Вінницький національний технічний університет
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

ІСАЄНКОВ ЯРОСЛАВ ОЛЕКСАНДРОВИЧ

УДК 004.054:[004.032.26+004.85]

ДИСЕРТАЦІЯ
ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ НА ОСНОВІ САМОВАЛІДОВАНОЇ ГЕНЕРАТИВНОЇ ЗМАГАЛЬНОЇ МЕРЕЖІ

Спеціальність 126 – «Інформаційні системи та технології»
Галузь знань 12 – «Інформаційні технології»

Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело _____ Я. О. Ісаєнков

Науковий керівник: Мокін Олександр Борисович,
доктор технічних наук, професор

Вінниця – 2024

АННОТАЦІЯ

Ісаєнков Я.О. Інформаційна технологія на основі самовалідованої генеративної змагальної мережі – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 126 «Інформаційні системи та технології». – Вінницький національний технічний університет, Вінниця, 2024.

Метою дослідження є покращення якості генерування даних шляхом створення моделей, заснованих на архітектурі генеративних змагальних мереж, і методів оцінювання результатів їх генерування, а також створення інформаційної технології на основі самовалідованої генеративної змагальної мережі.

У вступі наведено: актуальність теми дослідження, мету та завдання дослідження; об’єкт і предмет дослідження, а також наукову новизну отриманих результатів; зв’язок роботи з науковими програмами, планами, темами та грантами; практичну цінність отриманих результатів; апробацію матеріалів дисертації та публікації.

У першому розділі подано аналіз проблеми оцінювання результатів генерації штучних даних. Розділ включає опис задач, що вирішують генеративні моделі, різновиди генеративних моделей, їхні переваги та недоліки, метрики, що використовуються для оцінки згенерованих даних, а також проблеми використання генеративних моделей у задачах сегментації. Визначено задачі дослідження.

У другому розділі викладено етапи розробки самовалідованих генеративних змагальних мереж. Спочатку описано розробку базової версії ГЗМ Васерштейна та її апробацію на задачі генерації квітів та збільшення набору даних для оптичного розпізнавання символів на ID-картках. Далі наведено удосконалення архітектури ГЗМ за допомогою U-генератора для забезпечення створення сегментаційної маски разом із цільовим об’єктом. Після цього запропоновано архітектуру самовалідованої ГЗМ, яка включає компоненти

мінімізації змін вхідного зображення та механізм вибору найкращого стану генератора під час навчання.

У третьому розділі описано розробку методу оцінювання частково згенерованих даних. Він включає приклад недієздатності метрики FID для набору частково згенерованих стоматологічних панорманих знімків зубів. Після цього наведено опис нової метрики для оцінки таких даних. Розділ завершується прикладом використання розробленого методу для оцінювання частково згенерованих КТ-знімків зубів із залученням 30 осіб.

У четвертому розділі подано проектування, реалізацію, прикладне застосування та порівняння з аналогами інформаційної технології. Проектування програмного модуля описано з використанням схем, IDEF0 та UML-діаграм. Наведено програмну реалізацію із прикладами коду основних модулів та класів. Описано застосування системи на наборах даних стоматологічних пломб та супутникових знімків кораблів. Проведено порівняння функціональних можливостей з аналогами. Описано впровадження технології у навчальний процес ВНТУ.

У висновках подані основні результати дисертаційного дослідження.

Основні наукові та практичні результати полягають в наступному:

1. Проведено огляд відомих підходів до генерації даних за допомогою моделей глибокого навчання, таких як автокодувальники (звичайні та варіаційні), генеративні змагальні мережі (DCGAN, Wasserstein GAN, BigGAN, ProGAN, StyleGAN, StyleGAN2) і дифузійні моделі (Stable Diffusion 1-3, Stable Diffusion XL). Показано їхні переваги та недоліки. Вибір зроблено на користь генеративних змагальних мереж завдяки оптимальному балансу між перевагами та недоліками порівняно з іншими моделями. Недоліки автокодувальників включають втрату інформації через надмірне стиснення шару коду, що призводить до низької якості відновлених даних та обмежених можливостей генерації. Дифузійні моделі характеризуються високою обчислювальною складністю, що робить процес тренування та генерації ресурсомістким і тривалим, оскільки генерація даних може займати багато часу через поступовий характер процесу. Основними

недоліками генеративних змагальних мереж є складність тренування, яка вимагає ретельного налаштування гіперпараметрів, і нестабільність процесу, зокрема ризик перенавчання, коли генератор і дискримінатор перестають покращуватися, що може призвести до колапсу мод генерованих даних. У даній роботі пропонується рішення цих проблем за допомогою методу самовалідації.

2. Виконано аналіз основних проблем оцінювання результатів генерації: на основі людської оцінки (методи HYPEtime та HYPE∞) та автоматичних методів оцінки (методи Inception Score, Fréchet Inception Distance, влучність, точність, покриття, щільність, GAN-train та GAN-test, непараметричний тест для виявлення копіювання даних, Perceptual Path Length, метрика лінійної роздільності). Проаналізовано переваги, недоліки та обмеження кожного з методів, зроблено акцент на недостатній увазі методів щодо оцінки частково згенерованих даних. Показано неефективність метрики FID на наборі частково аугментованих КТ-знімків зубів: для відомо поганих трансформацій вона дає такі ж значення, як і для якісних трансформацій.

3. Розроблено нову технологію для підвищення ефективності генеративних змагальних мереж. Запропоновано метод застосування відомої архітектури U-генератора в складі генеративної змагальної мережі для задачі трансформування цільового класу на основі нульового класу фону з одночасним генеруванням сегментаційної маски для створеного цільового об'єкта. На основі цього вперше запропоновано модель самовалідованої генеративної змагальної мережі для задачі трансформування цільового класу сегментації. Цей підхід включає введення до архітектури навчання генеративної змагальної мережі додаткових самовалідаційних компонентів, зокрема: компонент штрафу за значну зміну структури зображення, яке надходить на вхід генератора; компонент автоматичних метрик оцінювання, серед яких можна виділити модель на основі претренованої сегментаційної моделі, що сегментує згенероване зображення, та метрику, яка порівнює цю маску з маскою, автоматично створеною генератором. Це дозволяє в процесі навчання моделі використовувати інтерпретовані метрики оцінювання, що демонструють, наскільки згенеровані дані є зрозумілими для

сегментаційної моделі та наскільки добре маска, згенерована генератором, відповідає вимогам.

4. Запропоновано підхід оцінювання частково аугментованих даних, заснований на людській оцінці. Він включає взаємодію з користувачами, яким необхідно визначити на повному зображенні зони, що були перетворені. В залежності від точності визначення зон (величини перетину зон вказаних користувачем з реально зміненими зонами) розраховуються метрики користувача (влучність, повнота, F1 міра). Ці метрики є оберненими до метрик моделі, результати якої оцінюються: чим краще людина ідентифікує аугментовані зони, тим менш правдоподібно їх створює модель. Проведено порівняльне оцінювання трьох розроблених моделей: 1) генеративна змагальна мережа на основі U-генератора; 2) та ж модель, але з постобробкою вихідного зображення і сегментаційної маски; 3) самовалідована ГЗМ. Оцінювання проводили 30 осіб для набору панорамних знімків зубів. Результати показали перевагу самовалідованої ГЗМ, що також підтверджується іншими метриками.

5. Розроблено структуру запропонованої інформаційної технології на основі самовалідованої генеративної змагальної мережі з окремим компонентом оцінювання та охарактеризовано 6 розроблених модулів. Побудовано контекстну діаграму IDEF0 для системи. Створено UML-діаграми використання для кожної з підсистем та UML-діаграми класів для опису програмної архітектури модулів. Також наведено діаграми послідовності та активності для опису взаємодії між модулями.

6. Розроблено програмно-інформаційне забезпечення для автоматизації запропонованих методів і компонентів інформаційної технології. На основні складові отримано свідоцтво про реєстрацію авторського права на твір (комп’ютерну програму) «Трансформація цільового класу для задачі сегментації з використанням U-GAN». Деякі модулі, такі як модуль тренування самовалідаційного компонента претренованої сегментаційної моделі, модуль тренування самовалідованої генеративно-змагальної мережі, модуль візуалізації результатів генерації, а також модуль тренування сегментаційної моделі для задачі

сегментації повних зображень, були реалізовані у вигляді програм-ноутбуків на платформі Kaggle, але можуть бути легко перенесені на інші популярні платформи (AWS SageMaker, Google Colab, Jupyter Notebook або JupyterLab, Microsoft Azure Notebooks, PyCharm, Visual Studio Code). Модуль часткової аугментації зображень і модуль оцінки частково згенерованих даних було реалізовано у Visual Studio Code у вигляді застосунків з веб-інтерфейсом для зручного використання іншими користувачами. Проведено випробування програмно-інформаційного забезпечення на реальних даних. Його успішно застосовано для аугментації даних у процесі навчання моделей сегментації на наборах даних панорамних знімків зубів (зокрема пломб) та супутникових знімках кораблів. Порівняно якість сегментаційних моделей для частин зображень із лише цільовим класом та для повних зображень, звідки ці частини були взяті, без використання розробленої технології та з її застосуванням. В усіх експериментах на обох наборах даних зафіксовано покращення метрик. Проведено порівняння функціональних можливостей та якості з такими аналогами, як бібліотека для мови програмування Python albumentations і веб-застосунок Stable Diffusion Web UI.

Основні результати дисертаційного дослідження викладені у 10 наукових працях. Опубліковано 4 статті у фахових журналах України зі спеціальності 126; 1 стаття, що включена до наукометричної бази даних Scopus; 5 тез доповідей на науково-практичних конференціях.

Наукова новизна дисертаційної роботи полягає в наступних положеннях:

1. Уперше запропоновано метод трансформування нульового класу фону в цільовий клас із одночасним генеруванням сегментаційної маски з використанням відомої архітектури U-генератора, що дозволяє використовувати цей підхід для покращення якості сегментаційних моделей, вирішуючи проблему дисбалансу кількості даних у цільових класах під час тренування шляхом формування або аугментації наборів даних.

2. Уперше запропоновано інформаційну модель самовалідованої генеративної змагальної мережі для задачі трансформування нульового класу фону в цільовий клас із одночасним генеруванням сегментаційної маски шляхом

введення самовалідованих компонентів. Один із цих компонентів – це штраф за значну зміну структури зображення, яке подається на вхід генератора, а інший – автоматична метрика оцінювання, яка заснована на претренованій сегментаційній моделі, що сегментує згенероване зображення, і метриці, яка порівнює цю маску з маскою, автоматично створеною генератором. Це дозволяє покращити якість генерування часткових даних, що інтегруються в основне зображення, а також додає нову інтерпретовану метрику оцінювання під час навчання моделі.

3. Уперше запропоновано метод оцінювання згенерованих даних у задачі часткової аугментації для набору даних панорамних рентгенівських знімків зубів, заснований на людській оцінці, що дозволяє вибирати з декількох моделей ту, яка створює найякісніші зображення. Метод включає взаємодію з користувачами, яким необхідно визначити зони на повному зображенні, які, на їх думку, були перетворені. Залежно від точності визначення зон розраховуються метрики, що відображають якість оцінювання користувача (повнота, влучність, міра F1). Ці метрики є оберненими до метрик моделі, яка оцінюється, оскільки чим краще користувач ідентифікує аугментовані зони, тим менш правдоподібно їх створює модель.

