними.

Урахування цих проблемних аспектів на етапі проектування архітектури та підготовки навчальних даних має вирішальне значення для успішного навчання моделей синхронізації губ та досягнення високої якості результатів.

2.3 Оновлення мережі SyncNet

2.3.1 Згорткові шари та субдискретизація

Архітектура модифікованої моделі SyncNet використовує згорткові шари (convolutional layers) для вилучення ознак із вхідних відео- та аудіоданих [17]. Згорткова операція формально визначається як згортка ваг фільтра з регіоном вхідного сигналу. Нехай $x_c(p, q)$ – значення активації на вхідному шарі з каналом c у позиції (p, q), а $w_{d,c}(i, j)$ – вага ядра згортки для вихідного каналу d і вхідного каналу c на зсуві (i, j). Тоді вихід згорткового шару без активації в каналі d для просторової позиції (u, v) можна записати як суму добутків відповідних значень і ваг у рецептивному полі:

$$y_d(u, v) = \sum_{c=1}^{C_{in}} \sum_{i=0}^{H_k-1} \sum_{j=0}^{W_k-1} x_c(u + i, v + j) w_{d,c}(i, j) + b_d, \qquad (2.1)$$

де $H_k \times W_k$ – розмір ядра згортки;

C_{in} – кількість каналів вхідного зображення;

b_d – зміщення вихідного каналу d.

Таким чином, кожен нейрон згорткового шару обчислює зважену суму пікселів попереднього шару в межах свого рецептивного поля.

Для зменшення розмірності результату часто застосовується крок згортки $S > 1$ (stride). Вихідні нейрони обчислюються лише для кожного S-того положення, що зменшує розмір карти ознак приблизно у S разів по

кожній осі. Наприклад, якщо ширина вхідного зображення W, розмір ядра K, а крок S, то ширина вихідної карти ознак дорівнює $(W - K + 2P)/S + 1$, де P – доповнення (padding). Це означає, що при $S = 2$ вихідна ширина приблизно вдвічі менша за вхідну.

Окрім використання кроку, субдискретизація може досягатися агрегувальними шарами (pooling layers). Агрегувальний шар діє незалежно на кожен канал глибини та зменшує просторовий розмір представлення. Найпоширеніший приклад – максимізаційне агрегування (max-pooling) з вікном 2×2 і кроком 2, яке знижує дискретизацію активацій удвічі по ширині й висоті, відкидаючи 75% значень активацій. У результаті після одного такого шару розмір зображення ознак зменшується вдвічі, що скорочує кількість параметрів на наступних шарах і обсяг обчислень, а також допомагає контролювати перенавчання мережі. Чергування згорткових шарів з операціями субдискретизації дозволяє виділяти все більш абстрактні ознаки, поступово зменшуючи просторову роздільність представлення.

2.3.2 Блоки самоуваги

Модифікована модель SyncNet, за потреби, може включати спеціальні блоки самоуваги (self-attention blocks) [18], які реалізують механізм уваги для контекстуального зважування ознак. Механізм самоуваги дозволяє моделі акцентувати увагу на різних частинах послідовності ознак, враховуючи їх взаємозв'язки. Кожен такий блок обчислює вихідні представлення як зважену суму вхідних, де ваги визначаються схожістю між ознаками.

Формально самоувага оперує трьома матрицями: запитів Q, ключів K і значень V (query, key, value), які отримують шляхом лінійного перетворення вхідних ознак. Нехай $X \in R^{T \times d}$ – матриця вхідних ознак для T елементів послідовності, розмірності d. Тоді: $Q = XW^Q$, $K = XW^K$, $V =$

XW^V, де $W^Q, W^K, W^V \in R^{d \times d_k}$ – параметри (матриці ваг) перетворень запитів, ключів і значень відповідно, а d_k – розмірність простору ключів/запитів. Далі вихід уваги для матриці X визначається як:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2.2)$$

де функція softmax застосовується до кожного ряду матриці подібності $QK^T/\sqrt{d_k}$.

Іншими словами, спочатку обчислюється матриця скалярних добутків $QK^T \in R^{T \times T}$, що містить міри схожості між кожною парою «запит–ключ». Потім ці схожості нормуються (поділяються на $\sqrt{d_k}$) для стабільності градієнтів при великих розмірностях d_k, після чого до них застосовується softmax, щоб отримати вагові коефіцієнти сумування. На завершення кожен вихідний вектор отримують як зважену суму значень V з цими коефіцієнтами уваги.

Завдяки цьому елемент послідовності через свій запит Q_i самостійно звертається до всіх елементів через їх ключі K_j і вибирає, наскільки сильно враховувати кожен із їхніх векторів значення V_j у своїй вихідній активації. У випадку self-attention всі Q, K, V породжені з одного й того ж набору ознак X, тобто модель оцінює внутрішньомодальні взаємозв'язки, наприклад, між різними часовими фрагментами однієї модальності.

Зазвичай використовується багатоголова увага (multi-head attention) [18], коли кілька таких механізмів уваги (голів) з різними матрицями W^Q, W^K, W^V працюють паралельно на різних підпросторах ознак, а їх результати конкатенуються. Це дозволяє моделі одночасно фокусуватися на різних аспектах взаємозв'язків у даних. В підсумку, блок самоуваги підвищує здатність мережі враховувати глобальний контекст:

наприклад, в нашій задачі він може зіставляти певні візуальні рухи губ з відповідними аудіо-особливостями в різних часових позиціях.

2.3.3 Нормалізація і активація

Для забезпечення стабільного навчання та прискорення збіжності в кожному блоці мережі застосовується шар нормалізації вихідних активацій, після чого застосовується нелінійна функція активації. Нормалізація покращує поширення сигналу по мережі [19], підтримуючи значення активацій у збалансованому діапазоні. Зокрема, часто використовується пошарове нормалізування типу пакетна нормалізація (batch normalization), що вирівнює розподіл активацій по міні-пакету.

Для кожного каналу k обчислюється середнє μ_k та дисперсія σ_k^2 значень активацій x_k по всіх об'єктах у пакеті; потім виконується нормалізація:

$$\widehat{x_k} = \frac{x_k - \mu_k}{\sqrt{\sigma_k^2 + \epsilon}}, \tag{2.3}$$

де ϵ – мале число для запобігання діленню на нуль.

Після цього вводяться два параметри – масштаб γ_k і зміщення β_k, за допомогою яких мережа може відновити потрібний діапазон значень: $y_k = \gamma_k \widehat{x_k} + \beta_k$. Таким чином, нормалізований вихід y_k все ще може репрезентувати різноманітні розподіли, якщо це потрібно моделі, але без зміщення по середньому і масштабу, які заважають навчанню.

Нормалізація зменшує проблему внутрішнього ковзання параметрів (internal covariate shift) і дозволяє використовувати більші швидкості навчання, сприяючи швидшій і стабільнішій збіжності мережі. В нашій реалізації використано групову нормалізацію (Group Normalization) з

певною кількістю груп, наприклад 32, замість пакетної нормалізації, оскільки це краще працює при невеликих розмірах пакету. Проте ідея залишається подібною: активації приводяться до однакового масштабу в межах груп компонентів перед застосуванням активаційної функції.

Після нормалізації до кожного нормалізованого компонента застосовується функція активації, що вводить нелінійність у модель. Нелінійність необхідна, щоб мережа могла апроксимувати складні нелінійні залежності. У сучасних згорткових мережах найбільш поширена функція активації – це випрямлений лінійний елемент (Rectified Linear Unit, ReLU). Вона визначається покомпонентно як:

$$\mathrm{ReLU}(z) = \max(0, z), \tag{2.4}$$

тобто перетворює від'ємні значення в нуль, а додатні залишає без змін. ReLU є простою кусково-лінійною функцією, яка значно покращує навчання глибоких мереж – вона розв'язує проблему зникнення градієнта для глибоких шарів, оскільки градієнт крізь ReLU або 0, або постійний для $z > 0$, та сприяє розрідженості представлень, адже багато виходів стають нульовими.

У випадку, коли вхід значно негативний, ReLU просто відкидає його, що запобігає насиченню негативних нейронів, але водночас може призвести до феномену «вимирання нейронів», коли нейрон ніколи не активується через постійно негативний вхід. Для розв'язання цієї проблеми інколи використовують модифіковану активацію витікаючий ReLU (Leaky ReLU), яка пропускає невелику частку негативного сигналу:

$$\mathrm{LeakyReLU}(z) = \max(\alpha z, z), \tag{2.5}$$

де α – малий коефіцієнт, наприклад 0.01, що визначає нахил для від'ємної частини.

Таким чином, Leaky ReLU не обнуляє негативні значення повністю, а лише масштабує їх, зберігаючи мінімальний потік градієнта навіть у «вимкненому» стані нейрона.

В архітектурі нашої моделі на прихованих шарах використовується сучасна варіація активації – сигмоїдна лінійна одиниця (Sigmoid Linear Unit, SiLU), відома як Swish, яка визначається як $\mathrm{SiLU}(z) = z \cdot \sigma(z)$, де $\sigma(z)$ – сигмоїдна функція. Втім, принцип дії той самий: активація додає нелінійність, необхідну для моделювання складних взаємозалежностей між аудіо- та відеоознаками.

Узагальнюючи, можна зазначити, що нормалізація і активація спільно формують типову послідовність обробки в кожному блоці нейромережі: нормалізований вихід згортки подається через нелінійний поріг, що забезпечує ефективне навчання багатошарової моделі.

2.3.4 Загальна архітектура модифікованої SyncNet

Загальна архітектура SyncNet побудована за модульним принципом і складається з двох окремих енкодерів – візуального та аудіального, які паралельно обробляють відеоряд і аудіосигнал відповідно. Загальна архітектурна оновлена схема зображена на рисунку 2.3.

Кожен енкодер має подібну структуру: спочатку застосовується згортковий шар для початкового виділення ознак, після чого йде каскад шарів пониження роздільної здатності. Такі шари реалізовано у вигляді резидуальних блоків (ResNetBlock2D) із можливістю зменшення просторової або часо-частотної для аудіо роздільної здатності на заданий фактор, а після кожного такого блоку за потреби додається модуль самоуваги (AttentionBlock2D), який покращує вилучення глобальних залежностей у даних.

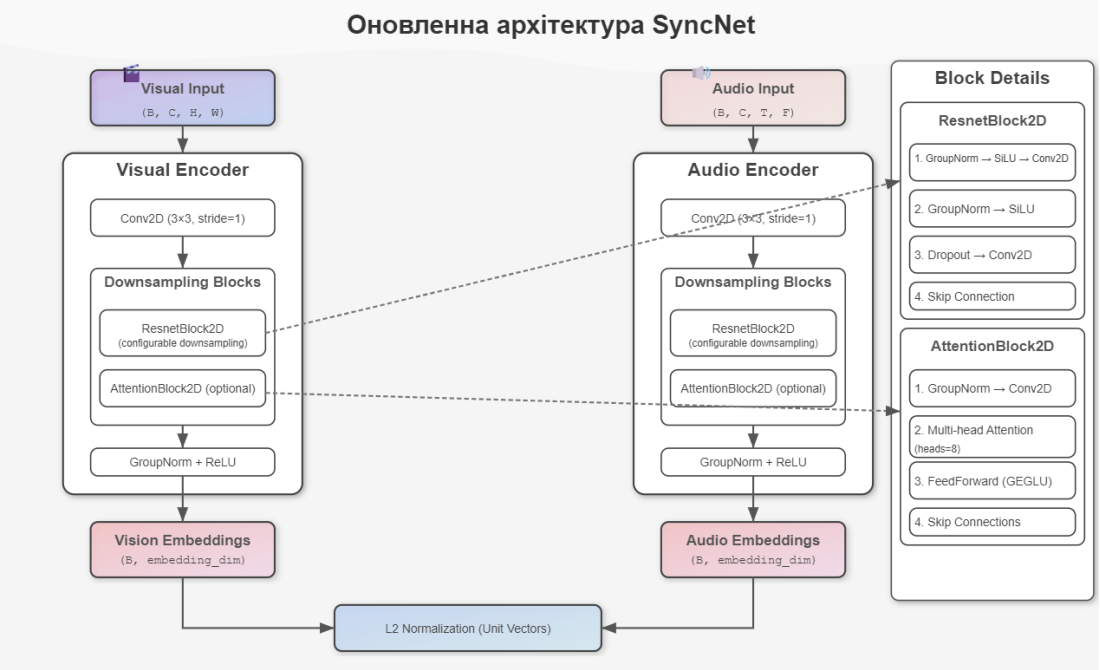

Рисунок 2.3 – Схематичне зображення пропонованої архітектури SyncNet

В кінці енкодеру застосовується нормалізація ознак (групова нормалізація, GroupNorm) та активація (випрямлений лінійний елемент, ReLU), що готує вихід до формування векторного представлення. У результаті роботи візуального та аудіального енкодерів отримуються векторні вкладення (embeddings) фіксованого розміру для кожної з модальностей, які надалі нормуються до одиничної довжини методом L_2-нормалізації.

2.3.5 Косинусна подібність ознак

На фінальному етапі архітектура SyncNet порівнює отримані векторні представлення аудіо та відео шляхом обчислення косинусної міри подібності між ними. Косинусна подібність двох векторів визначається як косинус кута між ними у багатовимірному просторі ознак:

$$\cos(a, b) = \frac{a \cdot b}{||a||_2 \cdot ||b||_2}, \tag{2.6}$$

де $a \cdot b$ – скалярний добуток векторів, а в знаменнику – добуток їх евклідових норм.

Ця величина набуває значень від -1 до 1, де $\cos(a, b) = 1$, якщо вектори співнапрямлені (ідеально ідентичні за напрямом ознаки), $\cos = 0$ при ортогональності (ознаки незалежні), і $\cos = -1$ якщо вектори протилежні (максимально різняться).

Косинусна міра є масштабно-інваріантною, оскільки залежить тільки від напрямку векторів, а не їх довжини. Це робить її зручною для порівняння ознак, коли абсолютна величина активацій може нести менше сенсу, ніж їх відносний розподіл. У нашій моделі перед обчисленням подібності вектори ознак нормуються до одиничної довжини ($||a||_2 = ||b||_2 = 1$) за допомогою L_2-нормалізації. За таких умов формула спрощується до $\cos(a, b) = a \cdot b$ – тобто звичайного скалярного добутку, оскільки довжини вже рівні 1. Тому фактично мережа навчається продукувати унітарні ембедінги, тобто вектори ознак на одиничній сфері, для яких скалярний добуток безпосередньо відображає ступінь схожості.

Косинусна подібність має кілька корисних властивостей: симетричність – $\cos(a, b) = \cos(b, a)$, що узгоджується з інтуїтивним поняттям взаємної подібності; обмеженість інтервалом $[-1,1]$, завдяки чому її можна інтерпретувати як коефіцієнт кореляції між двома наборами ознак; незалежність від масштабу – якщо збільшити всі компоненти a чи b на сталий множник, косинусна метрика не зміниться. Останнє особливо важливо: мережа може вільно варіювати загальну амплітуду вихідних векторів, що може покращувати оптимізацію, не впливаючи на значення міри близькості, адже воно визначається лише напрямками.

2.3.6 Процес навчання моделі SyncNet

Навчання модифікованої SyncNet формулюється як задача двокласової класифікації: модель повинна давати висновок, чи є дана пара аудіо та відео сегментів синхронізованою (клас «in-sync») чи ні («out-of-sync») [9]. Для цього використовується функція втрат на основі бінарної крос-ентропії (binary cross-entropy, BCE), яка обчислюється відносно показника косинусної подібності двох модальностей.

Зокрема, вихідна косинусна подібність інтерпретується як імовірність того, що відповідна аудіо-відео пара синхронна (P_{sync}). В силу нормування ембедінгів та використання небінарної активації на останньому шарі, значення $\cos(a, v)$ обмежене від 0 до 1, оскільки негативні значення практично обнуляються, тому її зручно трактувати як ймовірність позитивного класу.