Отримані результати є корисними для:

- рекомендовано ТОВ «ОРНЕТІКА» для впровадження в практичну діяльність при розробці інтелектуальних систем для вирішення задач комп’ютерного зору, а саме: генерування синтетичних зразків даних панорамних рентгенівських знімків зубів з використанням самовалідованих генеративних змагальних мереж; використання методу оцінки частково згенерованих даних для ситуацій коли необхідне використання частково згенерованих даних; проектування системи часткового аугментування даних; тренування сегментаційних моделей для вирішення задачі ідентифікації об’єктів на зображеннях (акт впровадження від 6 вересня 2024р.);

- навчального процесу Вінницького національного технічного університету під час викладання дисципліни «Інформаційні технології моніторингу та аналізу стану складних систем» для студентів, які навчаються за

освітньою програмою «Інформаційні системи та технології» рівня «аспірант» спеціальності 126 Інформаційні системи та технології, а також – дисципліни «Інтернет речей та інтелектуальний аналіз даних» для аспірантів, які навчаються за освітньою програмою «Інформаційні системи та технології» рівня «доктора філософії» спеціальності 126 для навчання знанням і навичкам застосування методів і моделей машинного навчання та глибокого навчання для генерування зображень, а саме підготовки набору даних для навчання генеративних моделей, тренування та використання моделей генерування, дослідження методів оцінки результатів генерування (акт впровадження від 19 вересня 2024р.).

Інформаційна технологія на основні самовалідованої генеративної змагальної мережі може бути застосована для будь яких доменів даних, для покращення якості тренування систем машинного і глибоко навчання в задачі сегментації за рахунок додавання якісно згенерованих даних разом з їх сегментаційною розміткою.

Ключові слова: генеративна змагальна мережа, нейрона мережа, автокодувальник, інтелектуальна інформаційна технологія, генерування, сегментація, оцінювання, метрика, глибоке навчання, машинне навчання, навчання, комп’ютерний зір, штучний інтелект, обробка зображень, розпізнавання зображень.

ABSTRACT

Isaienkov Y.O. Information Technology Based on Self-Validated Generative Adversarial Network. – Qualifying scientific work with manuscript rights.

Dissertation for obtaining the scientific degree of Doctor of Philosophy in specialty 126 "Information systems and technologies". – Vinnytsia National Technical University, Vinnytsia, 2024.

The aim of the dissertation is to improve the quality of data generation by developing models based on the architecture of Generative Adversarial Networks and methods for evaluating the results of their generation, as well as creating an information technology based on Self-Validated Generative Adversarial Network.

The introduction states: the relevance of the research subject, the purpose and task of the research; object, scope of the research and scientific novelty of the obtained results; connection of work with scientific programs, plans, topics, grants; practical value of the obtained results; approval of thesis and publication materials.

The first chapter presents an analysis of the problem of evaluating the results of artificial data generation. It includes a description of the tasks solved by generative models, the types of generative models, their advantages and disadvantages, the metrics used to evaluate generated data, and the challenges of using generative models in segmentation tasks. Research objectives are defined.

The second chapter describes the stages of developing Self-Validated Generative Adversarial Network. It begins with the development of the basic version of the Wasserstein GAN and its testing on tasks of generating flowers and expanding the dataset for optical character recognition on ID cards. Next, the chapter describes the improvement of the GAN architecture using a U-generator to ensure the creation of a segmentation mask along with the target object. The architecture of the Self-Validated GAN is then proposed, which includes components for minimizing changes to the input image and a mechanism for selecting the best generator state during training.

The third chapter focuses on the development of a method for evaluating partially generated data. It includes an example of the failure of the Fréchet Inception

Distance metric for a set of partially generated panoramic dental X-rays. A new metric for evaluating such data is then proposed, followed by an example of applying the developed method to evaluate partially generated CT scans of teeth, involving 30 individuals.

The fourth chapter presents the design, implementation, practical application, and comparison with analogs of the information technology. The design of the software module is described using schemas, IDEF0, and UML diagrams. The software implementation is provided with examples of code for the main modules and classes. The application of the system is demonstrated on datasets of dental fillings and satellite images of ships. A comparison of the functional capabilities with analogs is conducted. Described the implementation of the technology in the educational process at VNTU.

The conclusions summarize the main results of the dissertation research.

The main scientific and practical results are as follows:

1. A review of known approaches to data generation using deep learning models, such as autoencoders (standard and variational), Generative Adversarial Networks (DCGAN, Wasserstein GAN, BigGAN, ProGAN, StyleGAN, StyleGAN2), and diffusion models (Stable Diffusion 1-3, Stable Diffusion XL), was conducted. Their advantages and disadvantages were presented. GANs were chosen due to the optimal balance between advantages and disadvantages compared to other models. The drawbacks of autoencoders include the loss of information due to excessive compression in the code layer, which leads to poor quality of recovered data and limited generation capabilities. Diffusion models are characterized by high computational complexity, making the training and generation process resource-intensive and lengthy, as data generation can take a long time due to the gradual nature of the process. The main disadvantages of GANs include the difficulty in training, requiring careful hyperparameter tuning, and the instability of the process, particularly the risk of overfitting, where the generator and discriminator stop improving, which may lead to a mode collapse in the generated data. This work proposes a solution to these issues through the self-validation method.

2. An analysis of the main problems in evaluating generation results was conducted: based on human evaluation ($HYPE_{time}$ and $HYPE_{\infty}$ methods) and automated evaluation methods (Inception Score, Fréchet Inception Distance, precision, accuracy, coverage, density, GAN-train and GAN-test, non-parametric test for detecting data copying, Perceptual Path Length, linear separability metric). The advantages, disadvantages, and limitations of each method were analyzed, with an emphasis on the insufficient attention of methods for evaluating partially generated data. The inefficiency of the FID metric on a dataset of partially augmented dental CT images was demonstrated: it gives the same values for known poor transformations as for high-quality transformations.

3. A new technology to improve the efficiency of GANs was developed. A method of using the well-known U-generator architecture within a GAN for the task of transforming the target class based on the zero-background class while simultaneously generating a segmentation mask for the created target object was proposed. Based on this, for the first time, a Self-Validated GAN model for the task of transforming the target segmentation class was proposed. This approach includes the introduction of additional self-validation components into the GAN training architecture, including: a component that penalizes significant changes in the structure of the input image to the generator; an automatic evaluation metrics component, which includes a model based on a pre-trained segmentation model that segments the generated image, and a metric comparing this mask with the mask automatically created by the generator. This allows using interpretable evaluation metrics during model training, demonstrating how understandable the generated data is to the segmentation model and how well the mask generated by the generator meets the requirements.

4. An approach to evaluating partially augmented data based on human assessment was proposed. It involves interacting with users, who need to identify the areas on the full image that were transformed. Depending on the accuracy of zone identification (the intersection of zones indicated by the user with the actual transformed zones), user metrics (precision, recall, F1 score) are calculated. These metrics are inversely related to the model's metrics being evaluated: the better the user identifies

augmented zones, the less plausible the model generates them. A comparative evaluation of three developed models was conducted: 1) GAN based on U-generator; 2) the same model, but with post-processing of the output image and segmentation mask; 3) Self-Validated GAN. The evaluation was performed by 30 individuals on a set of panoramic dental images. The results showed the advantage of the Self-Validated GAN, which was also confirmed by other metrics.

5. The structure of the proposed information technology based on a Self-Validated GAN with a separate evaluation component was developed. Six developed modules were described. An IDEF0 context diagram for the system was built. UML use case diagrams for each subsystem and UML class diagrams to describe the software architecture of the modules were created. Sequence and activity diagrams were also provided to describe the interaction between the modules.

6. Software and information tools for automating the proposed methods and components of the information technology were developed. A copyright application was submitted for the main components of the software (computer program). Some modules, such as the training module of the self-validation component of the pre-trained segmentation model, the training module of the Self-Validated GAN, the visualization module of generation results, and the segmentation model training module for full image segmentation tasks, were implemented as notebook programs on the Kaggle platform but can be easily transferred to other popular platforms (AWS SageMaker, Google Colab, Jupyter Notebook or JupyterLab, Microsoft Azure Notebooks, PyCharm, Visual Studio Code). The partial image augmentation module and the partially generated data evaluation module were implemented using Visual Studio Code as web applications for convenient use by other users. The software tools were tested on real data. They were successfully applied for data augmentation in the process of training segmentation models on datasets of panoramic dental images (specifically, dental fillings) and satellite images of ships. The quality of the segmentation models was compared for image parts with only the target class and for full images from which these parts were taken, without using the developed technology and with its application. In all experiments on both datasets, metric improvements were noticed. A comparison of the functional capabilities

and quality with such analogs as the albumentations Python library and the Stable Diffusion Web UI web application was conducted.

The main results of the dissertation research are presented in 10 scientific works. Four articles were published in scientific professional periodicals; one article is published in the Scopus scientometric database; five conference abstracts were presented at scientific conferences.

The scientific novelty of the dissertation lies in the following:

1. For the first time, a method for transforming the background's null class into a target class with simultaneous generation of a segmentation mask using a known U-generator architecture has been proposed. This allows the use of this approach to improve the quality of segmentation models by addressing the problem of data imbalance in the target classes during training through the formation or augmentation of datasets.

2. For the first time, an information model of a Self-Validated Generative Adversarial Network for the task of transforming the background's null class into a target class with simultaneous segmentation mask generation has been proposed, introducing self-validation components. One of these components is a penalty for significant structural changes in the input image to the generator, while the other is an automatic evaluation metric based on a pretrained segmentation model that segments the generated image and a metric that compares this mask with the one automatically created by the generator. This improves the quality of partial data generation integrated into the main image and adds a new interpretable evaluation metric during model training.

3. For the first time, a method for evaluating generated data in the task of partial augmentation for a dataset of panoramic dental X-rays based on human assessment has been proposed, allowing the selection of the model that creates the highest quality images. The method involves user interaction, where users need to identify areas in the complete image that they believe have been transformed. Depending on the accuracy of identifying these areas, metrics such as recall, precision, and F1-score are calculated. These metrics are inversely related to the model’s metrics, as the better the user identifies augmented areas, the less plausible the model's creations are.

The obtained results are useful for:

- Recommended for implementation by LLC "ORNETICA" in practical activities for the development of intelligent systems for solving computer vision tasks, specifically: generating synthetic samples of panoramic dental X-ray images using Self-Validated Generative Adversarial Networks; applying the method of evaluating partially generated data in situations where the use of partially generated data is required; designing a system for partial data augmentation; and training segmentation models for solving object identification tasks in images (act of implementation dated September 6, 2024);

- The educational process at Vinnytsia National Technical University, during the teaching of the course "Information Technologies for Monitoring and Analysis of Complex Systems" for students enrolled in the educational program "Information Systems and Technologies" at the postgraduate level, specializing in 126 "Information Systems and Technologies," as well as the course "Internet of Things and Intelligent Data Analysis" for PhD candidates in the same specialty. The results will help train students in the knowledge and skills of applying machine learning and deep learning methods and models for image generation, specifically the preparation of datasets for training generative models, the training and use of generative models, and the study of methods for evaluating generation results (act of implementation dated September 19, 2024).

The information technology based on Self-Validated Generative Adversarial Network can be applied to any data domain to improve the quality of machine learning and deep learning systems for segmentation tasks by adding high-quality generated data with their segmentation labels.