Нехай s_i – передбачена моделлю косинусна подібність для і-ї пари (аудіо a_i та відео v_i), а y_i – бінарна мітка (1 якщо пара синхронна, 0 якщо ні). Тоді помилка (втрати) на одному прикладі задаються як:

$$l_i = -(y_i \log s_i + (1 - y_i) \log(1 - s_i)), \tag{2.7}$$

а функція втрат на міні-пакеті (batch loss) обчислюється як середнє цих величин по N прикладах:

$$L_{BCE} = \frac{1}{N} \sum_{i=1} l_i \,. \tag{2.8}$$

Ця формула є визначенням крос-ентропії між передбаченою моделлю імовірністю s_i та справжнім розподілом $(y_i, 1 - y_i)$ на дві класи. Вона штрафує відхилення s_i від цільового значення: якщо $y_i = 1$ (пара

синхронна), мінімум досягається при $s_i \to 1$ (максимальна схожість ознак), а якщо $y_i = 0$ (не синхронна) – при $s_i \to 0$.

Таким чином, мінімізуючи цю функцію втрат, мережа навчається збільшувати косинусну подібність для справжніх пар «губи-мова» і зменшувати для невідповідних. Застосування саме BCE на основі косинусної близькості замість, наприклад, контрастивної метрики з відступом, мотивоване інтерпретованістю і простотою: вихід моделі прямо інтерпретується як ймовірність синхронності, що спрощує поріг прийняття рішення і оцінку якості. Крім того, BCE не вимагає вибору гіперпараметрів типу відступу (маржі) і природно збалансований при використанні рівного числа позитивних та негативних зразків.

У процесі навчання позитивні приклади (правильні аудіо-відео пари) та негативні (навмисно невідповідні фрагменти аудіо і відео) подаються в модель з приблизно рівною частотою. Це гарантує, що модель бачить достатньо ситуацій обох класів і не зміщується в бік тривіального вирішення, наприклад, завжди «несинхронно».

Для мінімізації функції втрат використовується оптимізатор адаптивного моменту з ваговим затуханням (Adaptive Moment Estimation with Weight Decay, AdamW). AdamW – це сучасний стохастичний градієнтний метод, що поєднує ідеї імпульсу (momentum) та адаптивних кроків навчання, як у RMSProp. Принцип імпульсу полягає в тому, що оновлення ваг містить частку від попереднього оновлення, згладжуючи коливання градієнта і прискорюючи рух в напрямку стійкого спадання функції втрат.

Зокрема, Adam обчислює експоненційно згладжене середнє градієнтів (перша моментна оцінка m_t) та експоненційно згладжене середнє квадратів градієнтів (друга моментна оцінка v_t) на кожній ітерації t. Нехай g_t – градієнт функції втрат по параметру θ на кроку t. Тоді рекурентні співвідношення оптимізатора можна подати так:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad (2.9)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \quad (2.10)$$

де β_1, β_2 – коефіцієнти згладжування, наприклад, 0.9 і 0.999 відповідно.

Таким чином, m_t накопичує інформацію про напрямок градієнта, що є аналогом імпульсу – фактично ковзне середнє, що пам'ятає попередні градієнти, а v_t накопичує інформацію про масштаб градієнта, тобто середній квадрат, що є аналогом механізму вирівнювання кроку як у RMSProp.

Після цього для оновлення параметра використовується нормований скоригований момент:

$$\widehat{m_t} = \frac{m_t}{1 - \beta_1^t}, \widehat{v_t} = \frac{v_t}{1 - \beta_2^t}, \quad (2.11)$$

які враховують початкове зміщення (bias correction), і крок оновлення:

$$\theta_{t+1} = \theta_t - \alpha \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}} + \epsilon}, \quad (2.12)$$

де α – швидкість навчання;

ϵ – мале додатне число для стабільності, що запобігає діленню на нуль.

В реалізації AdamW додається ще член для покарання великих ваг (L2-регуляризація або weight decay): після основного оновлення параметр додатково множиться на фактор $(1 - \alpha\lambda)$, де λ – коефіцієнт регуляризації. Це зменшує величини ваг до нуля на кожному кроці на величину, пропорційну їх розміру, що запобігає неконтрольованому росту параметрів і покращує здатність моделі до узагальнення.

Концептуально, поєднання імпульсу та адаптивного масштабу кроку в AdamW дає змогу оптимізатору швидко проходити плато та долати дрібні

локальні мінімуми, при цьому автоматично підбираючи розмір кроку для кожного параметра. Це особливо корисно для нашої моделі, яка є відносно глибокою: AdamW забезпечує швидку та стабільну збіжність, що було підтверджено експериментально.

Запропонована архітектура SyncNet спроєктована з урахуванням принципів EfficientNet [20], які передбачають збалансоване масштабування глибини, ширини і роздільної здатності мережі для досягнення оптимальної продуктивності. Іншими словами, базову конфігурацію моделі можна пропорційно розширювати – збільшуючи кількість шарів (глибину), кількість каналів у шарах (ширину) та розмір вхідних даних (роздільну здатність). Такий підхід відповідає концепції ефективного масштабування, оскільки дозволяє підвищувати точність моделі без невиправданого зростання вимог до обчислювальних ресурсів. Завдяки збалансованому нарощуванню характеристик мережі, модель залишається відносно компактною та придатною для використання в умовах обмежених ресурсів, забезпечуючи при цьому належний рівень точності й ефективності.

Для оптимізації процесу також застосовувалися сучасні техніки на кшталт змішаних обчислень з плаваючою точкою (Mixed Precision Training), що зменшує використання пам'яті та контроль градієнтних норм (gradient clipping) для запобігання вибуху градієнтів.

2.4 Адаптована модель DINet для аудіо-візуальної синхронізації губ

У кваліфікаційній роботі для тестування впливу якості аудіо-візуальної синхронізації було обрано модель DINet. Ця модель була розроблена для реалістичного візуального дубляжу обличчя на відео високої роздільної здатності. Модель була обрана через свою просту архітектуру та ефективність. Адаптована версія DINet зберігає базову архітектуру двох частин – деформаційну та реконструктивну, але містить декілька модифікацій, спрямованих на підвищення ефективності моделі, зокрема

через обмеженість ресурсів. Нижче описано архітектуру цієї моделі з урахуванням внесених змін.

2.4.1 Архітектура адаптованої моделі DINet

Архітектура моделі DINet [12] складається з деформаційної частини P^D та частини реконструкції P^I. Деформаційна частина отримує на вході кадр з джерела I_s, п'ять довідкових зображень обличчя I_{ref} для різних форм рота, та керуючий аудіосигнал A_d, і генерує деформовані ознаки рота F_d, узгоджені з аудіо та позою голови. Реконструктивна частина об'єднує F_d з ознаками джерела F_s та за допомогою декодера ознак відновлює остаточне дубльоване зображення обличчя I_o. Повну структуру моделі можна побачити на рисунку 2.4.

Деформаційна частина P^D відповідає за просторове перетворення ознак з довідкових зображень таким чином, щоб форма рота синхронізувалася з мовленням, а положення голови відповідало джерельному кадру. З цією метою використовуються дві мережі-енкодери: енкодер зображень, який витягає ознаки джерела F_s із кадру I_s та ознаки довідки F_{ref} із I_{ref}, і енкодер аудіо, який аналізує керуюче аудіо A_d.

В оригінальній моделі аудіо представлялося послідовністю із $T = 5$ векторів розмірності 29, отриманих із моделі DeepSpeech. У адаптованій версії для подання аудіосигналу замість DeepSpeech використано модель HuBERT: аудіо A_d перетворюється на послідовність логітів HuBERT розмірності 29, те ж саме що і у DeepSpeech, але з трішки іншим словником перетворень, на кожен відрізок часу, узгоджений з кадровою частотою відео. Ця послідовність надходить до енкодера аудіо, який генерує ознаку аудіо $F_{audio} \in R^{128}$ – компактне представлення мовного вмісту поточного фрагмента аудіо.

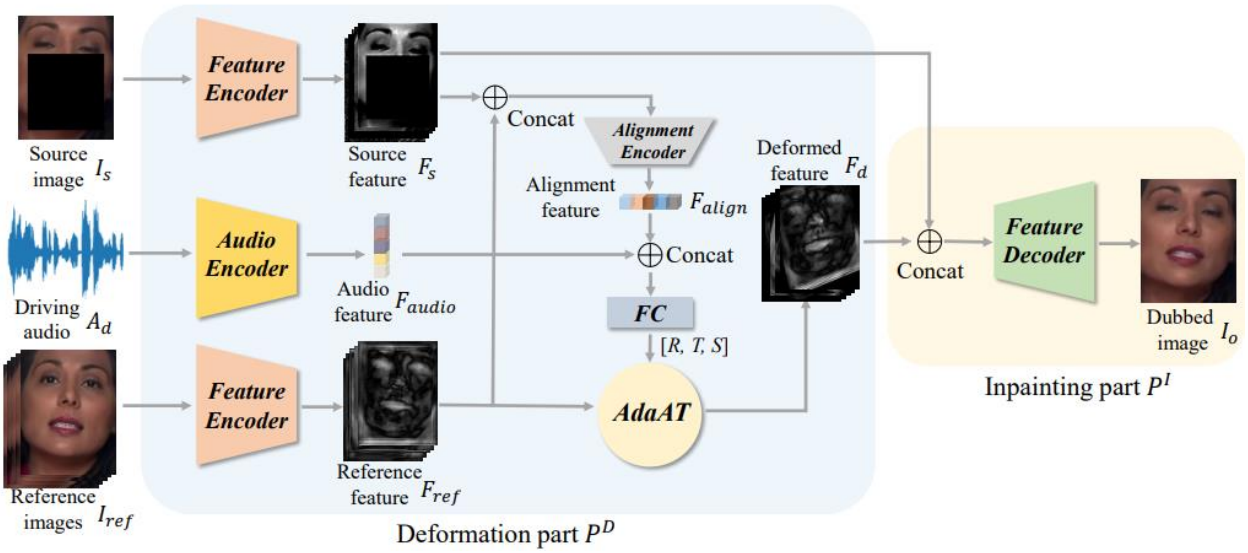

Рисунок 2.4 – Архітектура моделі DINet

Паралельно енкодер зображень обробляє кадр I_s і п'ять зображень I_{ref}, формуючи відповідно джерельний просторовий тензор F_s та довідковий тензор F_{ref}. В оригінальній моделі розмірність каналів цих ознак становила 256, проте в адаптованій реалізації кількість каналів у карті ознак рота після афінного перетворення скорочено з 256 до 128. Тобто, замість $F_{ref} \in R^{(256 \times H/4 \times W/4)}$ генерується $F_{ref} \in R^{(128 \times H/4 \times W/4)}$, і відповідно операція просторової деформації (AdaAT) виконується для $c = 1, \ldots, 128$ каналів. Це спрощення зменшує обсяг параметрів та обчислень без значної втрати деталізації результату, а також не потребує змінювати розмірність даних для SyncNet.

Для об'єднання інформації про позу голови та вміст мовлення використовується енкодер узгодження: він отримує на вході конкатенацію ознак F_s і F_{ref} та обчислює вектор узгоджених характеристик $F_{align} \in R^{128}$. Цей вектор містить інформацію про відносне розташування голови на довідкових зображеннях і в кадрі джерела.

Далі аудіо-ознака F_{audio} та ознака узгодження F_{align} спільно використовуються для керування модулем просторової деформації. Замість щільного поля зміщень у DINet застосовується оператор адаптивного

афінного перетворення (Adaptive Affine Transformation, AdaAT). AdaAT [12] обчислює для кожного каналу довідкових ознак окремі коефіцієнти афінного перетворення: кути повороту $\{\theta_c\}_{c=1}^{C}$, масштаби $\{s_c\}_{c=1}^{C}$, та зсуви $\{(t_c^x, t_c^y)\}_{c=1}^{C}$.

Отримані параметри використовуються для афінного перетворення кожного канального шару тензора F_{ref}, що можна виразити формулою перетворення координат пікселів (x_c, y_c) у кожному каналі c:

$$\begin{pmatrix} \widehat{x_c} \\ \widehat{y_c} \\ 1 \end{pmatrix} = \begin{pmatrix} s_c \cos\theta_c & -s_c \sin\theta_c & t_c^x \\ s_c \sin\theta_c & s_c \cos\theta_c & t_c^y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix}, \quad (2.13)$$

де $\widehat{x_c}$та $\widehat{y_c}$ – нові координати пікселя після афінного перетворення;

s_c – коефіцієнт масштабування;

θ_c – кут повороту;

t_c^x та t_c^y – параметри зсуву для каналу c.

У результаті застосування AdaAT довідковий тензор F_{ref} перетворюється на деформовані ознаки F_d, які містять форму рота, синхронізовану з аудіо, та вже вирівняні з позою голови на зображенні I_s. На рисунку 2.5 можна побачити візуалізацію роботи з вхідними даними та їх перетворення.

Реконструктивна частина P^I використовується для відновлення області рота на фінальному зображенні. Вона отримує на вході деформовані ознаки рота F_d та просторові ознаки обличчя з джерела F_s, об'єднані шляхом конкатенації по каналам. Декодер ознак, набір згорткових шарів, обробляє цей об'єднаний тензор і заповнює відсутні пікселі в масці рота, генеруючи дубльоване обличчя I_o розміру $H \times W$.

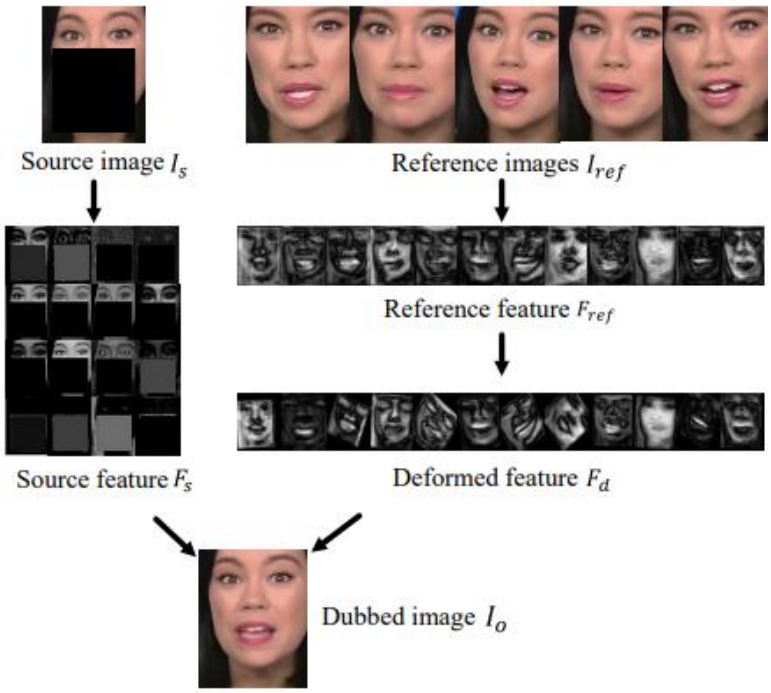

Рисунок 2.5 – Візуалізація використовуваних даних для генерації DINet

Таким чином, поєднання деформації і реконструкції дозволяє синтезувати реалістичне обличчя, що зберігає усі високочастотні текстурні деталі вихідного зображення та рухається у повній відповідності до аудіо.

2.4.2 Процес навчання адаптованої моделі DINet

Для навчання моделі використовується підхід, аналогічний до оригінального DINet, із сумарною функцією втрат, що складається з трьох компонентів: перцепційної, генеративно-адверсарної та аудіовізуальної синхронізації [12].

Перцепційна втрата L_{perc} оцінює різницю між дубльованим зображенням I_o та реальним зображенням I_{gt} на двох масштабах зображення з використанням ознак, виділених попередньо навченою мережею VGG-19:

$$L_{perc} = \sum_{i=1}^{N} \frac{\left|\left|V_i(I_o) - V_i(I_{gt})\right|\right|_1 + \left|\left|V_i(\hat{I}_o) - V_i(\hat{I}_{gt})\right|\right|_1}{2NW_iH_iC_i}, \quad (2.14)$$

де $V_i(\cdot)$ представляє і-й шар мережі VGG-19;

W_i, H_i, C_i – розміри карти ознак в і-му шарі (ширина, висота, кількість каналів);

$\hat{I}_o$ та $\hat{I}_{gt}$ – зменшені вдвічі версії зображень I_o та I_{gt} (до розміру $\frac{H}{2} \times \frac{W}{2}$).

Застосування втрати на двох масштабах зображення допомагає моделі краще зберігати як глобальну структуру обличчя, так і локальні деталі текстури. Нормалізація на $2NW_iH_iC_i$ врівноважує внесок різних шарів у загальну функцію втрат.

Адверсарна втрата L_{adv} забезпечує фотореалістичність згенерованого обличчя. Використовується ефективний підхід найменших квадратів, де функція L_{adv} складається з втрати дискримінатора L_D і втрати генератора L_G:

$$L_{adv} = L_D + L_G, \quad (2.15)$$

$$L_D = \frac{1}{2}\left(D(I_{gt}) - 1\right)^2 + \frac{1}{2}(D(I_o) - 0)^2, \quad (2.16)$$

$$L_G = (D(I_o) - 1)^2, \quad (2.17)$$

де D позначає дискримінатор, який намагається відрізнити реальні зображення від згенерованих, а генератор (DINet) намагається змусити дискримінатор класифікувати I_o як реальне зображення.

Адверсарна втрата застосовується як до окремих кадрів, так і до послідовностей із п'яти кадрів, що допомагає забезпечити часову узгодженість результатів.

Нарешті, втрата синхронізації L_{sync} забезпечує відповідність руху губ поданому аудіо. У адаптованій моделі, замість використання прямої оцінки синхронізації, модель повертає аудіо-вкладення та відео-вкладення. Косинусна подібність між цими векторами використовується для обчислення рівня синхронізації:

$$P_{sync} = \frac{v \cdot s}{\max\left(\left|\left|v\right|\right|_2 \cdot \left|\left|s\right|\right|_2, \epsilon\right)}, \tag{2.18}$$

де v – відео-вкладення;

s – аудіо-вкладення;

$\left|\left|\cdot\right|\right|_2$ – L_2-норма;

ϵ – мала константа для уникнення ділення на нуль.

Втрата синхронізації обчислюється як середньоквадратична помилка між обчисленою подібністю та цільовим значенням 1 (повна синхронізація):

$$L_{sync} = \left|\left|P_{sync} - 1\right|\right|_2^2. \tag{2.19}$$

Також, на відміну від оригінального DINet, додано четвертий компонент до загальної функції втрат – пряму втрату на зображення L_{img}. Ця втрата обчислює піксельну різницю між згенерованим зображенням та реальним за допомогою середньоквадратичної помилки:

$$L_{img} = \left|\left|I_o - I_{gt}\right|\right|_2^2, \tag{2.20}$$

де I_o – згенероване зображення;

I_{gt} – еталонне зображення.

Підсумковий критерій навчання адаптованої моделі DINet являє собою зважену суму всіх описаних компонент:

$$L_{total} = \lambda_{perc} \cdot L_{perc} + \lambda_{adv} \cdot L_{adv} + \lambda_{sync} \cdot L_{sync} + \lambda_{img} \cdot L_{img}, \qquad (2.21)$$

де λ_{perc}, λ_{adv}, λ_{img} та λ_{sync} – вагові коефіцієнти для відповідних функцій втрат.

У адаптованій моделі використано вагові коефіцієнти, близькі до запропонованих в оригіналі: $\lambda_{perc} = 10$, $\lambda_{sync} = 0.1$, $\lambda_{adv} = 1$, $\lambda_{img} = 1$.

Навчання проводиться на вибраному корпусі відеоданих із використанням оптимізатора AdamW. Для прискорення обчислень на GPU модель тренується в режимі змішаної точності: замість 32-бітних чисел з плаваючою комою використовується формат 16-бітних чисел з плаваючою комою. Перехід на половинну точність не призводить до деградації якості генерованих зображень, але суттєво підвищує продуктивність. Такий самий підхід до оптимізації навчання використовувався для пункту 2.3.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

У цьому розділі описані експериментальні дослідження, що були виконані у цій роботі. Показано обрані датасети, кроки їх обробки та аргументація їх вибору. Продемонстровано результати навчання моделей, представлені графіки та їх аналіз.

3.1 Вибір та обробка даних

3.1.1 Характеристика обраних датасетів

Успішне навчання моделей синхронізації губ суттєво залежить від якості та репрезентативності вхідних даних. Для забезпечення надійного процесу навчання, робота з аудіо-візуальною синхронізацією потребує збалансованого набору відеоматеріалів, що демонструють різноманітні умови зйомки, дикторів та артикуляційні особливості. Для вирішення цієї задачі було обрано два взаємодоповнюючих датасети: HDTF (High-Definition Talking Face) [21] як основний та Hallo3 [22] як додатковий.

HDTF являє собою великомасштабний аудіо-візуальний датасет, спеціально створений для задач генерації «говорячих облич» у високій якості. Він містить приблизно 16 годин відео з роздільною здатністю 720p-1080p, де понад 300 різних людей вимовляють більше 10 тисяч різноманітних фраз. Ключовою перевагою HDTF є його висока роздільна здатність та «реальність» середовища зйомки: відео зібрані з мережі Інтернет (переважно YouTube, 2018-2020 роки), що забезпечує різноманітність сцен та умов освітлення, зберігаючи при цьому фокус на обличчях мовців.

Особливо важливою характеристикою HDTF, що зумовила його вибір як основного датасету для навчання DINet, є висока якість аудіо-візуальної синхронізації. Додатковою перевагою HDTF є структура датасету – для

кожної людини наявні фрагменти відео тривалістю понад 2 хвилини, що створює сприятливі умови для тонкого налаштування моделей під конкретні особливості артикуляції окремих дикторів.

У порівнянні з іншими «in-the-wild» датасетами, HDTF демонструє кращу роздільну здатність (до 1080p проти 720p у VoxCeleb2 [23]) та оптимальне співвідношення якості й кількості даних. На відміну від інших наборів даних, таких як VoxCeleb2 (2442 години, близько 720p) чи LRS3 (118 тисяч речень, 720p) [24], які збиралися для задач розпізнавання мовця або мовлення і можуть мітити шум (кілька осіб у кадрі, нецентрований рот тощо), HDTF цілеспрямовано фокусується на крупних планах облич у момент мовлення. Така особливість забезпечує модель синхронізації більш детальними та якісними даними руху губ, що критично важливо для навчання точної моделі.

Для додаткової перевірки та збагачення даних було також використано датасет Hallo3 – відкритий набір даних для задач портретної анімації, розроблений дослідниками з університету Фудань. Hallo3 відзначається своїм масштабом, містячи понад 70 годин відео високої якості, де відібрано виключно обличчя людей, що говорять (так звані «pure talking-head videos»). Датасет включає понад 100 000 відеофрагментів (точна кількість – 101 543 оброблених відео) з текстовими описами до кожного, що значно спрощує категоризацію та вибірку матеріалів за необхідними критеріями.

Суттєвою перевагою Hallo3 є те, що він спеціально підготовлений для навчання генеративних моделей, має ліцензію CC BY-NC-ND 4.0 і надає вже розмічені та відсортовані дані мовлення, що мінімізує необхідність додаткової обробки. Hallo3 також включає приблизно 50 відеофрагментів із «диких» сцен (вулиця, натовп тощо), що дозволяє оцінити стійкість моделі до різноманітних фонових умов. Більшість сцен – це короткі фрагменти з фільмів та репортажів.

Після аналізу доступних обчислювальних ресурсів та оцінки масштабів датасету Hallo3 було прийнято рішення про його скорочення та цільовий відбір найбільш релевантних відеофрагментів. З повного набору було відібрано 5000 відеокліпів, що відповідають таким критеріям: особа диктора розташована по центру кадру (не біля краю екрану); обличчя чітко видиме та симетричне з похибкою не більше 17% (тобто людина дивиться практично прямо в камеру); губи чітко видимі на всіх кадрах відеофрагменту. Такий селективний підхід дозволив зосередитися на найбільш якісних зразках, що оптимально відповідають вимогам до валідаційного набору даних.

Комбінація HDTF і відібраних фрагментів Hallo3 дозволяє ефективно поєднати якість (HDTF: висока чіткість, реальні обличчя) та різноманітність (Hallo3: багато різних дикторів у контрольованих умовах) даних, що підвищує надійність навчання. Обидва датасети доповнюють один одного: якщо HDTF надає реальні новинні й блогові відео з різними людьми, то Hallo3 концентрується на портретних відео з багатою мімікою, знятих у більш контрольованих умовах, а також містить відео різними мовами (англійська, китайська тощо), що сприяє кращому узагальненню моделі для різних мовних артикуляцій.

Важливо відзначити, що для завдання аудіо-візуальної синхронізації суттєвими характеристиками набору даних є якість зображення обличчя, чіткість руху губ та точність часової відповідності між аудіо та відео. HDTF та селективна вибірка з Hallo3 в сукупності задовольняють ці вимоги, забезпечуючи необхідні умови для успішного навчання моделей синхронізації губ та їх валідації в різноманітних умовах.

3.1.2 Методологія обробки відеоданих

Для ефективного використання обраних датасетів необхідно здійснити комплексну попередню обробку з метою підготовки якісних

даних для навчання моделей. При розробці методології обробки було враховано специфіку вибраних датасетів та вимоги моделей синхронізації губ.

Аналіз структури HDTF та Hallo3 дозволив визначити оптимальний набір етапів обробки, уникаючи надлишкових операцій. Зокрема, традиційні для відеообробки кроки, такі як розпізнавання меж сцен (shot detection), не були включені до методології, оскільки обрані датасети містять здебільшого однорідні фрагменти з незмінним диктором та ракурсом. Це відрізняє їх від наборів даних, як-от LRS3, де часта зміна сцен та дикторів вимагала б обов'язкового виявлення переходів між кадрами. Також було вирішено відмовитися від додаткової фільтрації відео за якістю зображення, оскільки обидва датасети вже містять матеріали високої роздільної здатності з чітко видимими обличчями.

Розглянемо ключові етапи обробки даних, що формують цілісну методологію підготовки:

Перевірка цілісності відеофайлів – етап спрямований на виявлення та вилучення пошкоджених або неповних відеофайлів, які могли б спричинити помилки під час навчання моделі. Процес передбачає спробу відкриття кожного відео та зчитування його першого кадру; у випадку неможливості доступу до вмісту файл позначається як пошкоджений та вилучається з подальшої обробки.

Уніфікація формату даних – приведення всіх відео до стандартизованих технічних параметрів: частоти кадрів 25 fps та частоти дискретизації аудіо 16 кГц у моно-режимі. Цей крок забезпечує узгоджене співвідношення між аудіо та відео в усіх зразках, що є критичним для точності синхронізації.

Розрахунок ключових точок обличчя – використання інструменту OpenFace для обчислення координат 68 характерних точок обличчя (лендмарків) на кожному кадрі. Ці точки включають контури очей, брів, носа та, особливо важливих для нашої задачі, губ. Лендмарки є

фундаментальною основою для наступних етапів локалізації та вирізання області рота.

Кадрування відео – вирізання області, що містить губи та навколишню зону, на основі координат обчислених лендмарків. Цей процес дозволяє зосередити увагу моделі на релевантних елементах зображення, виключаючи фонові деталі та непотрібні частини обличчя.

Сегментація відео – розбиття довгих відеоматеріалів на короткі фрагменти оптимальної тривалості (5–10 секунд). Такий підхід забезпечує ефективніше управління даними та уникає проблем із обробкою занадто великих послідовностей кадрів.

Аналіз аудіо-візуальної синхронізації – перевірка та коригування часової відповідності між аудіодоріжкою та рухом губ. Процес базується на алгоритмічному визначенні потенційного зсуву між аудіо та відео з наступним застосуванням відповідної корекції для забезпечення ідеальної синхронізації.

Повторний розрахунок лендмарків – обчислення оновлених координат ключових точок для скоригованих та сегментованих відеофрагментів, що забезпечує точну відповідність між координатами та фактичним положенням елементів обличчя після всіх попередніх трансформацій.

На етапі перевірки цілісності відеофайлів із початкових 399 відеофайлів датасету HDTF було визначено 3 нефункціональні файли, які не відкривалися або містили помилки. Ці відео виключено з подальшого розгляду, залишивши 396 повноцінних відео для наступних етапів. У датасеті Hallo3, який містив 5000 попередньо відібраних відеофрагментів, пошкоджених файлів не виявлено.

При уніфікації формату даних всі 396 відеофайлів HDTF та 5000 відео з Hallo3 успішно конвертовано до єдиного формату, забезпечивши уніфіковану основу для подальшої обробки.

Розрахунок ключових точок обличчя для 396 відео HDTF пройшов успішно, що дозволило точно визначити положення губ на кожному кадрі. У випадку з Hallo3 виникли деякі складнощі з частиною відеоматеріалів через темне освітлення, що призвело до неточностей у розпізнаванні лендмарків OpenFace. Проте, більшість відео було успішно оброблено.

Кадрування відео було успішно виконано для всіх відеоматеріалів HDTF та Hallo3, що значно зменшило обсяг даних для подальшої обробки без втрати важливої інформації.

Сегментація відео проводилася лише для датасету HDTF, оскільки Hallo3 вже містив короткі відеофрагменти оптимальної тривалості. Для HDTF після сегментації отримано 5690 відеофрагментів, які надалі були розподілені на тренувальну (4836 кліпів, приблизно 85%) та валідаційну (854 кліпи, близько 15%) вибірки.

На етапі аналізу аудіо-візуальної синхронізації виявлено, що переважна більшість фрагментів вже були синхронізовані (офсет 0 кадрів) з високим коефіцієнтом впевненості. Однак для деяких кліпів алгоритм виявив ненульовий офсет – тобто аудіо трохи випереджало або відставало від відео. Особливим випадком стало відео «WDA_SuzanDelBene», для якого було виявлено систематичне зміщення приблизно –10 кадрів (аудіо запізнювалось на 10 кадрів відносно відео).

Після повторного розрахунку лендмарків для скоригованих та сегментованих відеофрагментів було отримано фінальний набір даних з точними просторовими орієнтирами для всіх 5690 відеофрагментів HDTF та 5000 відео Hallo3.

Оптимальний порядок застосування описаних етапів обробки був визначений експериментально з урахуванням залежностей між операціями та впливу кожного кроку на якість кінцевих даних.

Першим кроком обробки стала перевірка цілісності відеофайлів для видалення пошкоджених матеріалів, що дозволило уникнути подальших збоїв у процесі. Другий крок – уніфікація формату даних, що забезпечив

стандартизовану основу для всіх наступних операцій. На цьому етапі було виявлено специфічний випадок з відео «WDA_SuzanDelBene», де асинхронність між аудіо та відео була помітна неозброєним оком. Для цього відео було прийнято рішення про негайне коригування зсуву (приблизно 10 кадрів), не чекаючи на етап автоматичного аналізу синхронізації. Така стратегія дозволила уникнути накопичення помилок у наступних кроках обробки.

Третім кроком став розрахунок лендмарків за допомогою OpenFace, після чого проводилося кадрування відео для виділення області губ. П'ятим етапом стала сегментація відеоматеріалів на короткі фрагменти, що оптимізувало подальшу обробку та навчання моделей.

Завершальними етапами стали аналіз аудіо-візуальної синхронізації у всіх сегментах з використанням адаптованої версії SyncNet (Chung & Zisserman, 2016) та коригування виявлених зсувів. Варто зауважити, що незважаючи на відносну застарілість моделі SyncNet 2016 року, вона залишається ефективним інструментом для базової оцінки синхронності. Рішення про використання цієї моделі на завершальному етапі обробки було прийнято з урахуванням того, що ефективність підходу буде додатково оцінена в процесі навчання власних моделей.

Для сегментів з виявленою асинхронністю проводилася корекція зсуву, після чого виконувався повторний розрахунок лендмарків для оновлених відеоматеріалів. Такий комплексний підхід забезпечив створення високоякісного набору даних, оптимально підготовленого для навчання моделей аудіо-візуальної синхронізації. Та завдяки цьому можливо гнучке редагування обраних даних з використанням зсуву чи без.

3.1.3 Практична реалізація підготовки даних

Практична реалізація описаного методологічного підходу вимагала розробки спеціалізованих алгоритмів та врахування численних технічних

нюансів. Розглянемо ключові аспекти технічної імплементації, що забезпечили ефективну підготовку датасетів.

Перевірка цілісності відеофайлів була реалізована з використанням бібліотеки OpenCV для автоматичного виявлення нечитаних або пошкоджених файлів, рисунок 3.1.