Keywords: generative adversarial network, neural network, autoencoder, intelligent information technology, generation, segmentation, evaluation, metric, deep learning, machine learning, learning, computer vision, artificial intelligence, image processing, image recognition.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковані основні наукові результати дисертації:

[1] Я. О. Ісаєнков і О. Б. Мокін, "Аналіз генеративних моделей глибокого навчання та особливостей їх реалізації на прикладі WGAN," *Вісник Вінницького політехнічного інституту*, том 160, вип. 1, с. 82–94, Берез. 2022. doi: 10.31649/1997-9266-2022-160-1-82-94. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

(Особистий внесок: зроблено аналіз існуючих підходів до генерування зображень; підготовлено набір даних та на ньому реалізовано умовна генеративна змагальна мережа. О. Б. Мокін: узгодження задач дослідження та аналізу результатів, компонування і загальне редагування статті)

[2] O. Bychkov, L. Zubyk, D. Gololobov, Y. Isaienkov, G. Grynkevych, and A. Ivanytska, "The System for Recognizing Useful Information of the Client's ID-Card Based on Machine Learning Technologies," *CEUR Workshop Proceedings,* vol. 2845, pp. 159–169, Dec. 2023. (Входить до Scopus)

(Особистий внесок: Проведено практичні експерименти вибраних моделей для екстракції корисної інформації з ID карток. O. Bychkov: узгодження задач дослідження та аналізу результатів. L. Zubyk: компонування та загальне редагування статті. D. Gololobov: вибір моделей для поставленої задачі. G. Grynkevych: реалізація програми на основі результатів експериментів. A. Ivanytska: створення методу постобробки результатів моделей)

[3] Я. О. Ісаєнков і О. Б. Мокін, "Трансформація цільового класу для задачі сегментації з використанням U-GAN," *Вісник Вінницького політехнічного інституту*, том 172, вип. 1, с. 81–87, Лют. 2024. doi:10.31649/1997-9266-2024-172-1-81-87. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

(Особистий внесок: запропоновано метод трансформування нульового класу фону в цільовий клас із одночасним генеруванням сегментаційної маски з

використанням відомої архітектури U-генератора; використано цей підхід для створення додаткових пар зображень-масок під час тренування сегментаційних моделей. О. Б. Мокін: узгодження задач дослідження та аналізу результатів, компонування і загальне редагування статті)

[4] Я. О. Ісаєнков і О. Б. Мокін, "Самовалідований U-GAN для трансформації цільового класу в задачах сегментації," *Вісник Вінницького політехнічного інституту*, том 174, вип. 3, с. 102-111, Черв. 2024. doi: 10.31649/1997-9266-2024-174-3-102-111. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

(Особистий внесок: запропоновано інформаційну модель самовалідованої генеративної змагальної мережі для задачі трансформування нульового класу фону в цільовий клас із одночасним генеруванням сегментаційної маски шляхом введення самовалідованих компонентів; проведено ряд експериментів на різних наборах даних для оцінки якості створеного і реалізованого методу. О. Б. Мокін: узгодження задач дослідження та аналізу результатів, компонування і загальне редагування статті)

[5] Я. О. Ісаєнков і О. Б. Мокін, "Метод оцінки частково згенерованих даних," *Наука, технології, інновації*, том 31, вип. 3, с. 114-126, Вер. 2024. doi: 10.35668/2520-6524-2024-3-13. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

(Особистий внесок: зроблено детальний огляд автоматичних підходів та підходів із залученням людини для оцінювання генеративних моделей; запропоновано метод оцінювання згенерованих даних у задачі часткової аугментації для набору даних панорамних рентгенівських знімків зубів, заснований на людській оцінці, що дозволяє вибирати з декількох моделей ту, яка створює найякісніші зображення; проведено практичне застосування розробленого методу на наборі даних панорамних стоматологічних знімків. О. Б. Мокін: узгодження задач дослідження та аналізу результатів, компонування і загальне редагування статті)

Наукові праці, які засвідчують апробацію матеріалів дисертації:

[6] Я. Ісаєнков і О. Мокін, "Проблема оцінювання генеративних змагальних мереж," *у Матеріалах L Науково-технічної конференції підрозділів Вінницького національного технічного університету*, 2021, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2021/paper/view/12468.

(Особистий внесок: Представлено огляд основних сучасних методів оцінювання результатів роботи генеративних змагальних мереж, проаналізовано особливості, переваги та недоліки цих методів)

[7] Я. Ісаєнков і О. Мокін, "Артефакт межі у генеративних змагальних мережах," у *Матеріалах LI Науково-технічної конференції підрозділів Вінницького національного технічного університету*, 2022, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/16039.

(Особистий внесок: Описано новий артефакт, який є властивим згенерованим зображенням)

[8] Я. Ісаєнков, "Використання методів глибокого навчання для оцінки якості медичних записок пацієнтів," *у Матеріалах конференції Проблеми вищої математичної освіти: виклики*, 2022, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/pmovc/pmovc22/paper/view/15558.

(Особистий внесок: Описана задача класифікація медичних записок пацієнтів, описано можливості для розширення набору даних, побудована модель класифікації)

[9] Я. Ісаєнков і О. Мокін, "Вплив гіперпараметрів дифузійних моделей на якість генерування," у *Матеріалах LII Науково-технічної конференції підрозділів Вінницького національного технічного університету*, 2023, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/17174.

(Особистий внесок: Розглянуті дифузійні моделі та параметри, що відповідають за якість та реалістичність генерування)

[10] Я. Ісаєнков, О. Мокін, О. Хорошева, і А. Герасимович, "Розробка застосунку для часткової аугментації зображень на основі генеративних моделей," у *Матеріалах LIII Науково-технічної конференції підрозділів Вінницького національного технічного університету,* 2024, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20614.

(Особистий внесок: Описана роль аугментації у процесі тренування моделей машинного та глибокого навчання, зроблена інтеграція самовалідованих ГЗМ у веб-застосунок)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ 22

ВСТУП 23

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ ГЕНЕРУВАННЯ ШТУЧНИХ ДАНИХ 31

1.1 Задачі генерування даних 31

1.2 Огляд основних моделей та методів генерування штучних даних 34

1.2.1 Автокодувальники 34

1.2.2 Генеративні змагальні мережі 36

1.2.3 Дифузійні моделі 45

1.2.4 Висновки щодо основних типів моделей генерування даних .. 51

1.3 Огляд методів оцінювання результатів генерування 53

1.3.1 Автоматичні методи оцінювання 53

1.3.2 Методи на основі людської оцінки 60

1.4 Аналіз структурних проблем на згенерованих зображеннях 62

1.5 Проблема створення даних для задач сегментації 64

1.5.1 Проблема створення масок 65

1.5.2 Проблема дисбалансу кількості даних 65

1.5.3 Аугментація фрагментів зображень 66

1.6 Висновки до розділу та постановка задач дослідження 67

РОЗДІЛ 2 РОЗРОБКА САМОВАЛІДОВАНИХ ГЕНЕРАТИВНИХ ЗМАГАЛЬНИХ МЕРЕЖ ДЛЯ КОНТРОЛЮ ГЕНЕРУВАННЯ ШТУЧНИХ ДАНИХ 70

2.1 Розробка базової версії ГЗМ Васерштейна 70

2.2 Використання розробленої ГЗМ Васерштейна для збільшення набору даних для оптичного розпізнавання символів 84

2.3 Удосконалення архітектури ГЗМ, для автоматичного створення сегментаційної маски, завдяки використанню U-генератору 88

2.4 Удосконалення архітектури ГЗМ для мінімізації змін вхідного зображення шляхом інтеграції компонента самовалідації 100

2.5 Удосконалення архітектури ГЗМ для оптимізації вибору найкращого стану генератора через інтеграцію компонента самовалідації 104

2.6 Висновки до розділу 108

РОЗДІЛ 3 РОЗРОБКА МЕТОДУ ОЦІНЮВАННЯ ЧАСТКОВО ЗГЕНЕРОВАНИХ ДАНИХ 110

3.1 Потреба у розробці метрик для оцінювання частково аугментованих даних 110

3.2 Проектування метрики для оцінювання частково аугментованих даних, створених генеративними моделями 117

3.3 Перевірка метрики для оцінювання частково аугментованих даних .. 120

3.4 Висновки до розділу 126

РОЗДІЛ 4 ПРОЕКТУВАННЯ, РЕАЛІЗАЦІЯ ТА ПРИКЛАДНЕ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ САМОВАЛІДОВАНИХ ГЕНЕРАТИВНИХ ЗМАГАЛЬНИХ МЕРЕЖ 128

4.1 Проектування програмного модулю самовалідованих генеративних змагальних мереж 128

4.2 Реалізація комплексу програм для застосування інформаційної технології самовалідованих генеративних змагальних мереж 141

4.3 Застосування створеної інформаційної технології для аугментації даних в процесі навчання моделей сегментації 157

4.3.1 Трансформація зубів на стоматологічних панорамних рентгенівських знімках 157

4.3.2 Трансформація суден на супутникових знімках 162

4.4 Порівняння функціональних можливостей створеної інформаційної технології з аналогами 165

4.5 Впровадження розробленої технології у навчальний процес 170

4.6 Висновки до розділу 171

ВИСНОВКИ 174

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 178

ДОДАТКИ 192

ДОДАТОК А ТРАНСФОРМОВАНІ НАБОРИ ДАНИХ ПАНОРАМНИХ ЗНІМКІВ ЗУБІВ ТА ОЦІНЮВАННЯ ЇХ КОРИСТУВАЧАМИ 193

ДОДАТОК Б ТАБЛИЦЯ ОЦІНОК ГЕНЕРАТИВНИХ МОДЕЛЕЙ ДЛЯ КОЖНОГО ЗОБРАЖЕННЯ 204

ДОДАТОК В СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ 205

ДОДАТОК Г АКТ ПРО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ 207

ДОДАТОК Д АКТ ПРО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ У НАВЧАЛЬНИЙ ПРОЦЕС ВНТУ 208

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ВНТУ – Вінницький національний технічний університет

ГЗМ – Генеративна Змагальна Мережа

GAN – Generative Adversarial Network

WGAN – Wasserstein Generative Adversarial Networks

WGAN-GP – Wasserstein Generative Adversarial Networks with gradient penalty

DCGAN – Deep Convolutional Generative Adversarial Networks

DDPM – Denoising Diffusion Probabilistic Models

IS – Inception Score

FID – Fréchet Inception Distance

HYPE – Human eYe Perceptual Evaluation

CV – Computer Vision

КТ – Комп’ютерна Томографія

ООП – Об'єктно-Орієнтоване Програмування

UML – Unified Modeling Language

IDEF0 – Integration Definition for Function Modeling

UI – User Interface

ВСТУП

Обґрунтування вибору теми дослідження.

З моменту прориву у сфері глибокого навчання в 2012 році, напрямок генерування штучних даних за допомогою моделей машинного навчання розвивається з неймовірною швидкістю. Цей розвиток стає все більш актуальним і затребуваним у різних галузях науки і техніки. Згенеровані дані знаходять широке застосування в різних областях, включаючи розширення наборів даних для машинного навчання (так звана аугментація), створення навчальних матеріалів, дизайну та навіть у мистецтві. Це включає як моделі, що трансформують стиль одних зображень або картин на інші, так і ті, що генерують абсолютно нові екземпляри без будь-якої прив’язки до вхідних зображень, або засновуючись тільки на описі того, що потрібно згенерувати.

Серед всіх методів генерування даних, генеративні змагальні мережі, які були винайдені у 2014 році, демонструють дуже гарні результати. Ці мережі, а також більш складні і ресурсомісткі дифузійні моделі, на сьогоднішній день є провідними інструментами у задачах генерування даних. Генеративні змагальні мережі складаються з двох основних компонентів: генератора, який створює дані, та дискримінатора (або «критика»), який намагається відрізнити реальні дані від згенерованих. Хоча генеративні змагальні мережі показують гарні результати у багатьох задачах та їх дослідженням займаються такі провідні вчені як Ian Goodfellow, Shi Zhou, Antonia Creswell, Jean Pouget-Abadie, Xudong Mao, вони все ще мають кілька серйозних проблем з контролем та оцінюванням генерування, які потребують подальшого вивчення та вирішення.