Уніфікація формату даних реалізована через інтерфейс до бібліотеки FFmpeg, що забезпечило гнучкий підхід до стандартизації параметрів відео та аудіо. Розроблений модуль аналізував поточні характеристики кожного відеофайлу та обирав оптимальну стратегію конвертації до цільового формату з частотою 25 кадрів/с та частотою дискретизації аудіо 16 кГц.

```python
def check_video(video_path):
    """Перевірка цілісності відеофайлу."""
    try:
        # Спроба відкрити відеофайл
        video_capture = cv2.VideoCapture(video_path)
        if not video_capture.isOpened():
            return False

        # Спроба зчитування першого кадру
        success, _ = video_capture.read()
        if not success:
            video_capture.release()
            return False

        video_capture.release()
        return True
    except Exception:
        return False
```

Рисунок 3.1 – Код перевірки пошкоджений файлів

Особливу увагу було приділено алгоритму визначення області кадрування губ. Критичний аналіз оригінальної реалізації DINet виявив суттєвий недолік: нестабільність розміру області кадрування між різними кліпами одного відео. Оригінальний алгоритм обчислював окремий радіус кадрування для кожного сегмента з 9 послідовних кадрів, що призводило до коливань масштабу та погіршення якості результатів.

Для подолання цієї проблеми було розроблено удосконалений алгоритм обчислення статистики радіусів кадрування, рисунок 3.2.

Замість окремого радіуса для кожного сегмента, новий підхід аналізував усе відео та обчислював мінімальний, максимальний, середній та медіанний радіуси. Надалі для всього відео використовувався стабільний усереднений радіус, що забезпечувало постійний масштаб зображення області губ.

```python
def calculate_crop_radius_statistics(video_dimensions, facial_landmarks, sequence_length=16):
    """Розрахунок статистики радіусів кадрування для відеопослідовності."""
    crop_radii = []

    # Обчислення радіусу для кожного сегмента
    for frame_index in range(len(facial_landmarks) - sequence_length):
        landmarks_segment = facial_landmarks[frame_index:frame_index + sequence_length]
        if landmarks_segment.shape != (sequence_length, 68, 2):
            continue
        valid_crop, crop_radius = compute_crop_radius(video_dimensions, landmarks_segment)
        if valid_crop:
            crop_radii.append(crop_radius)

    if not crop_radii:
        raise ValueError("Не знайдено допустимих значень радіусу кадрування.")

    # Обчислення статистичних показників
    min_radius = min(crop_radii)
    max_radius = max(crop_radii)
    avg_radius = int(np.mean(crop_radii))
    median_radius = int(np.median(crop_radii))

    return min_radius, max_radius, avg_radius, median_radius
```

Рисунок 3.2 – Код обчислення радіусів кадрування

Коригування аудіо-візуальної синхронізації після виявлення зсуву реалізовано з використанням механізму часових зміщень FFmpeg, рисунок 3.3.

```python
def adjust_audio_video_sync(input_video_path, output_video_path, frame_offset, framerate=25):
    """Коригування зсуву аудіо відносно відео."""
    # Перетворення зсуву з кадрів у секунди
    time_offset = frame_offset / framerate

    # Формування та виконання команди FFmpeg
    command = (
        f"ffmpeg -loglevel error -y -i {input_video_path} "
        f"-itsoffset {time_offset} -i {input_video_path} "
        f"-map 0:v -map 1:a -c copy -q:v 0 -q:a 0 {output_video_path}"
    )
    subprocess.run(command, shell=True)
```

Рисунок 3.3 – Код застосування часових зміщень для синхронізації даних

Алгоритм кадрування кадрів використовував обчислений усереднений радіус та координати опорних точок обличчя, приклад використання можна побачити на рисунку 3.4.

```python
def crop_mouth_region(video_frame, facial_landmarks, crop_radius):
    """Кадрування області рота з використанням стабільного радіусу."""
    try:
        # Визначення центральної точки (ніс)
        if np.isnan(facial_landmarks).any() or not np.all(np.isfinite(facial_landmarks)):
            center_y = video_frame.shape[0] // 2
            center_x = video_frame.shape[1] // 2
        else:
            center_y = int(facial_landmarks[29][1])  # Y-координата кінчика носа
            center_x = int(facial_landmarks[33][0])  # X-координата кінчика носа

        # Обчислення координат області кадрування
        top = max(0, center_y - crop_radius)
        bottom = min(video_frame.shape[0], center_y + crop_radius * 2 + crop_radius // 4)
        left = max(0, center_x - crop_radius - crop_radius // 4)
        right = min(video_frame.shape[1], center_x + crop_radius + crop_radius // 4)

        # Вирізання області
        mouth_region = video_frame[top:bottom, left:right]

        # Стандартизація розміру з доповненням
        target_height = int(crop_radius * 3 + crop_radius // 4)
        target_width = int(crop_radius * 2 + crop_radius // 2)

        current_height, current_width = mouth_region.shape[:2]
        pad_top = (target_height - current_height) // 2
        pad_bottom = target_height - current_height - pad_top
        pad_left = (target_width - current_width) // 2
        pad_right = target_width - current_width - pad_left

        padded_region = cv2.copyMakeBorder(
            mouth_region,
            pad_top, pad_bottom, pad_left, pad_right,
            borderType=cv2.BORDER_CONSTANT, value=[0, 0, 0]
        )

        return padded_region
    except Exception:
        return None
```

Рисунок 3.4 – Код вирізання області голови з використанням радіусу

Сегментація відео реалізована з використанням FFmpeg через спеціалізований модуль, який враховував особливості датасету HDTF.

Алгоритм обчислював оптимальні точки розділення відео з урахуванням мінімальної допустимої тривалості останнього сегмента. Якщо залишковий фрагмент виявлявся коротшим за встановлений поріг (3 секунди), він об'єднувався з попереднім сегментом, що запобігало утворенню надто коротких фрагментів.

Впроваджені технічні рішення забезпечили високу якість підготовки даних з мінімальними втратами інформації. Особливо важливим удосконаленням стала стабілізація радіуса кадрування, що дозволило уникнути проблем з коливаннями масштабу при навчанні моделі DINet. Додатково було реалізовано механізм округлення радіусу до значення, кратного 4, для оптимізації обчислювальних операцій.

3.1.4 Структура завантажувача даних SyncNet

Для ефективного навчання моделі SyncNet необхідно забезпечити належну підготовку та обробку даних безпосередньо під час тренування. Розроблений завантажувач даних (dataloader) виконує низку критично важливих функцій, які суттєво впливають на якість навчання моделі. Основні завдання завантажувача: динамічна нормалізація вхідних зображень, афінне перетворення для вирівнювання області губ, обчислення та обробка мел-спектрограм для аудіоданих, формування позитивних і негативних пар «аудіо-відео» для навчання, а також оптимальне використання обчислювальних ресурсів через кешування та багатопоточну обробку.

Архітектурно завантажувач даних складається з трьох ключових класів: SyncNetDataLoader (основний координуючий клас), ImageProcessor (для обробки візуальних даних) та MelSyncNetDataset (реалізація PyTorch-датасету). Така модульна структура забезпечує гнучкість та розширюваність системи, дозволяючи незалежно модифікувати кожен компонент обробки даних.

Особливістю розробленого завантажувача є підтримка афінного перетворення для вирівнювання області губ. На відміну від простого вирізання фіксованої області, цей підхід враховує поворот голови у кадрі, що дозволяє отримати стандартизоване положення губ незалежно від ракурсу зйомки. Така нормалізація критично важлива для SyncNet, оскільки модель повинна зосередитися на русі губ, а не на загальному положенні обличчя.

Для реалізації афінного перетворення розроблено алгоритм, який обчислює кут нахилу на основі ключових точок обличчя, рисунок 3.5.

```python
def get_rotation_angles(self, video_path, start_idx, num_frames):
    try:
        video_basename = os.path.basename(video_path).replace(".mp4", "")
        landmarks_path = os.path.join(self.landmarks_dir, f"{video_basename}.npy")

        landmarks = np.load(landmarks_path)

        if landmarks.shape[0] < start_idx + num_frames:
            print(f"Not enough landmark frames for video {video_path}")
            return [None] * num_frames
        frame_landmarks = landmarks[start_idx:start_idx + num_frames]

        rotation_angles = []

        for i, lm in enumerate(frame_landmarks):
            pt_left_eyebrow = np.mean(lm[22:27], axis=0)
            pt_right_eyebrow = np.mean(lm[17:22], axis=0)
            pt_left_eye_outer = lm[36]
            pt_right_eye_outer = lm[45]
            pt_nose_tip = lm[30]

            angles = []
            angle_eyebrows = np.arctan2(
                pt_right_eyebrow[1] - pt_left_eyebrow[1],
                pt_right_eyebrow[0] - pt_left_eyebrow[0]
            )
            angles.append(angle_eyebrows)
            angle_eyes = np.arctan2(
                pt_right_eye_outer[1] - pt_left_eye_outer[1],
                pt_right_eye_outer[0] - pt_left_eye_outer[0]
            )
            angles.append(angle_eyes)
            angle_left_brow_nose = np.arctan2(
                pt_nose_tip[1] - pt_left_eyebrow[1],
                pt_nose_tip[0] - pt_left_eyebrow[0]
            )
            angles.append(angle_left_brow_nose)
            angle_right_brow_nose = np.arctan2(
                pt_right_eyebrow[1] - pt_nose_tip[1],
                pt_right_eyebrow[0] - pt_nose_tip[0]
            )
            angles.append(angle_right_brow_nose)
            angle_eyes_nose = np.arctan2(
                pt_right_eye_outer[1] - pt_left_eye_outer[1],
                pt_right_eye_outer[0] - pt_left_eye_outer[0]
            )
            angles.append(angle_eyes_nose)
            median_angle_rad = np.median(angles)
            angle_deg = -np.degrees(median_angle_rad)
            if video_basename not in self.rotation_angles:
                self.rotation_angles[video_basename] = {}
            self.rotation_angles[video_basename][start_idx + i] = angle_deg
            rotation_angles.append(angle_deg)
        return rotation_angles

    except Exception as e:
        print(f"Error in computing rotation angles: {e}")
        return [None] * num_frames
```

Рисунок 3.5 – Код обчислення куту нахилу для вирізання області рота

Використання п'яти різних опорних точок і обчислення медіанного кута значно підвищує стійкість алгоритму до шуму в даних ландмарків, забезпечуючи більш надійне вирівнювання навіть при недосконалому

розпізнаванні обличчя. Після обчислення кута застосовується власне трансформація до області губ, рисунок 3.6.

Ця реалізація використовує функцію grid_sample з бібліотеки PyTorch, яка дозволяє виконувати довільні просторові перетворення з білінійною інтерполяцією. Такий підхід забезпечує плавне обертання без артефактів, що критично важливо для збереження дрібних деталей руху губ.

Після трансформації виконується нормалізація пікселів до діапазону $[-1, 1]$, що відповідає вимогам SyncNet і підвищує стабільність навчання. Для цього застосовується стандартна трансформація з нормалізацією до значень в діапазоні від -0.5 до 0.5.

```
def crop_mouth_with_rotation(self, frame, angle):
    _, _, height, width = frame.shape

    y_start = self.radius
    y_end = self.radius + self.mouth_region_size
    x_start = self.radius_1_4
    x_end = self.radius_1_4 + self.mouth_region_size

    center_y = (y_start + y_end) / 2
    center_x = (x_start + x_end) / 2

    angle_rad = math.radians(angle)

    grid_h, grid_w = self.mouth_region_size, self.mouth_region_size

    y_grid, x_grid = torch.meshgrid(
        torch.linspace(0, grid_h-1, grid_h),
        torch.linspace(0, grid_w-1, grid_w),
        indexing='ij'
    )
    y_grid = y_grid - (grid_h-1)/2
    x_grid = x_grid - (grid_w-1)/2
    cos_theta = torch.cos(torch.tensor(angle_rad))
    sin_theta = torch.sin(torch.tensor(angle_rad))

    x_rot = x_grid * cos_theta + y_grid * sin_theta
    y_rot = -x_grid * sin_theta + y_grid * cos_theta
    x_rot = x_rot + center_x
    y_rot = y_rot + center_y

    x_norm = 2.0 * x_rot / (width - 1) - 1.0
    y_norm = 2.0 * y_rot / (height - 1) - 1.0

    grid = torch.stack([x_norm, y_norm], dim=-1).unsqueeze(0)

    mouth_region = F.grid_sample(
        frame.float(),
        grid,
        mode='bilinear',
        padding_mode='zeros',
        align_corners=True
    )
    if frame.dtype != torch.float32:
        mouth_region = mouth_region.to(frame.dtype)

    return mouth_region
```

Рисунок 3.6 – Код вирізання області рота з урахуванням знайдених кутів

Важливо зазначити, що нормалізація застосовується після масштабування значень пікселів до діапазону [0, 1], що забезпечує коректне відображення в цільовий діапазон.

Для обробки аудіоданих та створення мел-спектрограм було реалізовано спеціалізований модуль, який використовує бібліотеку torchaudio, рисунок 3.7.

Обробка аудіо включає декілька ключових етапів. Спочатку застосовується передпідсилення (pre-emphasis), що підвищує амплітуду високочастотних компонентів.

```
def _init_audio_processors(self):
    # Pre-emphasis coefficient
    self.preemphasis_coef = 0.97

    # Create mel spectrogram transformer
    self.mel_spectrogram = torchaudio.transforms.MelSpectrogram(
        sample_rate=self.audio_sample_rate,
        n_fft=800,  # FFT window size
        win_length=800,  # Window size
        hop_length=200,  # Hop size
        f_min=55,  # Minimum frequency
        f_max=7600,  # Maximum frequency
        n_mels=80,  # Number of mel filters
        power=1.0,  # Amplitude spectrum instead of power spectrum (sqrt)
        normalized=False,
        center=True,
        pad_mode="reflect",
        onesided=True,
        norm="slaney",  # Mel bank normalization
        mel_scale="slaney"  # Mel scale type
    )

    # Parameters for normalization
    self.min_level_db = -100
    self.ref_level_db = 20
    self.max_abs_value = 4.0
    self.symmetric_mels = True
    self.signal_normalization = True
    self.allow_clipping_in_normalization = True
```

Рисунок 3.7 – Код розрахунку мел-спектрограми

Після обчислення мел-спектрограми виконується перетворення амплітуди у децибели та нормалізація. Нормалізація мел-спектрограми має принципове значення для стабільності навчання SyncNet, оскільки

забезпечує узгоджений діапазон значень для різних аудіоматеріалів, незалежно від гучності оригінального запису.

Для роботи з відеофрагментами розроблено спеціалізовану функцію вирізання часового вікна з мел-спектрограми, з урахуванням всіх необхідних параметрів синхронізації з відео, рисунок 3.8.

Ключовим аспектом навчання SyncNet є формування позитивних (синхронізованих) та негативних (несинхронізованих) пар «аудіо-відео». Для кожного елемента датасету випадковим чином обирається, чи повертати коректну чи некоректну пару. Також для більш гнучкого використання, а саме тренування DINet, було розроблено додаткові параметри, що дозволять повертати кути нахилу для нормалізації і обробляти вже згенеровані дані, це потрібно для того, щоб можна було без проблем використовувати найкращу версію SyncNet.

```python
def crop_audio_window(self, original_mel, start_index):
    """Вирізання вікна мел-спектрограми, що відповідає фрагменту відео"""
    hop_size = 200

    # Обчислення індексів початку та кінця
    start_idx = int((self.audio_sample_rate / hop_size) * (start_index / float(self.video_fps)))
    end_idx = start_idx + self.mel_window_length

    # Перевірка чи вистачає даних
    if end_idx > original_mel.shape[1]:
        return None

    return original_mel[:, start_idx:end_idx].unsqueeze(0)
```

Рисунок 3.8 – Код вирізання необхідної частини мел-спектрограми

Розроблений завантажувач даних забезпечує надійну та ефективну підготовку тренувальних даних для моделі SyncNet, враховуючи специфічні вимоги до аудіо-візуальної синхронізації та оптимізуючи використання обчислювальних ресурсів.

3.2 Розробка та аналіз модифікованої моделі SyncNet

3.2.1 Реалізація модифікованої моделі SyncNet

На основі теоретичних засад, описаних у розділі 2.3, було розроблено практичну реалізацію модифікованої моделі SyncNet для аудіо-візуальної синхронізації, яку назвемо AVAlignNet. Модель реалізована з використанням фреймворку PyTorch, що забезпечує ефективне виконання тензорних операцій на графічних процесорах та спрощує процес глибокого навчання.

Архітектура реалізованої моделі відповідає концепції, запропонованій у попередньому розділі, із суттєвими модернізаціями порівняно з оригінальною версією SyncNet. Технічна реалізація моделі використовує бібліотеку einops для зручних перестановок і перетворень форми тензорів, а також компоненти з бібліотеки diffusers, які забезпечують реалізацію передових механізмів уваги та шарів прямого проходження. Основний клас моделі представлено на рисунку 3.9.

Головною структурною одиницею енкодерів є блок DownEncoder2D, який реалізує принципи згорткових шарів та субдискретизації, описані в розділі 2.3.1. Для програмної реалізації згорткових операцій використовуються модулі torch.nn.Conv2d з налаштовуваними параметрами kernel_size, stride та padding. Блок послідовно зменшує просторову розмірність вхідних даних через операції nn.ModuleList, що містить каскад резидуальних блоків з можливістю керованого пониження дискретизації через параметр downsample_factors.

Кожен резидуальний блок реалізований через клас ResnetBlock2D, який базується на архітектурі ResNet [25], але адаптований для специфічних потреб аудіо-візуальної синхронізації. Програмна реалізація включає два послідовні згорткові шари з використанням torch.nn.GroupNorm для нормалізації та torch.nn.SiLU для активації. Механізм skip-connections

реалізований через пряме додавання тензорів (+=), що забезпечує стабільність градієнтів під час навчання глибокої мережі.

```python
class AVAlignNet(nn.Module):
    Tabnine | Edit | Test | Explain | Document
    def __init__(self, config, gradient_checkpointing=False):
        super().__init__()
        self.audio_encoder = DownEncoder2D(
            in_channels=config["audio_encoder"]["in_channels"],
            block_out_channels=config["audio_encoder"]["block_out_channels"],
            downsample_factors=config["audio_encoder"]["downsample_factors"],
            dropout=config["audio_encoder"]["dropout"],
            attn_blocks=config["audio_encoder"]["attn_blocks"],
            gradient_checkpointing=gradient_checkpointing,
        )

        self.visual_encoder = DownEncoder2D(
            in_channels=config["visual_encoder"]["in_channels"],
            block_out_channels=config["visual_encoder"]["block_out_channels"],
            downsample_factors=config["visual_encoder"]["downsample_factors"],
            dropout=config["visual_encoder"]["dropout"],
            attn_blocks=config["visual_encoder"]["attn_blocks"],
            gradient_checkpointing=gradient_checkpointing,
        )

        self.eval()

    Tabnine | Edit | Test | Explain | Document
    def forward(self, image_sequences, audio_sequences):
        vision_embeds = self.visual_encoder(image_sequences)  # (b, c, 1, 1)
        audio_embeds = self.audio_encoder(audio_sequences)  # (b, c, 1, 1)

        vision_embeds = vision_embeds.reshape(vision_embeds.shape[0], -1)  # (b, c)
        audio_embeds = audio_embeds.reshape(audio_embeds.shape[0], -1)  # (b, c)

        # Make them unit vectors
        vision_embeds = F.normalize(vision_embeds, p=2, dim=1)
        audio_embeds = F.normalize(audio_embeds, p=2, dim=1)

        return vision_embeds, audio_embeds
```

Рисунок 3.9 – Основний клас пропонованої AVAlignNet моделі

Блоки самоуваги, теоретичні основи яких викладено в розділі 2.3.2, реалізовані через спеціалізовані модулі AttentionBlock2D з використанням torch.nn.MultiheadAttention. Матриці запитів, ключів та значень генеруються через nn.Linear перетворення вхідних ознак. Ключовою особливістю реалізації є використання мультиголової уваги з 8 головами через параметр num_heads=8, що дозволяє моделі одночасно фокусуватися на різних аспектах просторових залежностей.

Відповідно до принципів нормалізації та активації з розділу 2.3.3, реалізація включає диференційовану систему нормалізації через

torch.nn.GroupNorm з параметром num_groups=32 для згорткових блоків та torch.nn.LayerNorm для блоків уваги. Функції активації реалізовані через torch.nn.SiLU як основну активацію для прихованих шарів, що показала кращу продуктивність порівняно з традиційною ReLU в контексті аудіо-візуальної синхронізації через плавнішу форму функції.

Процес формування векторів ознак реалізований згідно з принципами косинусної подібності з розділу 2.3.5. Обидва енкодери продукують векторні представлення через nn.Linear шари, які нормалізуються до одиничної довжини за допомогою torch.nn.functional.normalize з параметром p=2, dim=1. Це дозволяє використовувати результати для розрахунку косинусної подібності. Практична реалізація включає адаптивне масштабування розмірності векторів ознак через конфігураційні параметри, що дозволяє легко переходити між 512, 1024 або 2048 розмірностями.

При розробці особлива увага приділялася ефективності обчислень, впроваджено mixed precision training з використанням torch.cuda.amp для прискорення навчання на сучасних GPU. Архітектура спроєктована з урахуванням принципів збалансованого масштабування через параметричну конфігурацію block_out_channels та downsample_factors, що забезпечує оптимальне співвідношення точності та обчислювальної ефективності при різних конфігураціях ресурсів.

3.2.2 Аналіз процесу навчання AVAlignNet

Ефективність навчання моделі AVAlignNet критично залежить від оптимального використання доступних обчислювальних ресурсів. Експериментальні дослідження проводилися на робочій станції, оснащеній графічним прискорювачем NVIDIA GeForce RTX 4070 Ti Super (16 ГБ відеопам'яті) та процесором Intel i5-13600KF. Враховуючи обмеження відеопам'яті, особливу увагу було приділено визначенню оптимальних гіперпараметрів навчання для досягнення максимальної ефективності.

Розмір навчального пакету (batch size) є одним із найвпливовіших гіперпараметрів навчання нейронних мереж, який безпосередньо визначає швидкість конвергенції та стабільність процесу оптимізації. Для комплексного дослідження цього впливу проведено серію експериментів з варіативними значеннями batch size: 128, 256 та 512 на датасеті HDTF з повним циклом обробки даних, включаючи коригування аудіо-візуальної синхронізації.

Результати, представлені на рисунку 3.10, ілюструють суттєвий вплив розміру міні-пакету на динаміку оптимізації функції втрат. Модель, навчена з batch size 512, характеризується найдинамічнішим зниженням функції втрат, досягаючи мінімального значення 0,37 після 5000 ітерацій навчання. Конфігурація з batch size 256 демонструє помірнішу швидкість збіжності зі стабілізацією на рівні 0,41. Найповільніше навчається модель з batch size 128, де значення функції втрат стабілізується близько 0,46. Усі досліджувані моделі мали ідентичну архітектуру з 7 блоками DownEncoder2D, де кількість каналів ознак в послідовних блоках аудіоенкодера зростала від 32 до 2048, а у візуальному енкодері – від 64 до 2048. Фактори пониження розмірності підбиралися згідно з принципами EfficientNet, а просторова роздільна здатність області губ становила 64×64 пікселі.

Аналіз отриманих даних свідчить, що збільшення розміру міні-пакету забезпечує стабільніші оцінки градієнта, що відображається у зменшенні варіативності функції втрат. Це особливо помітно за значно вужчою областю флуктуацій навколо кривих для більших значень batch size. Характерним для всіх конфігурацій є виразний перехід від інтенсивного спаду втрат на початкових 1000 ітераціях до поступового уповільнення прогресу на подальших етапах.

Варто відзначити нелінійний характер залежності якості навчання від розміру міні-пакету: збільшення з 256 до 512 призводить до суттєвішого приросту ефективності порівняно з переходом від 128 до 256. Більший batch

size також забезпечує оптимальніше використання паралельних обчислень GPU, що відображається у вищій швидкості обробки даних (кількості ітерацій за одиницю часу).

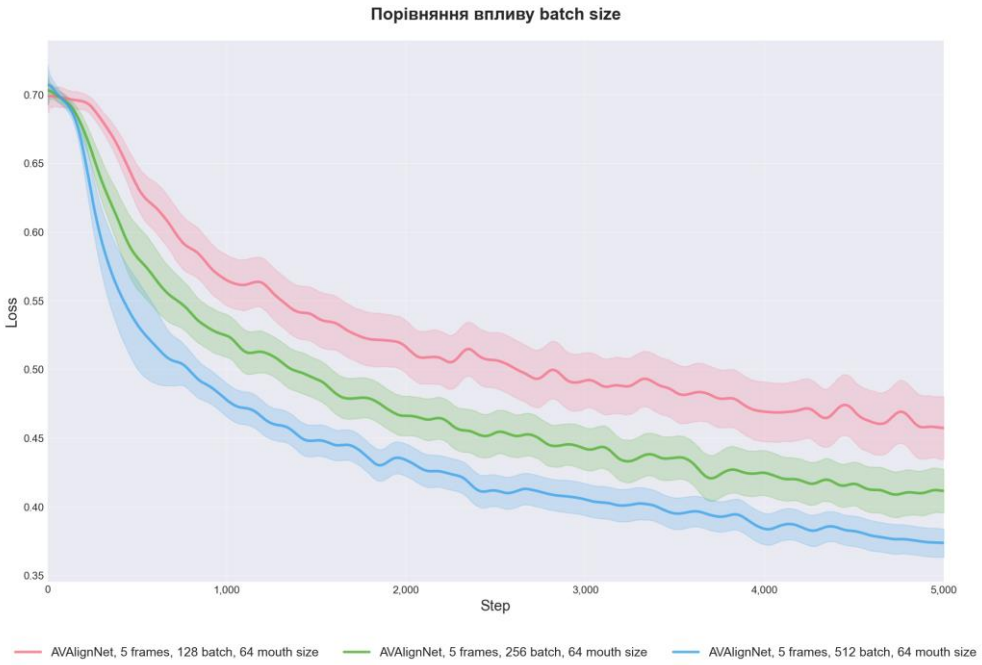

Рисунок 3.10 – Графік порівняння впливу batch size

Необхідно підкреслити, що навчання проводилося виключно на англомовних даних. При використанні мультимовного датасету навчання з малим batch size (128) імовірно не розпочалося б через неможливість узагальнення закономірностей при високій гетерогенності даних. Навіть з batch size 256 збіжність потребувала б значно більшої кількості ітерацій.

Розмір часового вікна – кількість послідовних кадрів для одночасної обробки – є критичним параметром для розуміння динамічних аспектів артикуляції. Рисунок 3.11 ілюструє порівняння ефективності навчання AVAlignNet з варіативною кількістю кадрів: 5, 10 та 15, при фіксованому batch size 512.

Аналіз результатів свідчить про позитивний вплив розширення часового контексту на якість навчання. Модель з 15-кадровим вікном

досягла найнижчого значення функції втрат (0,30) після 5000 ітерацій. Конфігурація з 10 кадрами характеризувалася значенням функції втрат 0,33, тоді як 5-кадрове вікно показало значення функції втрат 0,37.

Розширення часового вікна потребувало адаптації архітектури аудіоенкодера відповідно до зміни розмірності мел-спектрограми: (1, 80, 16) для 5 кадрів, (1, 80, 32) для 10 кадрів і (1, 80, 48) для 15 кадрів. Це вимагало корекції факторів пониження розмірності в аудіоенкодері при збереженні інших архітектурних параметрів.

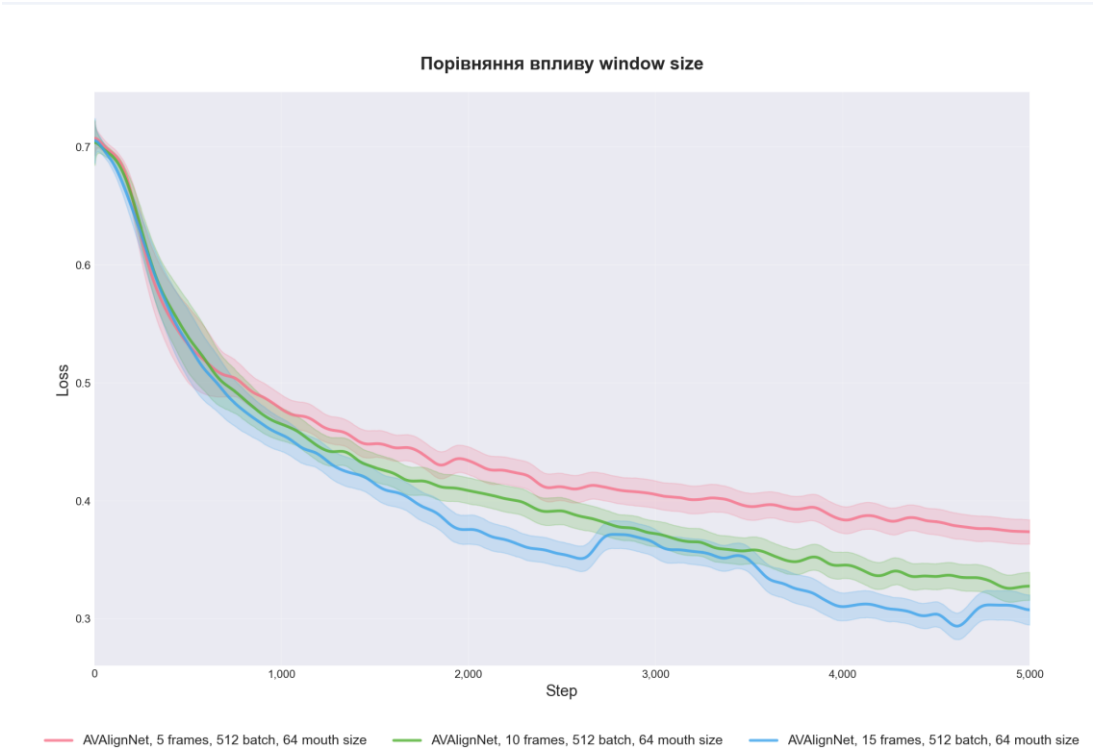

Рисунок 3.11 – Графік порівняння впливу window size

Дане дослідження розширює попередні роботи Wav2Lip, де аналізувалися менші часові вікна (1, 3, 5 кадрів), підтверджуючи, що збільшення часового контексту продовжує покращувати точність синхронізації навіть при значно більших інтервалах. Проте спостерігається зменшення граничної корисності: покращення при переході від 10 до 15 кадрів є менш вираженим, ніж від 5 до 10.

Розширений часовий контекст дозволяє моделі ефективніше вловлювати динамічні патерни артикуляції та коартикуляційні ефекти – явища взаємного впливу сусідніх звуків на артикуляцію. При 15 кадрах (приблизно 0,6 секунди відео) модель отримує достатній контекст для точнішого розрізнення складних фонетичних послідовностей.

Просторова роздільна здатність області губ безпосередньо впливає на здатність моделі розрізняти тонкі артикуляційні нюанси. Рисунок 3.12 представляє порівняння конфігурацій з розмірами області губ 64×64 та 128×128 пікселів при фіксованих параметрах: часове вікно 5 кадрів, batch size 256.

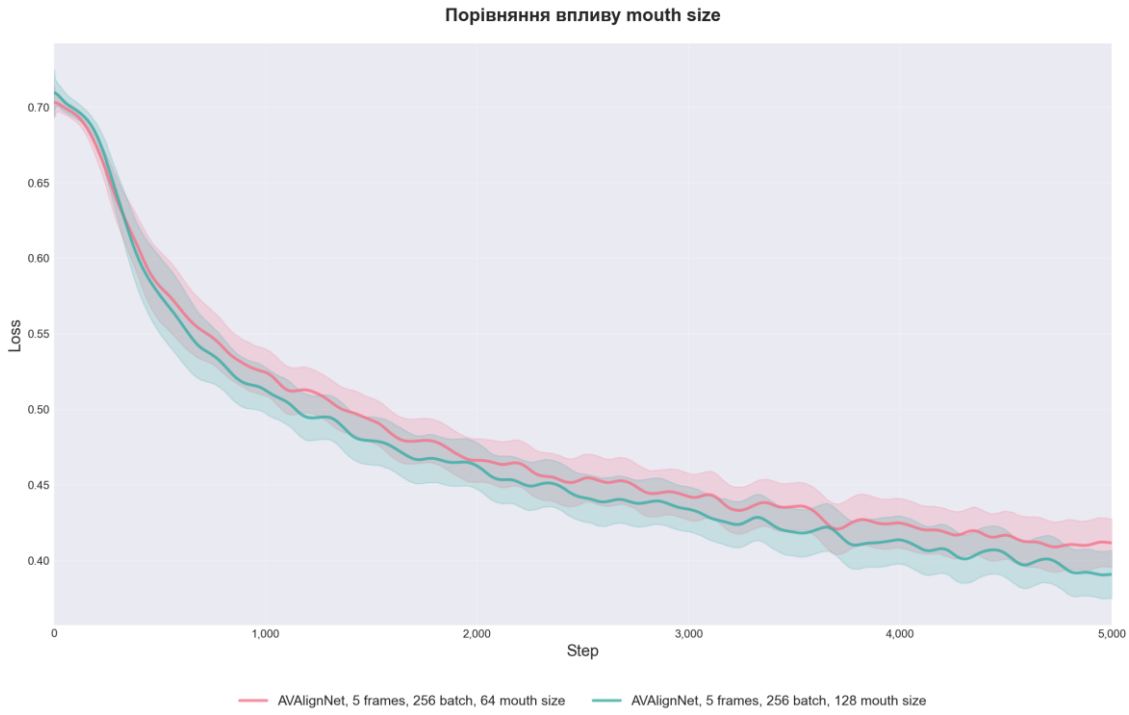

Рисунок 3.12 – Графік порівняння впливу mouth size

AVAlignNet з роздільною здатністю 128×128 пікселів продемонструвала кращі результати, досягнувши значення функції втрат 0,38 після 5000 ітерацій, тоді як для конфігурації з роздільною здатністю 64×64 цей показник склав 0,41. Для забезпечення коректної

роботи з різними розмірами було адаптовано архітектуру візуального енкодера: фактор пониження для шару з 256 ознаками становив 2 для 64×64 і 1 для 128×128, забезпечуючи ідентичну вихідну форму тензора.