По-перше, на сьогоднішній день не існує універсального методу оцінювання результатів генерування даних. У задачах класифікації, чи інших задачах навчання з учителем, результати роботи моделі можна легко порівняти з відомими правильними мітками вхідних даних, що дозволяє об'єктивно оцінити її якість. Однак, у випадку генеративних моделей, немає загальноприйнятого

стандарту для оцінювання якості згенерованих даних, а існуючі метрики мають суттєві обмеження у використанні для специфічних доменів.

По-друге, генеративні змагальні мережі не часто використовуються у задачах сегментації, де важливо отримувати чи трансформувати маску цільового класу.

По-третє, процес тренування генеративних моделей включає суб'єктивізм у виборі найкращих моделей, що може впливати на фінальні результати.

По-четверте, в задачах часткового генерування, коли трансформується тільки частина зображення, генеративні моделі часто створюють зразки, які важко вставити в основне зображення, через проблеми несхожості контексту, і вставлений шматочок виглядає неприродньо.

Виходячи з вищезазначеного, виникає потреба у розробленні нової інтелектуальної інформаційної технології самовалідованих генеративних змагальних мереж. Така технологія повинна забезпечувати генерування якісних штучних даних та сегментаційних міток, що будуть відповідати критеріям якості заданої предметної області. Вона також має включати розроблення нової метрики для оцінювання частково згенерованих даних, яка буде боротись з недоліками існуючих методів, та розробку програмного забезпечення, яке зробить застосування цієї технології зручним і доступним для широкого кола користувачів.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота здійснювалася здобувачем відповідно до наукового напрямку кафедри системного аналізу та інформаційних технологій Вінницького національного технічного університету, зокрема, під час виконання науково-дослідної роботи 28-К7 «Інформаційні технології та системи моніторингу, моделювання та системного інтелектуального аналізу й оптимізації даних у складних об’єктах» кафедри САІТ. Здобувач був виконавцем госпдоговірної НДР № 2860 «Розробка інформаційних технологій для оптимізації роботи зернового елеватора з використанням нейромережевих моделей та методів навчання з підкріпленням» на замовлення ТОВ «ІННОВІННПРОМ» за фінансування ЄС, яка

виконувалась у ВНТУ в науково-дослідній лабораторій штучного інтелекту та когнітивістики кафедри САІТ ВНТУ у 2023 році.

Мета і завдання дослідження.

Метою дослідження є покращення якості генерування даних шляхом створення моделей, заснованих на архітектурі генеративних змагальних мереж, і методів оцінювання результатів їх генерування, а також створення інформаційної технології на основі самовалідованої генеративної змагальної мережі.

У результаті проведеного аналізу були сформульовані наступні задачі дослідження для досягнення поставленої мети:

1. Здійснити детальний аналіз сучасних підходів до генерування даних за допомогою моделей глибокого навчання. Особливу увагу приділити автокодувальникам, генеративним змагальним мережам та дифузійним моделям, розглянувши їх ключові принципи, переваги та недоліки.

2. Проаналізувати ключові проблеми, пов'язані з оцінюванням результатів генерування даних. Зокрема, визначити основні виклики, які виникають під час оцінки якості згенерованих даних, а також дослідити обмеження існуючих метрик оцінювання та їх вплив на точність результатів.

3. Створити нові підходи, методи та алгоритми, які сприятимуть покращенню якості згенерованих даних, оптимізації процесу генерування та зменшенню часу, необхідного для вибору найкращих моделей, забезпечуючи при цьому якість роботи мереж.

4. Удосконалити методи оцінювання результатів генерування: розробити нові технології та інструменти для оцінювання якості частково згенерованих даних, які будуть боротись з недоліками існуючих методів і дозволять об'єктивно оцінювати результати.

5. Розробити структуру нової інформаційної технології самовалідованих генеративних змагальних мереж, визначивши її архітектуру та ключові компоненти з використанням UML-діаграм. Описати функціональні можливості кожного компонента та їх взаємодію, що мають забезпечувати ефективність і узгодженість роботи всієї системи.

6. Розробити комплекс програмних модулів мовою програмування Python для впровадження та використання інформаційної технології самовалідованих генеративних змагальних мереж. Цей набір модулів має забезпечити ефективне та універсальне вирішення практичних задач, дозволяючи легко адаптувати технологію до різних сценаріїв застосування.

7. Провести апробацію та практичне застосування розробленої інформаційної технології самовалідованих генеративних змагальних мереж на низці реальних задач. Оцінити ефективність запропонованих рішень та їхню доцільність для використання у конкретних прикладних сценаріях.

Об'єкт дослідження – процес створення, тренування та оцінювання результатів генеративних змагальних мереж.

Предмет дослідження – математичні моделі та методи глибокого навчання для генерування даних, оцінювання якості згенерованих даних.

Методи дослідження.

У дослідженні використовувалась загальнонаукова методологія проведення досліджень і принципи системного підходу, а саме: аналіз літературних джерел та предметної області, огляд та порівняння існуючих моделей, експериментальні дослідження, комп'ютерні експерименти, проектування програмного забезпечення, методи машинного та глибокого навчання, методи комп'ютерного зору та обробки зображень.

Наукова новизна отриманих результатів.

1. Уперше запропоновано метод трансформування нульового класу фону в цільовий клас із одночасним генеруванням сегментаційної маски з використанням відомої архітектури U-генератора, що дозволяє використовувати цей підхід для покращення якості сегментаційних моделей, вирішуючи проблему дисбалансу кількості даних у цільових класах під час тренування шляхом формування або аугментації наборів даних.

2. Уперше запропоновано інформаційну модель самовалідованої генеративної змагальної мережі для задачі трансформування нульового класу фону в цільовий клас із одночасним генеруванням сегментаційної маски шляхом

введення самовалідованих компонентів. Один із цих компонентів – це штраф за значну зміну структури зображення, яке подається на вхід генератора, а інший – автоматична метрика оцінювання, яка заснована на претренованій сегментаційній моделі, що сегментує згенероване зображення, і метриці, яка порівнює цю маску з маскою, автоматично створеною генератором. Це дозволяє покращити якість генерування часткових даних, що інтегруються в основне зображення, а також додає нову інтерпретовану метрику оцінювання під час навчання моделі.

3. Уперше запропоновано метод оцінювання згенерованих даних у задачі часткової аугментації для набору даних панорамних рентгенівських знімків зубів, заснований на людській оцінці, що дозволяє вибирати з декількох моделей ту, яка створює найякісніші зображення. Метод включає взаємодію з користувачами, яким необхідно визначити зони на повному зображенні, які, на їх думку, були перетворені. Залежно від точності визначення зон розраховуються метрики, що відображають якість оцінювання користувача (повнота, влучність, міра F1). Ці метрики є оберненими до метрик моделі, яка оцінюється, оскільки чим краще користувач ідентифікує аугментовані зони, тим менш правдоподібно їх створює модель.

Практичне значення отриманих результатів.

На основні отриманих наукових результатів розроблена інформаційна технологія, що включає комплекс програмних модулів на Python, які реалізують функціонал всіх запропонованих методів, моделей та підходів генерування та оцінювання частково згенерованих даних. Використовуючи інформаційну технологію, проведено ряд експериментів на наборах даних панорамних рентгенівських знімків зубів та супутникових знімків кораблів, де показано покращення якості результуючих моделей сегментації в порівнянні з моделями без додаткових згенерованих даних.

Програми-ноутбуки є публічно доступними у Google-платформі експертів аналізу даних та штучного інтелекту Kaggle. Інша частина модулів, які переважно реалізують веб-сервіси для взаємодії з користувачами, публічно доступні у вигляді GitHub-репозиторіїв.

Результати роботи рекомендовані до впровадження в практичну діяльність при розробці інтелектуальних систем для вирішення задач комп'ютерного зору ТОВ «ОРНЕТІКА» (див. Додаток Г).

Також, результати роботи впроваджено у навчальний процес і наукову діяльність кафедри системного аналізу та інформаційних технологій ВНТУ (див. Додаток Д).

Особистий внесок здобувача.

Усі теоретичні результати, що виносяться на захист, отримані здобувачем особисто. У роботі [1] зроблено аналіз існуючих підходів до генерування зображень; підготовлено набір даних та на ньому реалізовано умовна генеративна змагальна мережа. У [2] проведено практичні експерименти вибраних моделей для екстракції корисної інформації з ID карток. У [3] запропоновано метод трансформування нульового класу фону в цільовий клас із одночасним генеруванням сегментаційної маски з використанням відомої архітектури U-генератора; використано цей підхід для створення додаткових пар зображень-масок під час тренування сегментаційних моделей. У [4] запропоновано інформаційну модель самовалідованої генеративної змагальної мережі для задачі трансформування нульового класу фону в цільовий клас із одночасним генеруванням сегментаційної маски шляхом введення самовалідованих компонентів; проведено ряд експериментів на різних наборах даних для оцінки якості створеного і реалізованого методу. У [5], [6] зроблено обширний огляд автоматичних підходів та підходів з залученням людини для оцінки генеративних моделей; проаналізовано особливості, переваги та недоліки цих методів; запропоновано метод оцінювання згенерованих даних у задачі часткової аугментації для набору даних панорамних рентгенівських знімків зубів, заснований на людській оцінці, що дозволяє вибирати з декількох моделей ту, яка створює найякісніші зображення; проведено практичне застосування розробленого методу на наборі даних панорамних стоматологічних знімків. У [7] описано новий артефакт, який є властивим згенерованим зображенням. У [8] описана задача класифікація медичних записок пацієнтів, описано можливості для

розширення набору даних, побудована модель класифікації. У [9] розглянуті дифузійні моделі та параметри, що відповідають за якість та реалістичність генерування. У [10] описана роль аугментації у процесі тренування моделей машинного та глибокого навчання, зроблена інтеграція самовалідованих ГЗМ у веб-застосунок.

Апробація матеріалів дисертації.

Основні наукові результати та практичні розробки дисертаційної роботи пройшли апробацію на 6 наукових конференціях, у тому числі на одній міжнародній:

L науково-технічна конференція підрозділів Вінницького національного технічного університету, Вінниця, 18 березня 2021 р.;

LI науково-технічна конференція підрозділів Вінницького національного технічного університету, Вінниця, 30-31 травня 2022 р.;

Проблеми вищої математичної освіти: виклики сучасності (2022), ВНТУ, Вінниця, 11-12 жовтня 2022 р.;

LII науково-технічна конференція підрозділів Вінницького національного технічного університету, Вінниця, 21-23 березня 2023 р.;

LIII науково-технічна конференція підрозділів Вінницького національного технічного університету, Вінниця, 20-22 березня 2024 р.;

Dynamical System Modeling and Stability Investigation (DSMSI-2023), Київ, 19-21 грудня 2023 р.

Публікації.

Основні результати дисертаційного дослідження викладені у 10 наукових працях (див. Додаток В). Опубліковано 4 статті у фахових журналах України зі спеціальності 126 [1], [3], [4], [5]; 1 стаття, що включена до наукометричної бази даних Scopus [2]; 5 тез доповідей на науково-практичних конференціях [6], [7], [8], [9], [10].

Документи інтелектуальної власності, що отримані за результатами дисертації.

Отримано свідоцтво про реєстрацію авторського права на твір (комп’ютерну програму) «Трансформація цільового класу для задачі сегментації з використанням U-GAN» [11].

Структура та обсяг дисертації.

Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаної літератури зі 131 найменування та 5 додатків. Загальний обсяг дисертації становить 208 сторінок, з них 155 сторінки основного тексту, який містить 109 рисунків та 5 таблиць.

РОЗДІЛ 1
АНАЛІЗ ПРОБЛЕМИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ ГЕНЕРУВАННЯ ШТУЧНИХ ДАНИХ

1.1 Задачі генерування даних

Генеративні моделі стали надзвичайно популярним напрямом досліджень та розробок у галузі штучного інтелекту, що підтверджується такими розробками як ChatGPT [12] (див. рис. 1.1), що може генерувати текст як справжній співрозмовник, або Midjourney [13] (див. рис.1. 2), що створює зображення будь-якої якості лише за вхідним реченням користувача. На сьогоднішній день генеративні моделі використовуються для розв'язання широкого спектру задач. Вагомі результати були досягнуті у наступних напрямках:

1). Генерування реалістичних зображень, зокрема у високій роздільній здатності [14], набуває дедалі більшого поширення для збільшення обсягу даних (аугментації) під час навчання або вдосконалення моделей машинного навчання. Це особливо важливо для наборів даних, де кількість прикладів окремих класів є обмеженою або спостерігається значний дисбаланс між класами. Така ситуація часто зустрічається при роботі з медичними даними, наприклад, у завданні класифікації медичних станів за рентгенівськими знімками грудної клітини [15]. Однак, варто пам'ятати, що робота з медичними даними потребує особливої обережності, оскільки генеративні «вигадують» ці дані [16]. Згенеровані зображення також можуть виступати і у якості самостійних креативних творів або слугувати джерелом натхнення для розробки нових концепцій, як-от ідей для дизайну посуду [17], меблів [18], логотипів [19] тощо.

2). Покращення роздільної здатності зображень [20]. Це може бути корисним, наприклад, для відновлення старих фотографій або навіть покращення якості МРТ і рентгенівських знімків. Проте, як уже було згадано, варто бути надзвичайно обачними при застосуванні цього підходу, оскільки генеративні

моделі часто «домислюють» відсутні деталі, спираючись на приклади, з якими стикалися під час навчання, особливо при значній зміні якості.

3). Зміна характеристик об’єкта. Наприклад, додавання окулярів, вус на зображенні, зміна інших рис обличчя, одягу, інших деталей. Хоча цей інструмент переважно використовується в розважальних і торговельних цілях [21], існують і більш серйозні сценарії його застосування. Наприклад, генеративна модель може бути використана для приховування ідентичності свідків або інших осіб у резонансних судових справах [22]. На жаль, такі технології також знаходять застосування у зловмисників для створення «діпфейків» [23], коли штучно генеруються фальшиві компрометуючі зображення чи відео шляхом заміни обличчя.

4). Створення зображень за вхідним текстовим описом [24], що також може використовуватись для створення трансформованих або навіть абсолютно нових даних.

5). Трансформація вхідного зображення в інше [25], як створення із супутникового знімка карту географічних або фізичних об'єктів, створення кольорового зображення з чорно-білого, перенесення характеристик однієї породи на іншу та подібні приклади.

6). Генерування мови, музики, звуків [26].

7). Конвертація текстів в аудіо, що дозволяє автоматизувати процес створення нових прикладів для тренування моделей для аудіо або генерувати аудіоконтент (аудіокниги) на основі звичайних текстів [27].

8). Зміна стилю аудіозаписів [28], як перетворення виконання на піаніно на гітарне звучання або трансформація чоловічого голосу в жіночій. Це може бути корисним, наприклад, для дубляжу фільмів, зберігаючи оригінальні голоси акторів [29].

9). Багато з прикладів, описаних вище, можуть застосовуватися і до відео як в комплексі, так і окремо [30].

10). Для текстів генеративні моделі можуть використовуватися для вирішення задач генерації [31] та реферування [32].

11). Окрім задач, пов'язаних зі створенням об'єктів, існують підходи до застосування генеративних моделей для виявлення аномальних даних [33] та кластеризації [34].

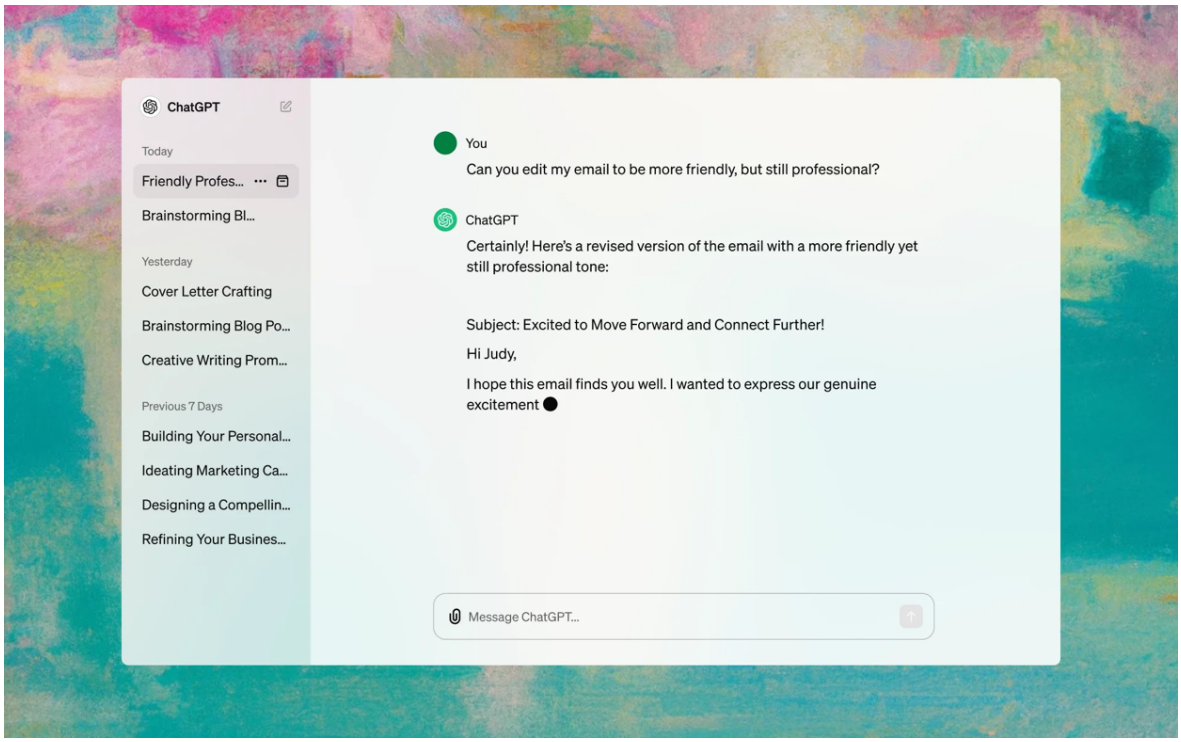


Рис. 1.1 ChatGPT – приклад використання генеративних моделей [11]

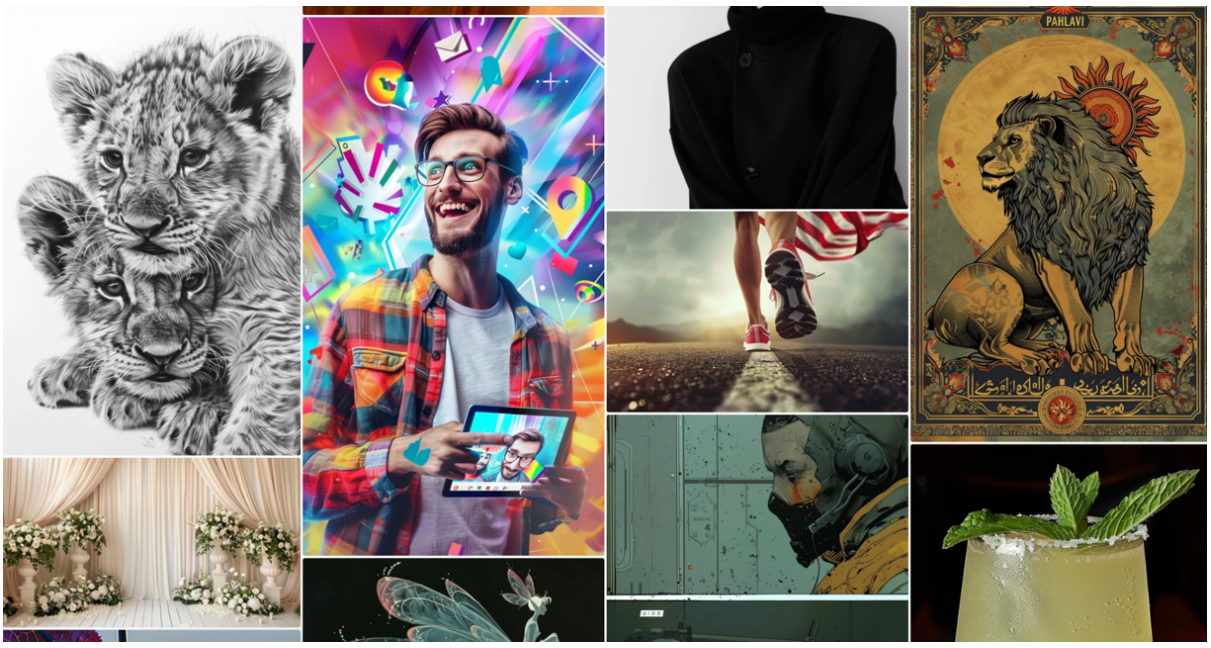


Рис. 1.2 Midjourney – приклад використання генеративних моделей [12]

1.2 Огляд основних моделей та методів генерування штучних даних

Основними типами генеративних моделей є автокодувальники, генеративні змагальні мережі, дифузійні моделі. В цьому розділі наведено опис цих технологій, їх переваги та недоліки.

1.2.1 Автокодувальники

Ці моделі складаються з двох штучних нейронних мереж: енкодера та декодера (див. рис. 1.3). Між ними розташовується код (часто називають bottleneck, або «вузьке місце»), що являє собою шар нейронів з меншою кількістю параметрів порівняно з іншими шарами. Основною метою такої моделі є зменшення розмірності даних, тобто кодування вхідної інформації у вектор ознак меншого розміру (код) та подальше відновлення (декодування) початкових даних з цього вектору [35]. Навчання автокодувальників відбувається за допомогою методу зворотного поширення помилки, а функція втрат ґрунтується на різниці між оригінальними даними (наприклад, зображеннями), поданими на вхід енкодера, та відновленими декодером.

З точки зору генерації даних, одним із головних недоліків класичних автокодувальників є те, що під час мінімізації функції втрат вони можуть суттєво змінювати прихований простір, навіть за наявності регуляризації [36]. У результаті, близькі точки в прихованому просторі можуть відповідати дуже різним даним, або можуть виникати області між точками, які не відображають реального розподілу даних. Цей недолік суттєво впливає на вибір класичних автокодувальників як моделей для генерації даних.

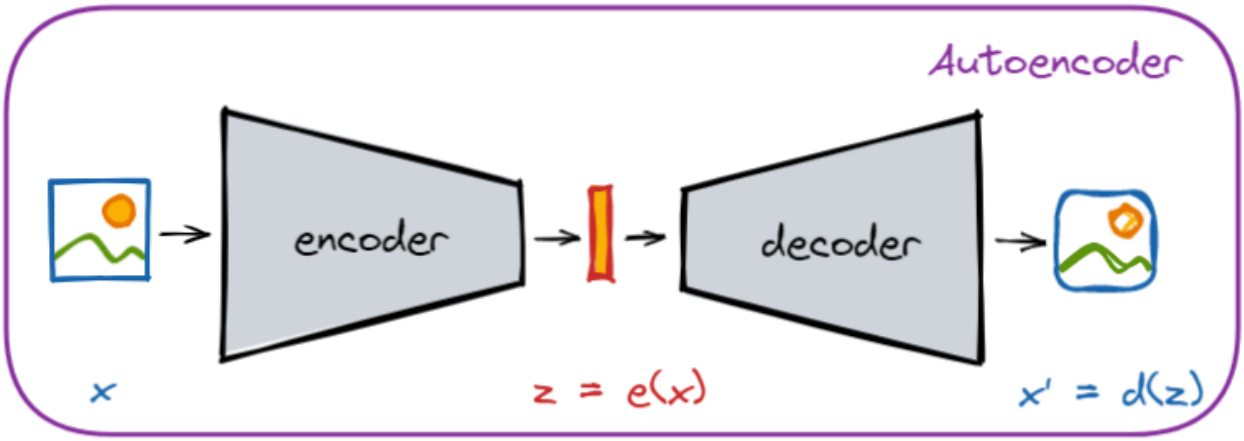


Рис. 1.3 Спрощена структура звичайного автокодувальника

Варіаційні автокодувальники [36] є вдосконаленим типом автокодувальників, який не має недоліків, властивих класичним версіям. Замість того, щоб для кожного вхідного прикладу встановлювати фіксовані значення коду, варіаційний автокодувальник моделює параметри розподілу, стискаючи вхідну інформацію до ймовірнісного латентного простору. В декодувальник надходить вектор, випадково обраний з цього розподілу.