Підвищена роздільна здатність надає доступ до дрібніших деталей артикуляції: незначних рухів кутиків губ, мікро-зморшок навколо рота, форми внутрішньої частини губ. Ці деталі містять цінну інформацію для розрізнення фонетично схожих елементів, які відрізняються лише тонкими артикуляційними особливостями.

Варто відзначити, що різниця в продуктивності стає помітною після 2000 ітерацій, що свідчить про важливість вищої роздільної здатності саме на етапі тонкого налаштування моделі для виявлення складніших закономірностей. Слід зазначити, що оригінальна модель Wav2Lip SyncNet навчалася на роздільній здатності 256×256, що через ресурсні обмеження поточної конфігурації може обмежувати досяжну точність.

Експеримент з використанням латентних представлень замість прямої піксельної обробки представлено на рисунку 3.13. Порівнювалися традиційний піксельний підхід (при batch size 256) та метод на основі латентного простору (при batch size 512) з використанням автоенкодера Stable Diffusion 2 VAE.

Вхідні дані розміром 15×128×128 (5 кадрів, роздільна здатність 128×128) перетворювалися у латентні вектори 20×16×16. Через обмеження відеопам'яті неможливо було експериментувати з латентними представленнями 20×32×32 з вхідних зображень 256×256.

Піксельний підхід демонструє чітку перевагу, досягаючи значення функції втрат 0,38 після 5000 ітерацій, тоді як латентний метод обмежується значенням 0,42, навіть при більшому batch size. Це свідчить про втрату критично важливих просторових деталей артикуляції при компресії даних автоенкодером. Хоча латентні представлення ефективні для генеративних завдань, вони недостатньо точно зберігають тонкі особливості руху губ, необхідні для якісної аудіо-візуальної синхронізації.

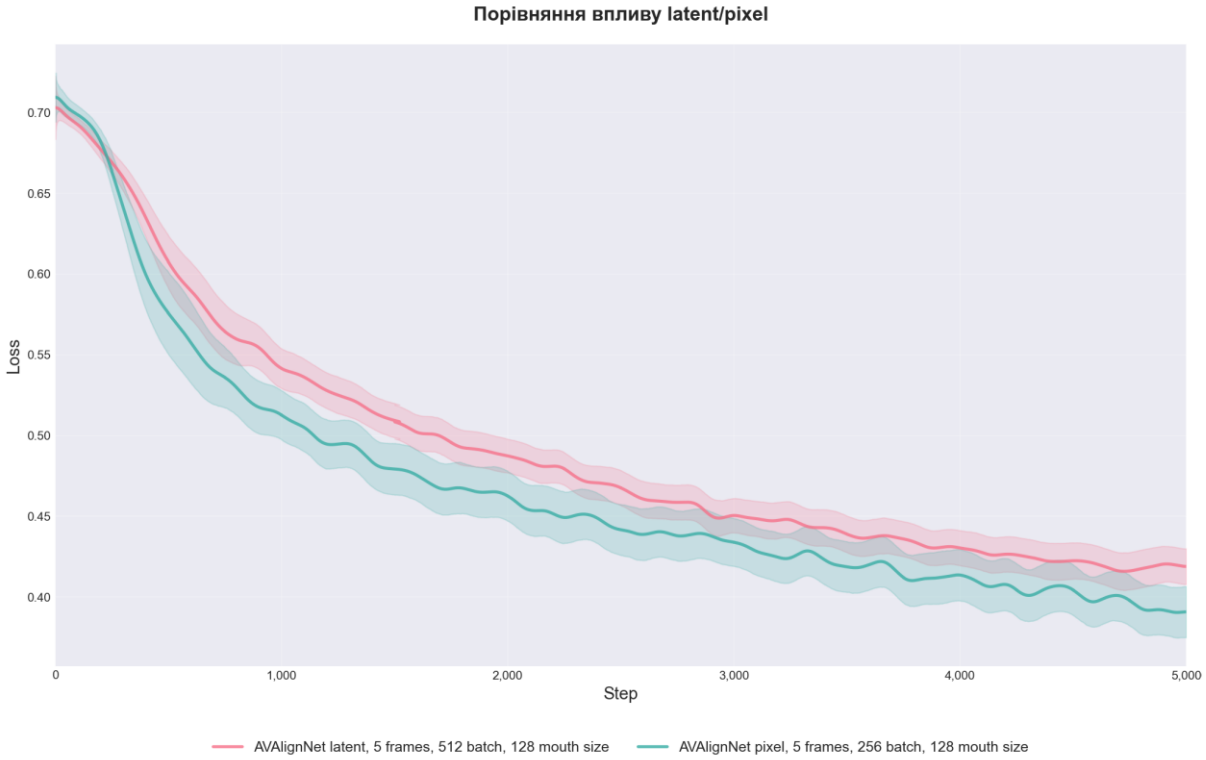

Рисунок 3.13 – Графік порівняння впливу latent/pixel

Рисунок 3.14 представляє пряме порівняння AVAlignNet з оригінальною реалізацією SyncNet (компонента Wav2Lip) при ідентичному розмірі області губ (128×128) і кількості кадрів (5).

Протягом усього процесу навчання, AVAlignNet стабільно демонструє нижчі значення функції втрат порівняно з SyncNet. Зокрема, після 3000 ітерацій значення функції втрат для AVAlignNet становить близько 0,39–0,40, тоді як для SyncNet – приблизно 0,42–0,43. Наприкінці спостережуваного періоду навчання функція втрат для AVAlignNet досягає рівня близько 0,34–0,35, в той час як для SyncNet вона залишається на рівні 0,39–0,40. Це свідчить не лише про значно кращу початкову ефективність AVAlignNet, але й про її перевагу в досягнутій якості моделі за однакової кількості ітерацій та ідентичних налаштувань, представлених на даному графіку.

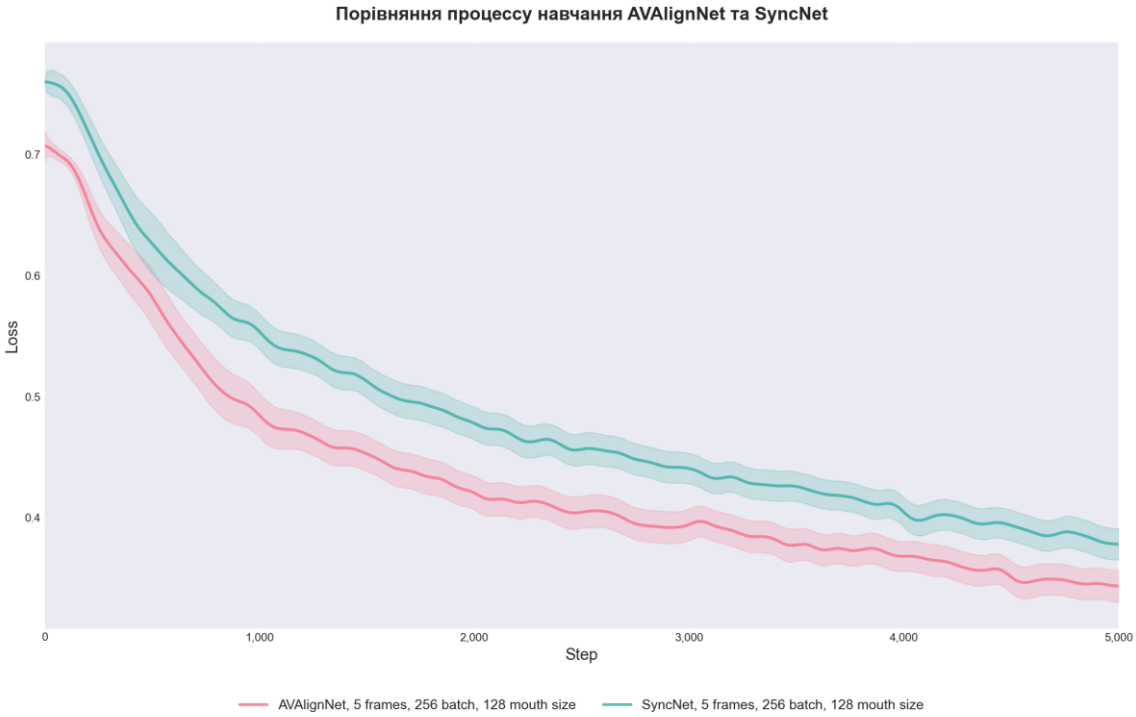

Рисунок 3.14 – Графік порівняння AVAlignNet та SyncNet

Рисунок 3.15 демонструє ефект включення блоків уваги в архітектуру AVAlignNet. Порівнювалися варіанти архітектури: без використання механізму уваги та з його включенням, при ідентичних параметрах: 5 кадрів, batch size 512, область губ 64×64.

На початкових етапах обидві конфігурації показують подібні результати, проте перевага механізму уваги проявляється на етапі тонкого налаштування. Після 4000 ітерацій модель, що використовує механізм уваги, досягла значення функції втрат 0,32, тоді як модель без нього – 0,34. Хоча різниця у значеннях функції втрат невелика, вона спостерігалася стабільно, що вказує на її потенційну значущість.

Механізм уваги дозволяє встановлювати зв'язки між просторово та часово віддаленими елементами, що критично важливо для розпізнавання складних фонетичних патернів і коартикуляційних ефектів. Враховуючи незначне збільшення обчислювальних витрат (за оцінками, 5-10%),

включення механізму уваги є обґрунтованим рішенням для покращення якості моделі.

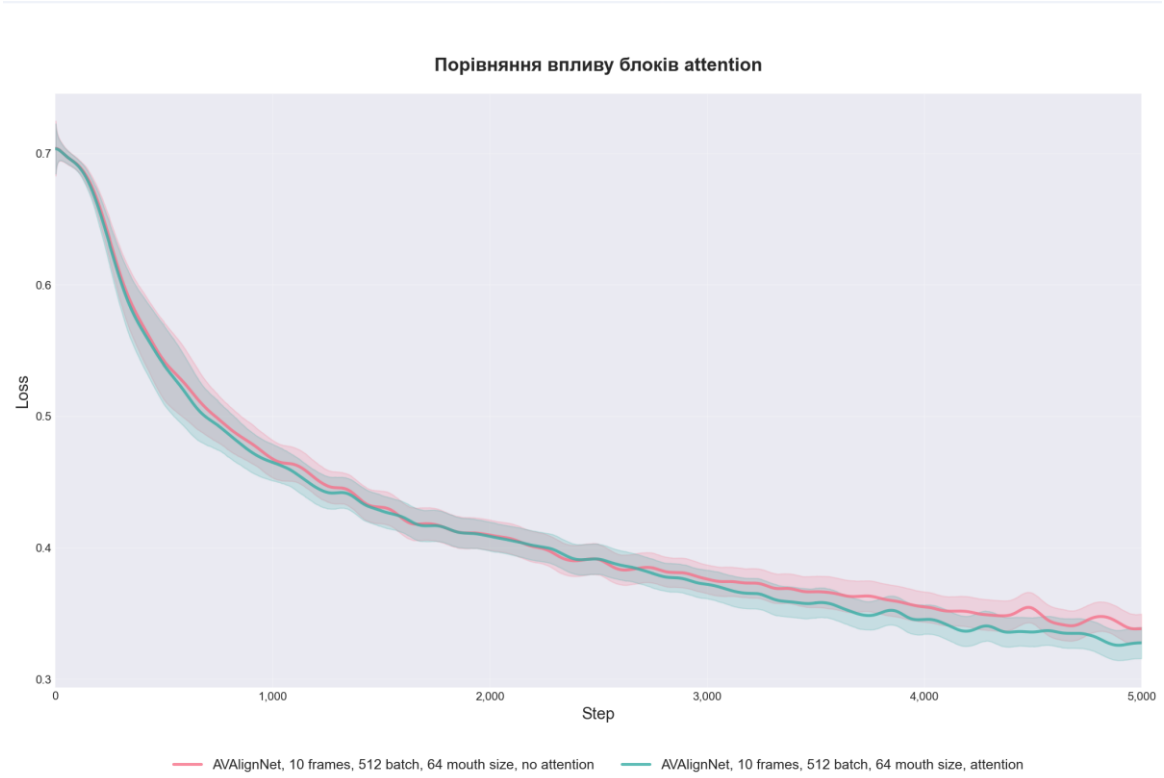

Рисунок 3.15 – Графік порівняння впливу блоків attention

Останній експеримент, рисунок 3.16, досліджував вплив максимальної кількості каналів ознак на ефективність AVAlignNet. Порівнювалися конфігурації з 512, 1024 та 2048 максимальними каналами ознак при збереженні глибини мережі.

Модель з 2048 максимальними каналами ознак досягла найнижчого значення функції втрат (0,32), за нею слідували конфігурації з 1024 (значення втрат 0,33–0,34) та 512 ознаками (значення втрат 0,35). Різниці стають помітними після 3000 ітерацій, підкреслюючи важливість високої ємності моделі для тонкого налаштування та виявлення складних закономірностей.

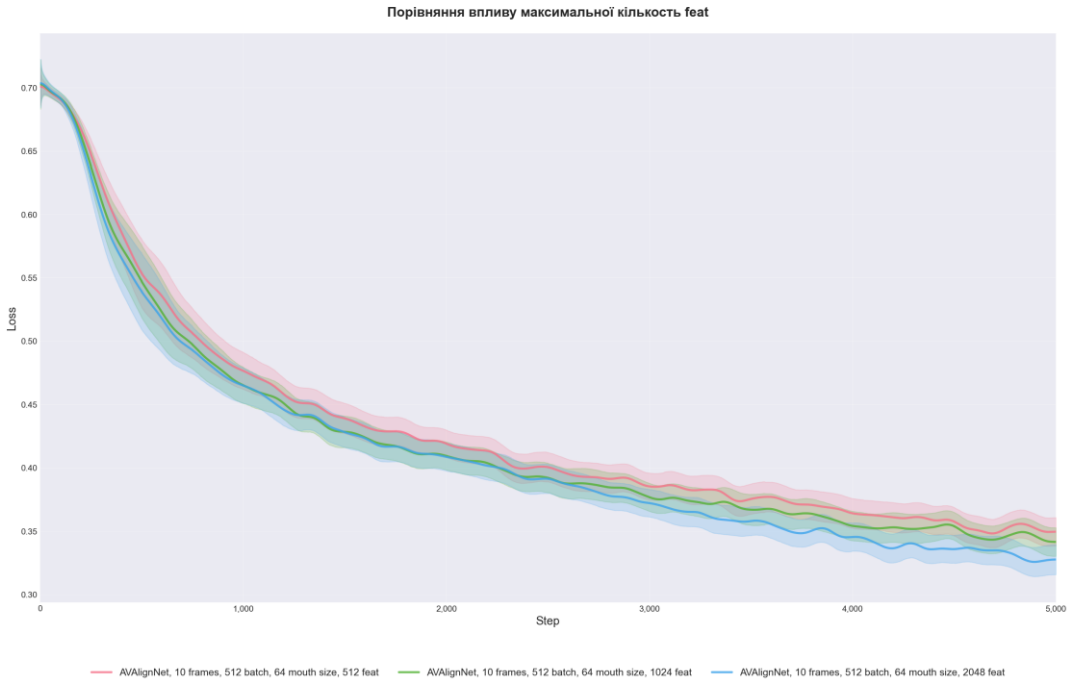

Рисунок 3.16 – Графік порівняння впливу кількості feat

Результати узгоджуються з принципами глибокого навчання, де більша кількість параметрів, як правило, забезпечує вищу ємність моделі. Проте збільшення кількості ознак вимагає додаткових обчислювальних ресурсів, тому оптимальна конфігурація повинна враховувати наявні можливості та вимоги до швидкодії.