В VAE застосовується так званий трюк перепараметрування, що дозволяє під час навчання моделі використовувати алгоритм зворотного поширення помилки, незважаючи на випадковий вибір значень вектора кодування з розподілу. Суть цього трюку полягає в тому, що стохастичність вибору значень вектора відокремлюється від основного шляху поширення сигналу, перетворюючи випадковий вибір в окремий вхід. Це виключає стохастичну складову з процесу зворотного поширення помилки (див. рис. 1.4).

Однак, можливий випадок, коли енкодер може створювати розподіли з нульовою дисперсією або з дуже далекими середніми значеннями, що може призводити до аналогічних проблем, як у класичних автокодувальниках. Для уникнення цього, у функцію втрат додають регуляризацію за допомогою дивергенції Кульбака–Лейблера [37] між згенерованим розподілом та стандартним нормальним розподілом. Це сприяє групуванню даних ближче до нормального розподілу, що дає змогу вибирати випадкові вектори для генерації нових даних і

виконувати інтерполяцію між двома векторами для створення проміжних варіантів.

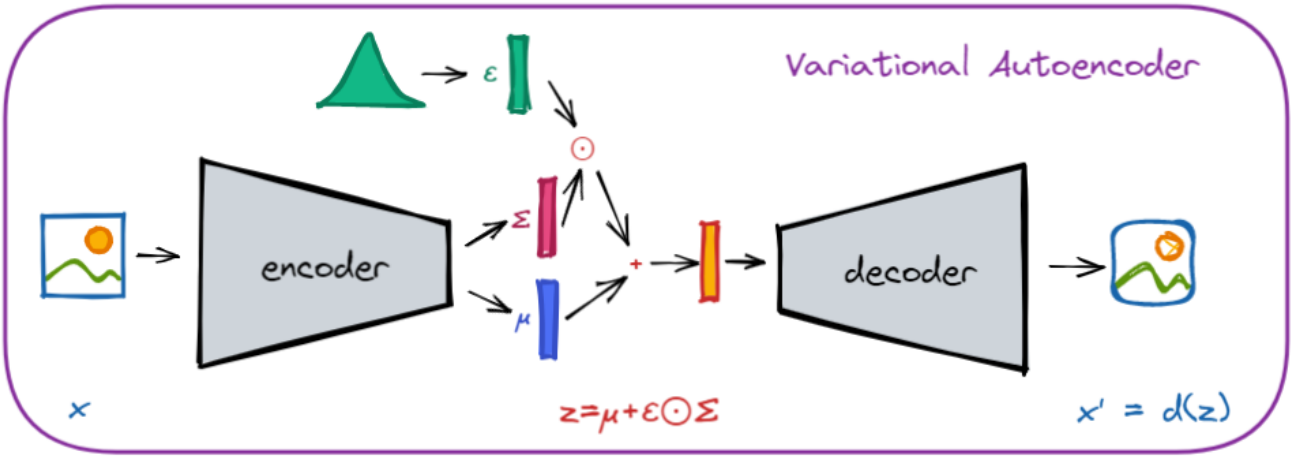


Рис. 1.4 Спрощена структура варіаційного автокодувальника

1.2.2 Генеративні змагальні мережі

ГЗМ були розроблені у 2014 році [38], і вони швидко здобули популярність. Основними компонентами ГЗМ (генеративних змагальних мереж) є дві окремі нейронні мережі: генератор і дискримінатор (у деяких варіаціях архітектури дискримінатор називають критиком), що показано на рисунку 1.5. Завдання генератора полягає в імітації розподілу реальних даних та генеруванні псевдореальних, або так званих фейкових («fake», що означає «підробка»), даних. Мета дискримінатора – визначити, чи дані, що подані на його вхід, є реальними чи згенерованими генератором. Процес навчання цієї моделі можна описати як мінімаксну гру між двома гравцями [39]: генератор намагається максимізувати помилку дискримінатора.

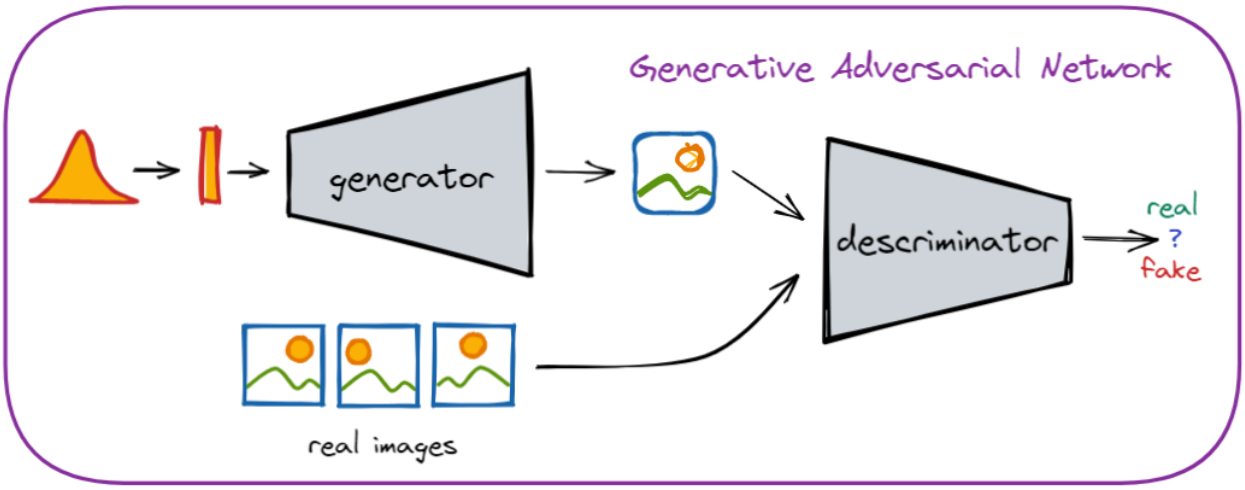


Рис. 1.5 Типова структура ГЗМ

Процесс навчання ГЗМ відбувається через зворотне поширення помилки і складається з двох етапів:

Спочатку відбувається навчання дискримінатора. Генератор створює набір фейкових екземплярів, який об'єднується з набором реальних даних. Ці обидва набори подаються на вхід дискримінатору, який намагається визначити, які дані є реальними, а які створені генератором. Мітки дискримінатора порівнюються з реальними мітками за допомогою функції втрат (наприклад, бінарної перехресної ентропії [39]), після чого дискримінатор оновлює свої ваги.

Після цього відбувається навчання генератора. Він створює набір згенерованих фейкових даних, який подається дискримінатору на оцінку (на цьому етапі реальні дані не використовуються). Функція втрат порівнює виходи дискримінатора з мітками, що вказують на те, що ці дані є справжніми (оскільки мета генератора – обманути дискримінатор і змусити його вважати фейкові дані реальними). На основі отриманого зворотного зв’язку ваги генератора оновлюються. Важливим є той факт, що хоча градієнт під час зворотного поширення помилки проходить через дискримінатор, його ваги на цьому етапі залишаються незмінними, це потірбно для розрахунку зворотного поширення помилки для генератора.

Щоб генератор кожного разу створював унікальні дані, на його вхід подається вектор згенерованого шуму з нормального розподілу (див. рис. 1.5), а

розмір цього вектору визначається як гіперпараметр [41] моделі. У випадках умовного генерування, коли потрібно створювати дані певного класу (з наявних класів), до цього вектора додається унітарний код. Код має значення «1» у позиції, що відповідає цільовому класу для генерування. Ця інформація також передається дискримінатору.

Оскільки в задачах генерації зображень на вході дискримінатора зображення з висотою H, шириною W та кількістю каналів C (зазвичай C=3), то унітарний код подається у вигляді додаткових каналів зображення. Кількість цих додаткових каналів відповідає кількості класів, і кожен клас представлений матрицею розміром H×W, заповненою нулями або одиницями. Тобто загальна кількість каналів входу до генератору становитиме:

$$N = C + K, \tag{1.1}$$

де K – це кількість класів.

У цьому випадку генератор має виконувати не лише завдання створення реалістичних даних, а й перевірку того, чи відповідають згенеровані дані закодованому класу. Наразі існує багато різновидів ГЗМ, які відрізняються архітектурно за типами моделей, використовуваних як генератори та критики, за типом функції втрат, що застосовується для оцінки якості моделей, та іншими характеристиками.

Якщо головне завдання автокодувальників полягає у стисненні інформації та відтворенні вхідних даних, то ГЗМ спрямовані на генерацію реалістичних даних із шуму. Навчання автокодувальників передбачає мінімізацію різниці між вхідними та вихідними даними, тоді як у ГЗМ один з основних компонентів – генератор – прагне відтворити розподіл реальних зображень.

Далі наведено опис найбільш відомих генеративних змагальних мереж.

Глибока Згорткова Генеративна Змагальна Мережа (Deep Convolutional Generative Adversarial Networks) [42].

Цей тип моделей поєднує як переваги згорткових нейронних мереж [43] (див. рис. 1.6), які є основними інструментами для задач обробки зображень, таких

як класифікація зображень [44], сегментація [45], і детекція об'єктів [46], так і класичних ГЗМ, побудованих на основі звичайних повноз'єднаних шарів [47].

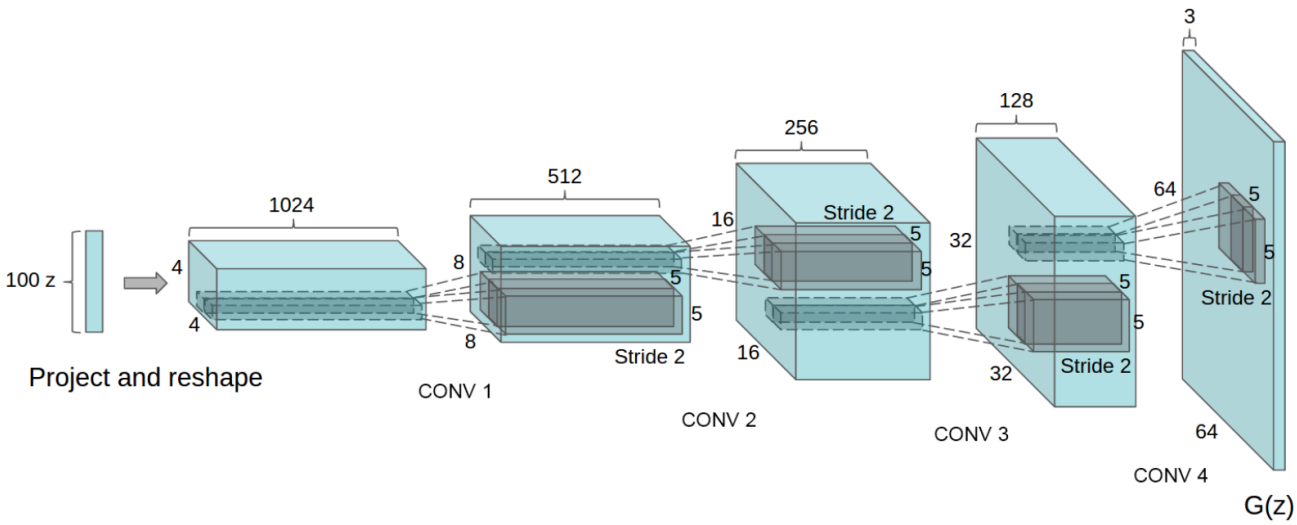


Рис. 1.6 Структура DCGAN [42]

Генеративна Змагальна Мережа Васерштейна (Wasserstein Generative Adversarial Networks) [48].

У ГЗМ Васерштейна замість класичного дискримінатора вводиться поняття критика. Завдання критика полягає в наданні числової оцінки реальним і фейковим зображенням та в якомога кращому розрізненні цих двох наборів даних, максимізуючи відстань Васерштейна [49]. Метою генератора, відповідно, є мінімізація цієї дистанції. Заміна функції втрат на Васерштейнську відстань зумовлена тим, що в функції втрат що використовуєтсья в звичайних ГЗМ (бінарній перехресній ентропії) може виникати проблема затухання градієнта [50] – коли дискримінатор надто впевнено розпізнає фейкові зображення, і градієнт активації досягає плато. WGAN вирішує цю проблему (див. рис. 1.7).

Автори WGAN показали, що для стабільного навчання критик має задовольняти умову обмеження 1-Ліпшиц (градієнт у кожній точці не повинен перевищувати одиниці). Для цього в класичній реалізації використовується відсікання ваг, але цей метод має декілька недоліків. Якщо параметр відсікання досить великий, критик може потребувати багато часу для досягнення своїх меж,

що уповільнює процес навчання. Якщо ж відсікання занадто мале, це може призвести до зникнення градієнтів, особливо коли в моделі багато шарів або не застосовується пакетна нормалізація [51]. Використання цього методу зумовлене його простотою та тим, що він дозволяє досягти прийнятних результатів у базовій версії моделі.

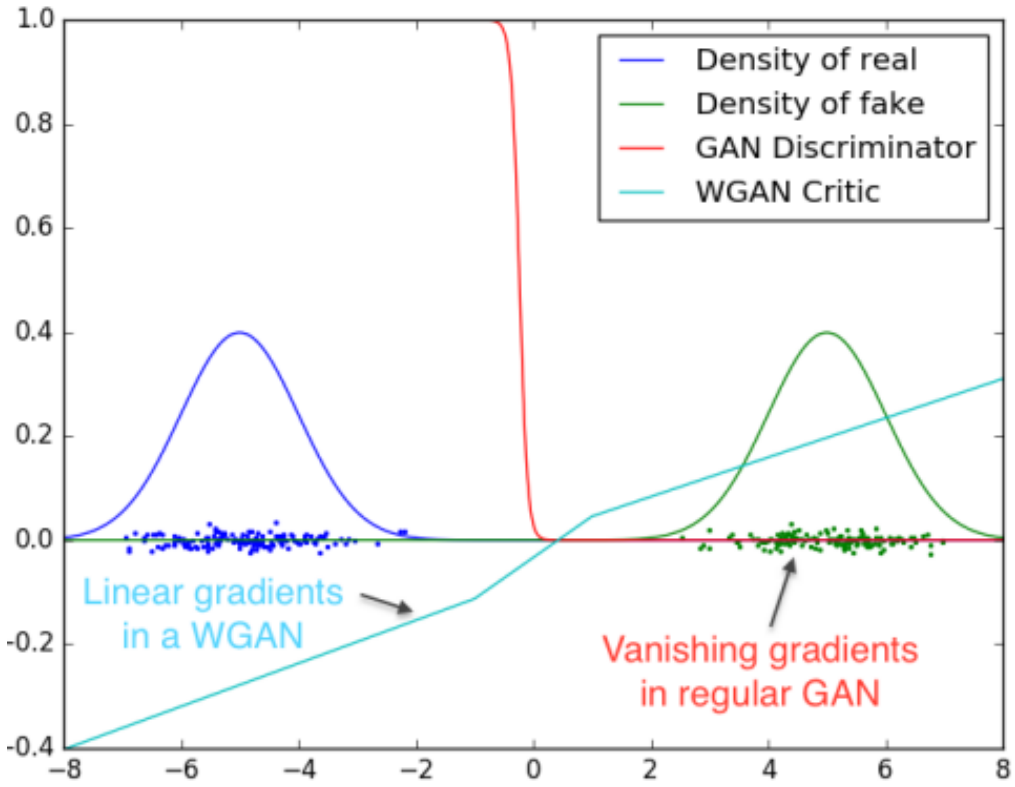


Рис. 1.7 Різниця між градієнтами у класичних ГЗМ та WGAN [48]

ГЗМ Васерштейна з градієнтним штрафом (Wasserstein GAN with gradient penalty) [52].

WGAN-GP дає штраф моделі, замість прийому відсікання ваги, який примушує градієнти дискримінатора бути близькими до 1 (це рівнозначно виконанню ліпшицевої умови). Градієнтний штраф обчислюється на лінійній інтерполяції між реальними та згенерованими зразками, що гарантує, що градієнт дискримінатора залишається стабільним під час навчання.

Велика ГЗМ (BigGAN) [53].

Наведемо особливості, що використовуються в даному типі ГЗМ:

- використання максимально можливого розміру пакету даних під час навчання дозволило не тільки покращити показники таких метрик, як Inception Score та Fréchet Inception Distance, але й уникнути проблеми колапсу режиму;

- збільшення кількості параметрів моделі як по ширині (кількість параметрів у шарі), так і по довжині (кількість шарів) також сприяло підвищенню якості генерованих даних;

- застосування функції завісних втрат, яка зазвичай використовується в методі опорних векторів [54], дало змогу покращити процес навчання моделі;

- завдяки умовній класовій пакетній нормалізації інформація про клас передається до генератора, що забезпечує ефективне навчання моделей для генерації даних різних класів;

- використання ортогональної регуляризації допомогає застосовувати до генератора трюк відсікання, який контролює компроміс між точністю та різноманітністю даних. Трюк відсікання передбачає обрізку вектора випадкового шуму, відрізаючі значення, що перевищують певний поріг: чим нижчим є це порогове значення, тим кращіми, але менш різноманітними будуть отримуватись згенеровані дані.

Прогресивно зростаюча ГЗМ (Progressively Growing GAN) [55].

Ця архітектура генеративної змагальної мережі надає можливість моделі поступово збільшувати розмір генерованих зображень на етапі тренування. Цей підхід був запропонований для того, щоб покращити якість зображень і стабільність навчання ГЗМ, особливо для зображень високороздільної якості. Наприклад, зображення набору CELEBA [56], що мають розмірність 1024x1024 пікселі. Схема навчання показана на рисунку 1.8.

На початкових етапах тренування модель генерує зображення низької роздільної здатності (наприклад, 4x4 пікселів). У міру прогресу тренування, кількість шарів у мережі збільшується, що дозволяє генерувати зображення з вищою роздільною здатністю (8x8, 16x16 і так далі до, наприклад, 1024x1024). Архітектура як генератора, так і дискримінатора змінюється поступово. Спочатку вони мають менше шарів, а з підвищенням роздільної здатності до мереж

додаються нові шари, що дозволяє їм обробляти дедалі більші зображення. Під час додавання нових шарів у мережу використовуються методи інтерполяції, що робить процес безперервним. Це дозволяє уникнути різких змін у розподілі даних і зменшує ризик нестабільності.

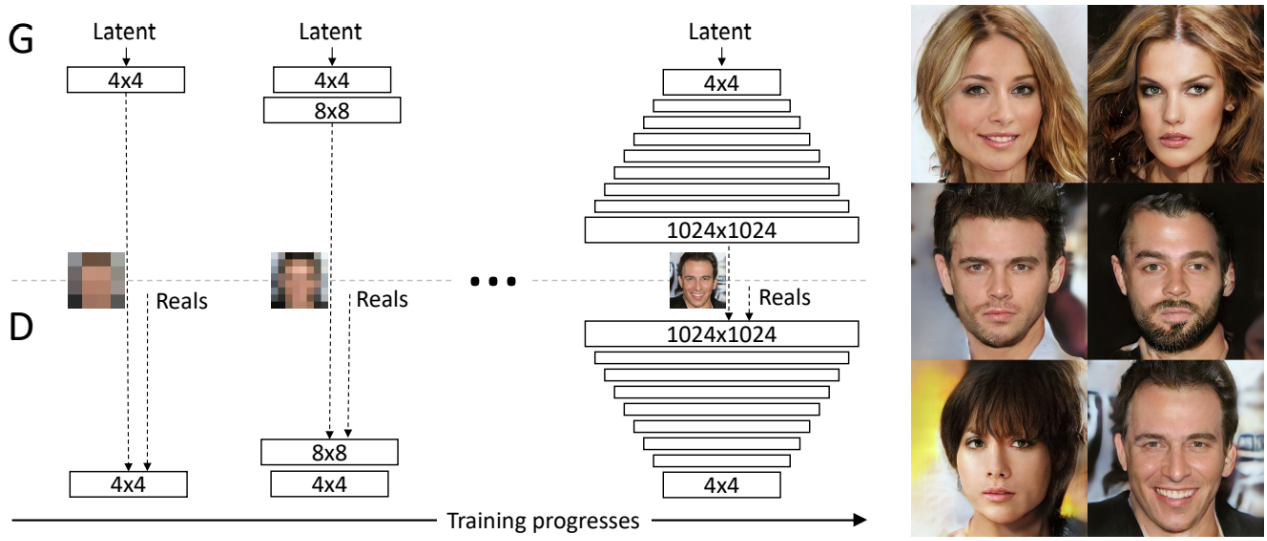


Рис. 1.8 Схема прогресивного навчання [55]

StyleGAN [57].

Однією з ключових особливостей цієї моделі, завдяки чому вона отримує гарні результати, є використання прогресивного зростання, яке описано вище. Цей підхід дає змогу спочатку опрацьовувати великомасштабні структури зображення, а потім переходити до більш детальних елементів, замість одночасного навчання всієї моделі.

Ще однією важливою особливістю цієї архітектури є мережа відображення шуму, що реалізує повноз'єднаний перцептрон, який перетворює вектор випадкового шуму на новий вектор стилю. Цей вектор подається до кожного з блоків основної частини моделі, тоді як у звичайних ГЗМ вектор поступає лише на перший шар.

Важливо також відзначити використання адаптивної нормалізації екземплярів (AdaIN), яка дозволяє передавати стильову інформацію з вектора стилю, створеного мережею відображення шуму, до зображення, що генерується.

На початкових етапах, коли розмір зображення невеликий, використовуються глобальні риси, тоді як на завершальних етапах додаються детальніші характеристики.

Останньою важливою розробкою є мікс стилів, коли для зображення генерується не один вектор стилю, а декілька. Кожен з них подається в різні блоки генератора, що додає більше варіативності в модель. Усі ці компоненти показані на рисунку 1.9.