3.2.3 Фінальні результати та порівняльний аналіз AVAlignNet

Для об'єктивного оцінювання ефективності розробленої архітектури AVAlignNet проведено детальне порівняння з оригінальною реалізацією SyncNet (компонента Wav2Lip). Обидві моделі навчалися протягом 30000 ітерацій з ідентичними налаштуваннями: швидкість навчання $1 \cdot 10^{-5}$, оптимізатор AdamW, розмір міні-пакету 256, розмір області губ 128×128 пікселів.

Рисунок 3.17 ілюструє перевагу моделі AVAlignNet протягом усього процесу навчання.

Кінцеві результати демонструють суттєву різницю у значенні функції втрат: AVAlignNet досягає 0,21, тоді як оригінальна SyncNet – 0,26. Це відповідає відносному покращенню приблизно на 19%. Динаміка навчання також висвітлює архітектурні переваги: AVAlignNet показує стабільну збіжність.

Валідаційні результати підтверджують вищу ефективність нової архітектури: AVAlignNet досягла точності 92,73% проти 90,44% у оригінальної моделі. Ці показники набувають особливого значення при зіставленні з відомими результатами оригінальної Wav2Lip SyncNet на датасеті LRS2 (91,6% точності).

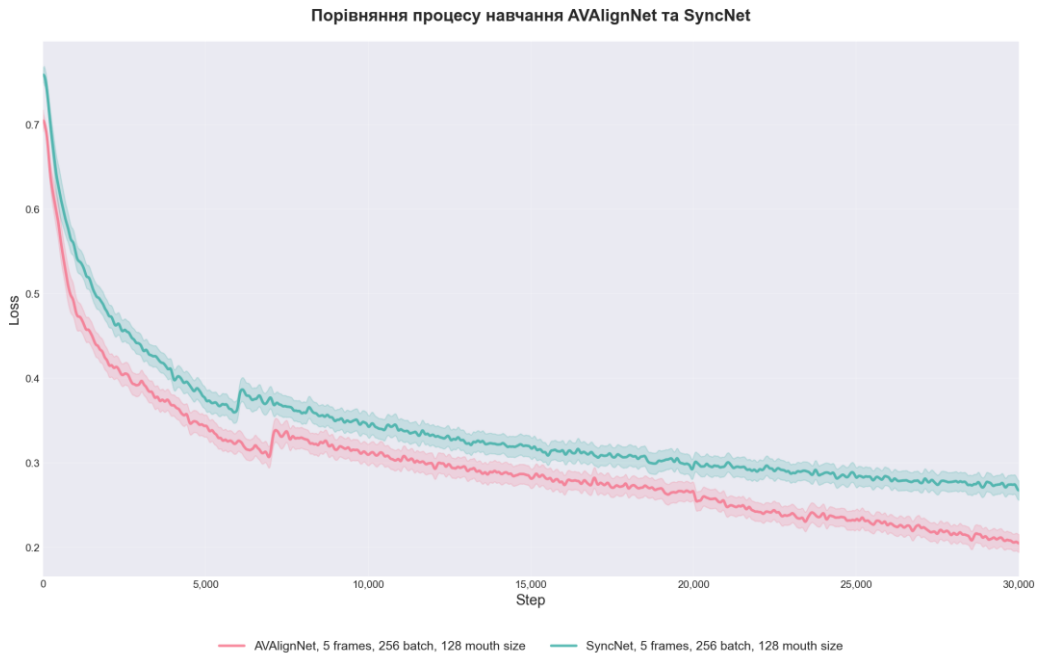

Рисунок 3.17 – Графік порівняння AVAlignNet та SyncNet

Важливо підкреслити контекст цих результатів. Оригінальна модель Wav2Lip навчалася на датасеті LRS2 – значно різноманітнішому наборі даних, що включає множину дикторів, мов та умов зйомки. Адаптована версія SyncNet, навчена на датасеті HDTF, показала точність 90,44%, що лише на 1,16% поступається оригінальному результату на LRS2,

підтверджуючи адекватність адаптації моделі до нового датасету без суттєвих втрат узагальнювальної здатності.

На цьому тлі результат AVAlignNet у 92,73% на HDTF є переконливим, оскільки перевага досягнута на ідентичних навчальних даних, демонструючи ефективність запропонованих архітектурних удосконалень. Варто зазначити, що порівняння проводилося з версією оригінальної SyncNet, адаптованою (поглибленою) для обробки збільшеного обсягу даних датасету HDTF, проте навіть це не дозволило їй досягти рівня продуктивності AVAlignNet.

Ефективність розробленої системи попередньої обробки даних ілюструють рисунок 3.18, з прикладами кадрів датасету Hallo3. На зображенні представлені оригінальні кадри з розрахованими кутами нахилу голови та результати афінного перетворення, застосованого для вирівнювання області губ.

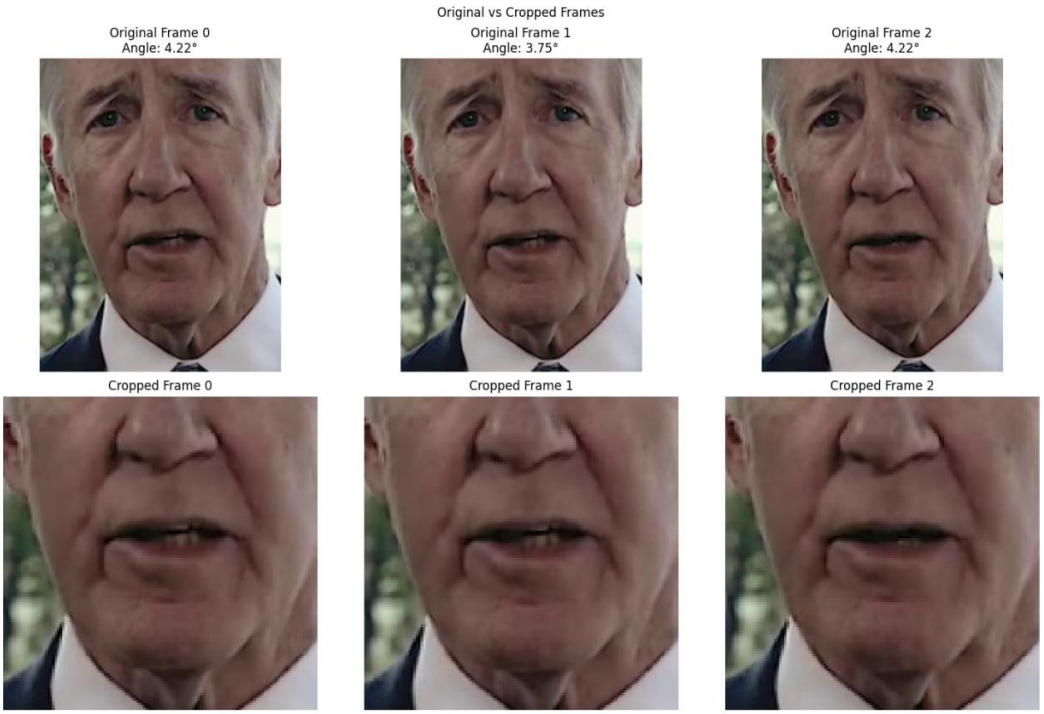

Рисунок 3.18 – Візуалізація даних з Hallo3 після завантаження

Наданий приклад демонструє типові коливання положення голови з кутами нахилу в діапазоні 3,75–4,22. Система ефективно компенсує ці відхилення, забезпечуючи стандартизоване горизонтальне положення губ.

Однак тестування на датасеті HALLO3 виявило значні труднощі для обох моделей: AVAlignNet показала точність 64,70%, а Wav2Lip SyncNet – 63,42%. Такий різкий спад продуктивності порівняно з результатами на HDTF потребує ретельного аналізу причин та їхнього впливу на практичне застосування розроблених моделей.

Фундаментальною причиною зниження точності є кардинальна відмінність характеристик датасету HALLO3 від HDTF. HALLO3 створений на основі реальних медіа-матеріалів з кінофільмів, телепрограм та іншого різноманітного контенту.

Особливо показовим є той факт, що навіть оригінальна Wav2Lip SyncNet, навчена на різноманітному датасеті LRS2, демонструє на подібних «in-the-wild» даних точність лише 63,87% – результат, практично ідентичний отриманому нами для адаптованої SyncNet на HALLO3 (63,42%).

Отримані результати на HALLO3 висвітлюють фундаментальну проблему узагальнювальної здатності в машинному навчанні. Моделі, що демонструють високу продуктивність на контрольованих навчальних даних, можуть значно втрачати якість в умовах реального світу. Це є особливо критичним для задач комп'ютерного зору, де варіативність візуальних умов надзвичайно висока.

Важливо відзначити, що AVAlignNet зберігає свою перевагу над SyncNet навіть у складних умовах датасету HALLO3 (64,70% проти 63,42%). Хоча абсолютна різниця в точності невелика (1,28%), це може свідчити про кращу стійкість архітектури AVAlignNet до розподілу даних, що суттєво відрізняється від навчального. Ця властивість може бути особливо цінною для практичних застосувань, де експлуатаційні умови рідко відповідають ідеалізованим навчальним сценаріям.

Проведені дослідження та отримані результати окреслюють кілька критичних напрямків для подальших наукових пошуків. По-перше, існує нагальна необхідність у розробці більш масштабних та різноманітних навчальних датасетів, які б краще відображали умови реального світу, як Hallo3 але в 2К або 4К якості. По-друге, важливим є розвиток та вдосконалення методів доменної адаптації та технік навчання, спрямованих на підвищення стійкості моделей до змін у характеристиках вхідних даних. По-третє, актуальною залишається потреба в розробці таких архітектурних рішень для нейронних мереж, які б були природно більш стійкими до значної варіативності візуальних умов.

3.3 Навчання та результати модифікованої DINet

3.3.1 Адаптація завантажувача даних та процес навчання DINet

Для ефективного навчання модифікованої моделі DINet ключовим аспектом стала адаптація завантажувача даних (Dataloader). Базуючись на принципах, описаних у розділі 3.1.4, завантажувач даних було модифіковано для роботи з DINet, зокрема для забезпечення можливості обробки вже згенерованих кадрів. Однією з важливих доробок стала можливість повернення кутів нахилу голови, розрахованих на основі ключових точок обличчя. Це дозволяє коректно вирізати та нормалізувати область рота зі згенерованих моделлю DINet кадрів, що є критичним для точної оцінки синхронізації за допомогою зовнішньої моделі, такої як AVAlignNet, яка використовується у функції втрат синхронізації. Таким чином, завантажувач даних готує не тільки вхідні дані для генератора DINet (вихідний кадр, опорні кадри, аудіоознаки), але й надає необхідну геометричну інформацію для подальшої обробки та оцінки згенерованих зображень.

Візуалізація роботи завантажувача даних, що демонструє вихідний кадр, кадр із маскою, оброблений кадр (вирівняна область рота), опорний кадр та відповідну мел-спектрограму, наведена на рисунку 3.19.

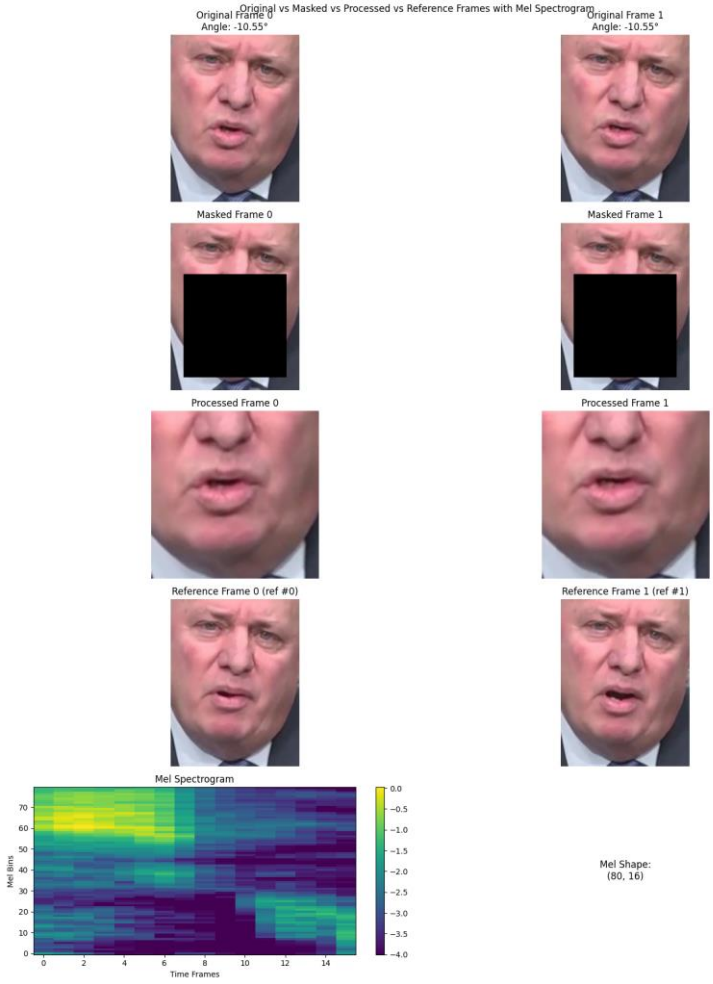

Рисунок 3.19 – Візуалізація даних з процесу навчання DINet

Процес навчання адаптованої моделі DINet базується на оптимізації комбінованої функції втрат, як детально описано в розділі 2.4.2. Ця функція включає перцепційну втрату, адверсарну втрату, втрату синхронізації, та пряму втрату на зображення. Для мінімізації загальної функції втрат використовується оптимізатор AdamW, а для прискорення обчислень застосовується тренування зі змішаною точністю.

Важливою модифікацією в архітектурі DINet, а також в інших залучених компонентах, стала заміна шарів BatchNorm1D та BatchNorm2D на InstanceNorm1D та InstanceNorm2D відповідно. Ця зміна є поширеною практикою в генеративних моделях, особливо в задачах, пов'язаних із передачею стилю та генерацією зображень.

InstanceNorm нормалізує ознаки для кожного екземпляра даних (зображення) окремо по просторових розмірностях, на відміну від BatchNorm, яка нормалізує по всьому пакету даних. Такий підхід дозволяє зберегти індивідуальні стилістичні характеристики кожного зображення, що є важливим для генерації фотореалістичних та візуально узгоджених результатів. BatchNorm може «змивати» важливу для стилю інформацію, усереднюючи статистику по батчу, тоді як InstanceNorm допомагає усунути специфічні для екземпляра контрастні відмінності, спрощуючи для моделі завдання генерації.

3.3.2 Оцінка якості генерації та синхронізації

Для кількісної оцінки результатів роботи розроблених моделей було використано стандартний набір метрик, детально описаний у розділі 1.5: SSIM (Structural Similarity Index) та PSNR (Peak Signal-to-Noise Ratio) для оцінки піксельної та структурної схожості згенерованих кадрів з еталонними; LPIPS (Learned Perceptual Image Patch Similarity) для оцінки перцептивної якості зображень; а також LSE-D (Lip Sync Error-Distance) та LSE-C (Lip Sync Error-Confidence) для вимірювання точності аудіовізуальної синхронізації.

Приклад візуальної якості згенерованих кадрів модифікованою моделлю DINet наведено на рисунку 3.20, де показано вхідний кадр, маскований кадр та результат генерації області рота.

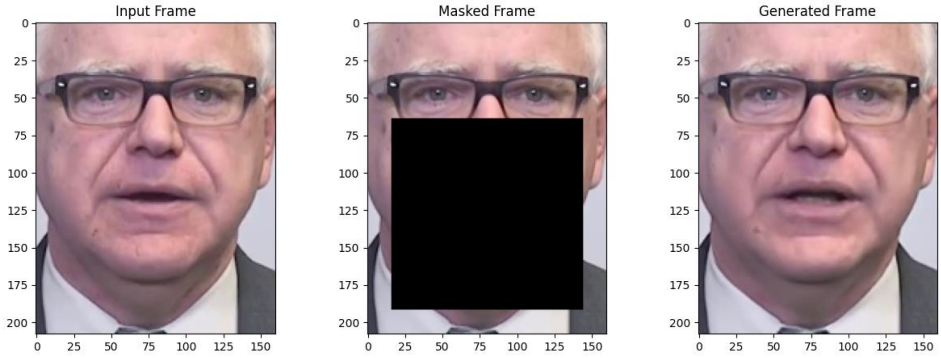

Рисунок 3.20 – Візуалізація згенерованого фрейму

Порівняльні результати для ключових конфігурацій представлені в таблиці 3.1. Основна увага приділяється порівнянню DINet-AVAlignNet (модифікована DINet, що використовує покращену модель AVAlignNet для розрахунку втрати синхронізації) та DINet-Wav2lip (аналогічна DINet, але з використанням оригінальної моделі SyncNet від Wav2Lip). Обидві моделі тренувалися з розміром області рота 128x128 пікселів та з використанням змішаної точності.

Таблиця 3.1 – Порівняння метрик якості для різних моделей синхронізації губ

	SSIM ↑	PSNR ↑	LPIPS ↓	LSE-D ↓	LSE-C ↑
Wav2Lip-96	0.9078	29.2875	0.0576	8.3771	6.8416
DINet-Wav2lip	0.9265	29.4812	0.0545	7.4523	7.6587
DINet-AVAlignNet	0.9382	29.9153	0.0315	7.0112	8.1345
DINet	0.9425	30.0082	0.0289	6.8714	8.2908

При порівнянні DINet-AVAlignNet та DINet-Wav2lip спостерігаються значущі відмінності як у якості синхронізації, так і в якості генерації зображень, незважаючи на ідентичну архітектуру основного генератора DINet.

Модель DINet-AVAlignNet демонструє суттєво кращі показники синхронізації порівняно з DINet-Wav2lip. Це очікувано, оскільки AVAlignNet, як було показано в розділі 3.2, є більш точною та стійкою моделлю для оцінки аудіовізуальної синхронізації, ніж стандартна SyncNet, що використовується в Wav2Lip. Зменшення LSE-D на приблизно 0.45 та збільшення LSE-C на приблизно 0.48 свідчать про те, що генератор, керований більш якісним «вчителем» у вигляді AVAlignNet, навчається продукувати рухи губ, які краще відповідають аудіоряду.

Цікаво, що DINet-AVAlignNet також перевершує DINet-Wav2lip за всіма метриками якості генерації зображень. Покращення візуальної якості, незважаючи на те, що архітектура самого генератора DINet ідентична в обох випадках, а відрізняється лише модуль, що відповідає за втрату синхронізації, заслуговує на окрему увагу.

Покращена модель синхронізації губ AVAlignNet суттєво підвищує якість роботи генератора, надаючи йому більш інформативний та стабільний градієнт від компоненти втрат, що відповідає за синхронізацію. Коли генератор отримує чіткіші сигнали щодо оптимального формування губ для ідеальної синхронізації, він може ефективніше налаштовувати свої параметри. Це не лише безпосередньо покращує синхронізацію, але й опосередковано сприяє генерації більш природних та візуально коректних текстур і форм в області рота, оскільки рухи, «правильні» з точки зору синхронізації, часто виглядають природніше.

Крім того, точніший сигнал від AVAlignNet зменшує конфлікт з іншими функціями втрат. Загальна функція втрат у таких системах, як DINet, є комбінацією кількох складових, що оцінюють різні аспекти якості, включно із синхронізацією та фотореалістичністю зображення. Менш точна модель синхронізації, подібна до тієї, що використовується в DINet-Wav2lip, може генерувати «шумні» або навіть суперечливі градієнти для компоненти, відповідальної за синхронізацію. Це змушує генератор йти на компроміси, які негативно впливають на візуальну якість, оскільки він

намагається задовольнити неточні вимоги синхронізації. Навпаки, більш надійний сигнал від AVAlignNet дозволяє генератору краще узгоджувати вимоги синхронізації з вимогами фотореалістичності від інших компонентів функції втрат, що призводить до загального покращення якості.

Нарешті, якісніший зворотний зв'язок від AVAlignNet сприяє ефективнішому дослідженню простору рішень. Це спрямовує процес навчання генератора в такі області простору параметрів, де одночасно досягається як висока точність синхронізації, так і висока візуальна якість. Завдяки цьому генератор може краще моделювати складні взаємозв'язки між аудіо-ознаками та тонкими нюансами міміки, що відповідають за реалістичне мовлення. Таким чином, точніша синхронізація виступає каталізатором для загального підвищення природності та візуальної привабливості генерованого контенту.

Отже, результати чітко вказують, що використання досконалішої моделі оцінки синхронізації, як AVAlignNet, у процесі навчання генеративної моделі DINet є ключовим фактором для одночасного досягнення вищої точності синхронізації та кращої візуальної якості згенерованих відео. Це підтверджує гіпотезу про те, що якість «навчального сигналу» від компоненти синхронізації має прямий та суттєвий вплив не тільки на саму синхронізацію, але й на здатність генератора відтворювати реалістичні зображення.

ВИСНОВКИ

В цій кваліфікаційній роботі було успішно досягнуто ключової мети – розроблено, реалізовано та всебічно досліджено вдосконалену модель оцінки аудіо-відео синхронізації, названу AVAlignNet. Також було детально оцінено її вплив на якість та стабільність систем генерації синхронізованих рухів губ, зокрема при інтеграції в генеративну модель DINet.

Робота була зосереджена на вирішенні проблеми обмеженої точності та стабільності існуючих моделей оцінки синхронізації, які є критично важливими компонентами для навчання сучасних генеративних систем «говорячих облич». Виходячи з цього, було запропоновано нову архітектуру AVAlignNet. Ця модель була спроєктована з урахуванням принципів ефективності, подібних до EfficientNet, та інкорпорувала передові архітектурні елементи, такі як глибокі каскади резидуальних блоків для ефективного вилучення ознак, механізми самоуваги для врахування глобальних контекстуальних залежностей у відео та аудіо потоках, групову нормалізацію для стабілізації навчання.

Значна увага в дослідженні була приділена підготовці та обробці навчальних даних, що є фундаментальним аспектом для успішного навчання глибоких нейронних мереж. Для експериментів було обрано та ретельно підготовлено аудіо-візуальні датасети HDTF та Hallo3. Було розроблено та впроваджено комплексну методологію попередньої обробки відеоматеріалів, що включала етапи перевірки цілісності файлів, стандартизації форматів, точного розрахунку ключових точок обличчя, кадрування області губ, сегментації довгих відео на коротші фрагменти, а також алгоритмічне виявлення та коригування часових зсувів між аудіо та відео. Спеціально для навчання AVAlignNet та подібних моделей було створено гнучкий завантажувач даних, який забезпечував динамічне афінне перетворення області губ для компенсації поворотів голови та ефективну обробку мел-спектрограм з аудіосигналів.

Експериментальне дослідження AVAlignNet було багатогранним. Було систематично проаналізовано вплив ключових гіперпараметрів та архітектурних виборів на кінцеву ефективність моделі. Зокрема, досліджувалась тривалість часового вікна для аналізу послідовностей кадрів, просторова роздільна здатність області губ, доцільність використання латентних представлень порівняно з прямою обробкою пікселів, вплив максимальної кількості каналів ознак у згорткових шарах. Результати цих експериментів підтвердили, що AVAlignNet стабільно демонструє вищу ефективність в оцінці аудіо-візуальної синхронності порівняно з поширеними аналогами, такими як SyncNet, що використовується в Wav2Lip, особливо при навчанні на датасеті HDTF.

Ключовим етапом роботи стала інтеграція розробленої моделі AVAlignNet як компонента функції втрат синхронізації у більш комплексну генеративну систему – модель DINet, призначену для візуального дубляжу облич. Порівняльний аналіз роботи DINet при використанні AVAlignNet проти використання стандартної SyncNet від Wav2Lip виявив важливу закономірність. Застосування AVAlignNet призвело не лише до очікуваного покращення показників точності синхронізації рухів губ зі звуком, але й до незначного підвищення загальної візуальної якості та реалістичності згенерованих відеозображень [26]. Це свідчить про те, що більш точний та стабільний сигнал зворотного зв'язку від вдосконаленої моделі синхронізації генератору ефективніше оптимізувати свої параметри, краще узгоджуючи вимоги синхронізації з вимогами фотореалістичності та природності артикуляції.

Незважаючи на результати, у ході дослідження також було виявлено певні виклики. Зокрема, тестування на датасеті Hallo3, який характеризується значно більшою варіативністю умов зйомки, освітлення, ракурсів та мовців, продемонструвало суттєве зниження продуктивності як для розробленої AVAlignNet, так і для базової моделі SyncNet. Це підкреслює фундаментальну проблему узагальнювальної здатності моделей

машинного навчання при переході від контрольованих лабораторних умов до складності реального світу. Такі результати вказують на гостру необхідність у створенні більш масштабних, різноманітних та репрезентативних навчальних датасетів, а також у подальшому розвитку методів доменної адаптації та розробці архітектур, більш стійких до варіативності вхідних даних.

Отримані результати та виявлені проблеми формують певний напрям для подальших наукових пошуків, спрямованих на створення ще досконаліших та надійніших систем синхронізації мовлення для віртуальних аватарів та дослідженню впливу моделей, як SyncNet, на процес навчання моделей генерації, різноманіття яких було показано в першому розділі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Audio-Driven facial animation with deep learning: a survey / D. Jiang та ін. *Information*. 2024. Т. 15, № 11. С. 675. URL: https://doi.org/10.3390/info15110675 (дата звернення: 04.04.2025).

2. Extended reality telemedicine collaboration system using patient avatar based on 3D body pose estimation / M. Šarić та ін. *Sensors*. 2023. Т. 24, № 1. С. 27. URL: https://doi.org/10.3390/s24010027 (дата звернення: 04.04.2025).

3. ChatGPT sets record for fastest-growing user base – analyst note / Hu K. *Reuters*. 2023. URL: https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/ (дата звернення: 04.04.2025).

4. AI Avatars Market Size & Share | Statistics Report 2024–2032 / Global Market Insights. *Global Market Insights*. URL: https://www.gminsights.com/industry-analysis/ai-avatars-market (дата звернення: 03.04.2025).

5. Voice Search Statistics (2025) – Worldwide Users & Trends / DemandSage. *DemandSage*. URL: https://www.demandsage.com/voice-search-statistics/ (дата звернення: 03.04.2025).

6. Seeing What You Said: Talking Face Generation Guided by a Lip Reading Expert / Wang J. та ін. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023. С. 14653–14662. URL: https://doi.org/10.1109/CVPR52729.2023.01410 (дата звернення: 03.04.2025).

7. Generative adversarial networks / Goodfellow I. та ін. *Communications of the ACM*. 2020. Т. 63, № 11. С. 139–144. URL: https://doi.org/10.1145/3422622 (дата звернення: 04.04.2025).

8. A lip sync expert is all you need for speech to lip generation in the wild / Prajwal K. R. та ін. *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*. New York, NY, USA, 2020. С. 484–492. URL: https://doi.org/10.1145/3394171.3413532 (дата звернення: 04.04.2025).

9. Out of time: automated lip sync in the wild / Chung J. S., Zisserman A. *Computer Vision – ACCV 2016 Workshops*. Cham, 2017. С. 251–263. URL: https://doi.org/10.1007/978-3-319-54427-4_19 (дата звернення: 04.04.2025).

10. Denoising Diffusion Probabilistic Models / Ho J., Jain A., Abbeel P. *arXiv.org*. 2020. URL: https://doi.org/10.48550/arXiv.2006.11239 (дата звернення: 04.04.2025).

11. Diff2Lip: audio conditioned diffusion models for lip-synchronization / Mukhopadhyay S. та ін. *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA, 2024. С. 5428–5437. URL: https://doi.org/10.1109/wacv57701.2024.00521 (дата звернення: 04.04.2025).

12. DINet: deformation inpainting network for realistic face visually dubbing on high resolution video / Zhang Z. та ін. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2023. Т. 37, № 3. С. 3543–3551. URL: https://doi.org/10.1609/aaai.v37i3.25464 (дата звернення: 04.04.2025).

13. Identity-Preserving talking face generation with landmark and appearance priors / Zhong W. та ін. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vancouver, BC, Canada, 2023. С. 9983–9993. URL: https://doi.org/10.1109/cvpr52729.2023.00938 (дата звернення: 04.04.2025).

14. MuseTalk: Real-Time High-Fidelity Video Dubbing via Spatio-Temporal Sampling / Zhang Y. та ін. *arXiv.org*. 2024. URL: https://doi.org/10.48550/arXiv.2410.10122 (дата звернення: 04.04.2025).

15. EmoTalker: audio driven emotion aware talking head generation / Shen X., Khan F. F., Elhoseiny M. *Lecture Notes in Computer Science*. Singapore, 2024. С. 131–147. URL: https://doi.org/10.1007/978-981-96-0917-8_8 (дата звернення: 04.04.2025).

16. GeneFace++: Generalized and Stable Real-Time Audio-Driven 3D Talking Face Generation / Huang R. та ін. *arXiv.org*. 2023. URL: https://doi.org/10.48550/arXiv.2305.00787 (дата звернення: 04.04.2025).

17. ImageNet classification with deep convolutional neural networks / Krizhevsky A., Sutskever I., Hinton G. E. *Advances in Neural Information Processing Systems 25*. 2012. С. 1097–1105. URL: https://doi.org/10.1145/3065386 (дата звернення: 04.04.2025).

18. Attention is all you need / Vaswani A. та ін. *arXiv.org*. 2024. URL: https://doi.org/10.48550/arXiv.1706.03762 (дата звернення: 04.04.2025).

19. Group normalization / Wu Y., He K. *arXiv.org*. 2024. URL: https://doi.org/10.48550/arXiv.1803.08494 (дата звернення: 04.04.2025).

20. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks / Tan M., Le Q. V. *Proceedings of the 36th International Conference on Machine Learning (ICML)*. Long Beach, California, USA: PMLR, 2019. Т. 97. С. 6105–6114. URL: https://doi.org/10.48550/arXiv.1905.11946 (дата звернення: 04.04.2025).

21. Flow-Guided One-Shot Talking Face Generation With a High-Resolution Audio-Visual Dataset / Zhang Z. та ін. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, 2021. С. 3888–3897. URL: 10.1109/CVPR46437.2021.00366 (дата звернення: 04.04.2025).

22. Hallo3: Highly Dynamic and Realistic Portrait Image Animation with Video Diffusion Transformer / Cui J. та ін. *arXiv.org*. 2024. URL: https://doi.org/10.48550/arXiv.2412.00733 (дата звернення: 04.04.2025).

23. VoxCeleb2: Deep Speaker Recognition / Chung J. S., Nagrani A., Zisserman A. *Proceedings of Interspeech 2018*. Hyderabad, India: ISCA, 2018. С. 1086–1090. URL: https://doi.org/10.48550/arXiv.1806.05622 (дата звернення: 04.04.2025).

24. LRS3-TED: a large-scale dataset for visual speech recognition / Afouras T. та ін. *arXiv.org*. 2018. URL: https://doi.org/10.48550/arXiv.1809.00496 (дата звернення: 04.04.2025).

25. Deep Residual Learning for Image Recognition / He K. та ін. *Proceedings of the IEEE Conference on Computer Vision and Pattern*

Recognition (CVPR). Las Vegas, NV, USA: IEEE, 2016. С. 770–778. URL: https://doi.org/10.1109/CVPR.2016.90 (дата звернення: 04.04.2025).

26. Мирошник Ю., Рябова Н. Методи оцінки якості синхронізації аудіо та відео на основі нейронних мереж. Радіоелектроніка та молодь у XXI столітті: матеріали 29-го Міжнар. молодіж. форуму, 16–19 квіт. 2025 р. Харків, 2025. Т. 6 С. 42–43.