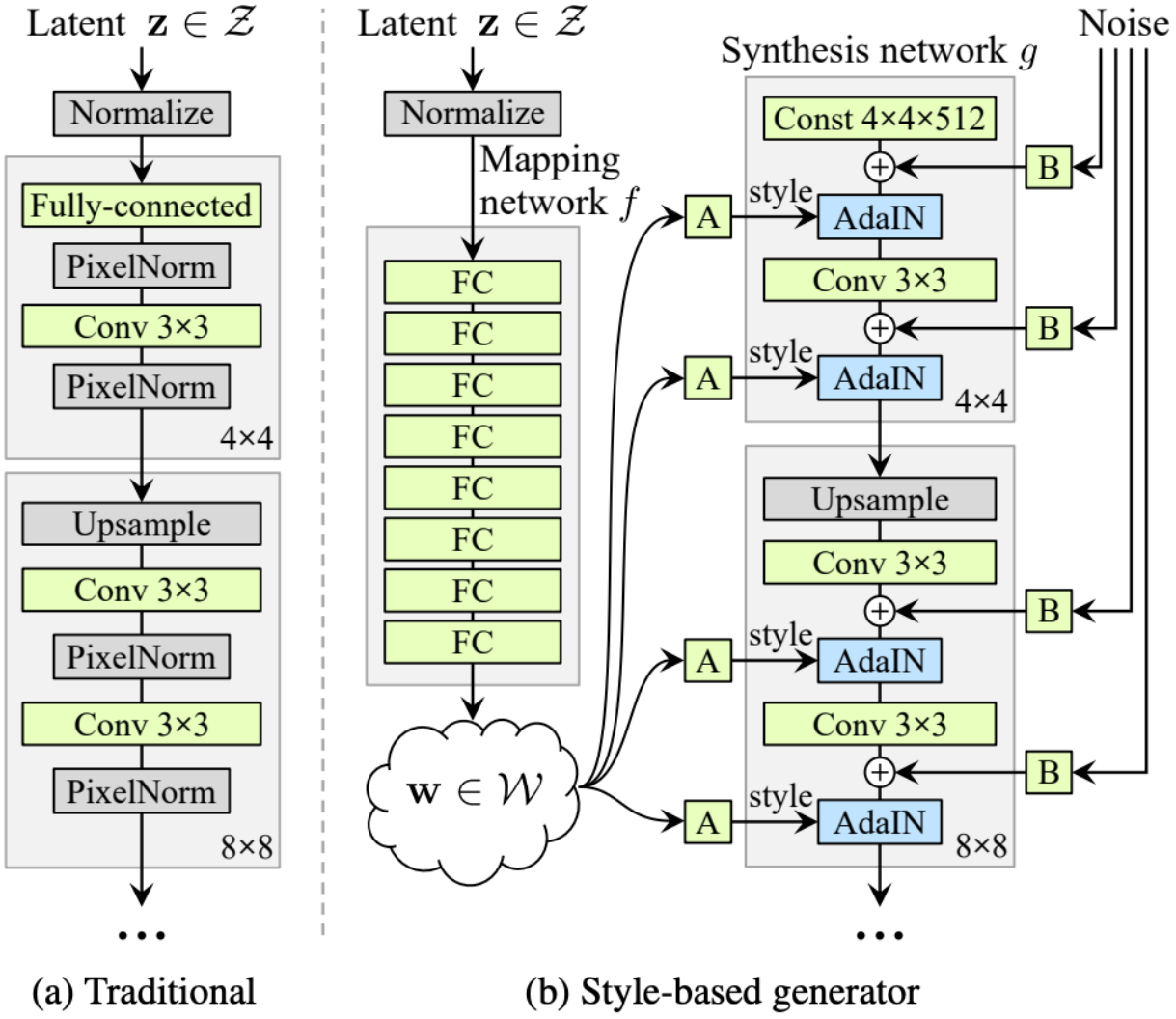


Рис. 1.9 Архітектура StyleGAN [57]

StyleGAN2 [58].

Ця модель, як видно з назви, побудована на основі StyleGAN, проте деякі з попередніх особливостей зазнали значних змін. Зокрема, адаптивна нормалізація

екземплярів, яка може спричиняти краплеподібні артефакти (приклад показано на рисунку 1.10), була модифікована шляхом введення демодуляції ваг для уникнення цих недоліків.

Рис. 1.10 Приклади краплеподібних артефактів [58]

Прогресивне зростання реалізоване за допомогою механізму пропуску з'єднань, що дозволяє мережі прогресивно навчатися, використовуючи всі роздільні здатності зображення одночасно, при цьому спочатку надається пріоритет меншим зображенням з плавним переходом до більших, без змін у топології мережі.

Серед інших особливостей StyleGAN2 варто зазначити:

- регуляризацію довжини шляху, яка додається до функції втрат, забезпечуючи більш плавну інтерполяцію латентного простору. Це сприяє

покращенню якості зображень і дозволяє створювати більш якісні проміжні кадри під час інтерполяції між двома зображеннями.

- лініву регуляризацію (lazy regularization), коли регуляризаційна частина функції втрат застосовується окремо від основної і виконується рідше, що допомагає оптимізувати процес навчання моделі.

1.2.3 Дифузійні моделі

Дифузійні моделі (DDPM) [59] – це передова техніка в галузі генеративних моделей, зокрема в області синтезу та маніпулювання зображеннями. Вона використовує принципи дифузійних процесів для генерації високоякісних зображень із шуму, поступово їх вдосконалюючи для створення цілісних і реалістичних візуалізацій. Процес дифузії можна розділити на:

- Пряма дифузія: Починаючи з реального зображення, Гауссівський шум поступово додається протягом декількох кроків, перетворюючи зображення в чистий шум (див. рис. 1.11).
- Зворотна дифузія: Процес зворотний, де модель (наприклад, нейронна мережа), починаючи з чистого шуму поступово зменшує шум для отримання високоякісного зображення (див. рис. 1.12).

Рис. 1.11 Приклад прямої дифузії

Рис. 1.12 Приклад зворотної дифузії

І прямий, і зворотний процес відбуваються протягом деякої кінцевої кількості кроків (автори DDPM використовують 1000). Процес починається з нульового кроку, де береться реальне зображення з розподілу даних (наприклад, зображення кота з ImageNet [60]). Прямий процес відбирає деякий шум із розподілу Гауса і додає його на вхідне зображення. На кожному кроці вибирається новий випадковий шум і додається до зображення попереднього часового кроку. Враховуючи досить велику кількість кроків і правильне відбирання додаткового шуму на кожному кроці часу, отримується те, що називається ізотропним розподілом Гауса.

З точки зору архітектури нейромережі для відокремлення зображення від шуму, автори DDPM вибрали U-Net [61], (яка на той час досягла найкращих результатів для сегментації медичних зображень). Ця мережа, як і будь-який автокодувальник, складається з вузького місця посередині, яке гарантує, що мережа вивчає лише найважливішу інформацію.

Розглянемо найпопулярніші дифузійні моделі.

Stable Diffusion [62].

Однією з найбільш великих переваг Stable Diffusion на відміну від попередніх дифузійних моделей є латентний простір. Латентний простір створюється за допомогою великих попередньо навчених автокодувальників, які забезпечують високу якість реконструкції зображень. Вони перетворюють вхідне зображення в вектор в меншій розмірності, за рахунок чого навчання моделей дифузії у латентному просторі зменшеної вимірності дозволяє значно зменшити обчислювальні витрати.

Другим ключовим аспектом є впровадження шарів перехресної уваги в модель денойзінгу, автори перетворюють дифузійні моделі на потужні та гнучкі генератори для загальних вхідних даних, що можуть приймати текстові підказки, обмежувальні прямокутники, ключові точки (див. рис. 1.13). Разом з синтезом із високою роздільною здатністю зображень, що стає можливим використовуючи згортковий алгоритм, така дифузійна модель стає потужним інструментом контрольованого генерування.

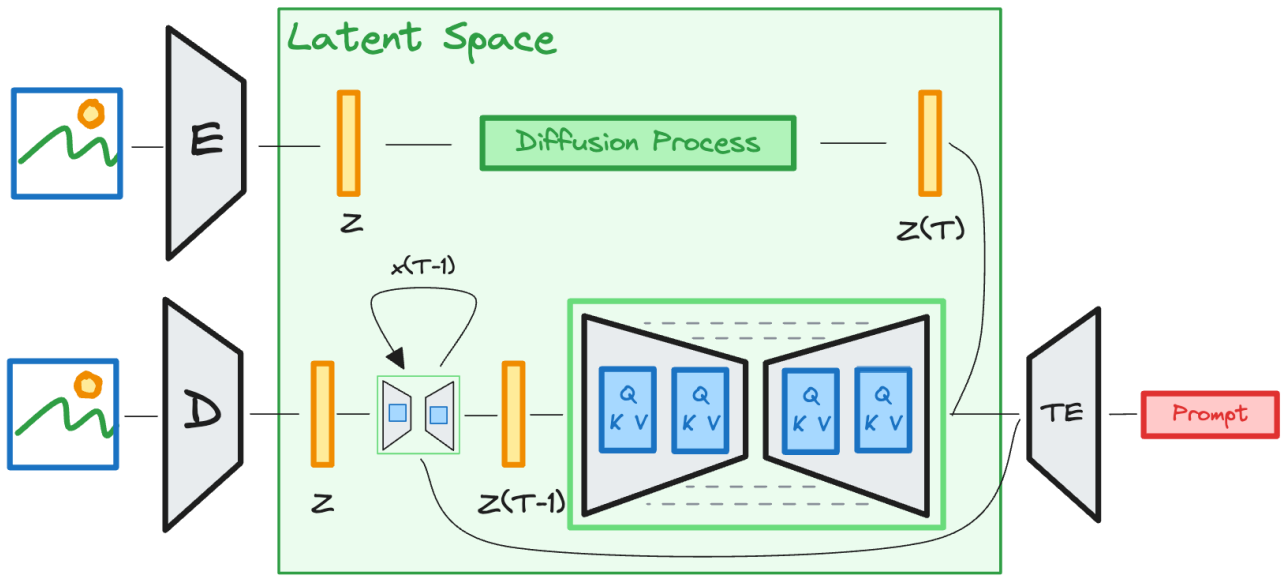


Рис. 1.13 Спрощена структура Stable Diffusion

Stable Diffusion 2.0 [63].

Stable Diffusion 2.0 включає надійні моделі перетворення тексту в зображення, навчені за допомогою абсолютно нового кодувальника тексту OpenCLIP [64], розробленого LAION за підтримки Stability AI, який значно покращує якість створюваних зображень порівняно з попередніми версіями підходу. Моделі перетворення тексту в зображення в цій версії можуть створювати зображення з роздільною здатністю за замовчуванням 512x512 пікселів і 768x768 пікселів.

Stable Diffusion 2.0 також включає модель Upscaler Diffusion, що підвищує роздільну здатність зображень у 4 рази. На рисунку 1.14 наведено приклад моделі, яка покращує згенероване зображення з низькою роздільною здатністю (128x128) до зображення з вищою роздільною здатністю (512x512). У поєднанні з моделями перетворення тексту в зображення Stable Diffusion 2.0 може створювати зображення з роздільністю 2048x2048.

Рис. 1.14 Зліва: зображення з низькою роздільною здатністю. Справа: зображення з високою роздільною здатністю, створене Upscaler моделлю [63]

Stable Diffusion 3.0 [65].

Для генерації тексту в зображення модель має враховувати обидві модальності: текст і зображення. Ось чому нова архітектура називається Multimodal Diffusion Transformer, посилаючись на її здатність обробляти кілька модальностей. Як і в попередніх версіях Stable Diffusion, використовується попередньо підготовлені моделі для отримання відповідних представлень тексту та зображень. Зокрема, використовується три різні мережі для вбудовування тексту – дві моделі CLIP і T5 [66] – для кодування представлень тексту та покращену модель автокодування для кодування токенів зображень.

Архітектура SD3 базується на дифузійному трансформері (Diffusion Transformers) [67]. Оскільки вбудовування тексту та зображення концептуально досить різні, використовується два окремих набори ваг для двох модальностей. Як показано на рисунку 1.15, це еквівалентно наявності двох незалежних трансформерів для кожної модальності, але об’єднання послідовностей двох модальностей для операції уваги, так що обидва представлення можуть працювати у своєму власному просторі, але враховують інше.

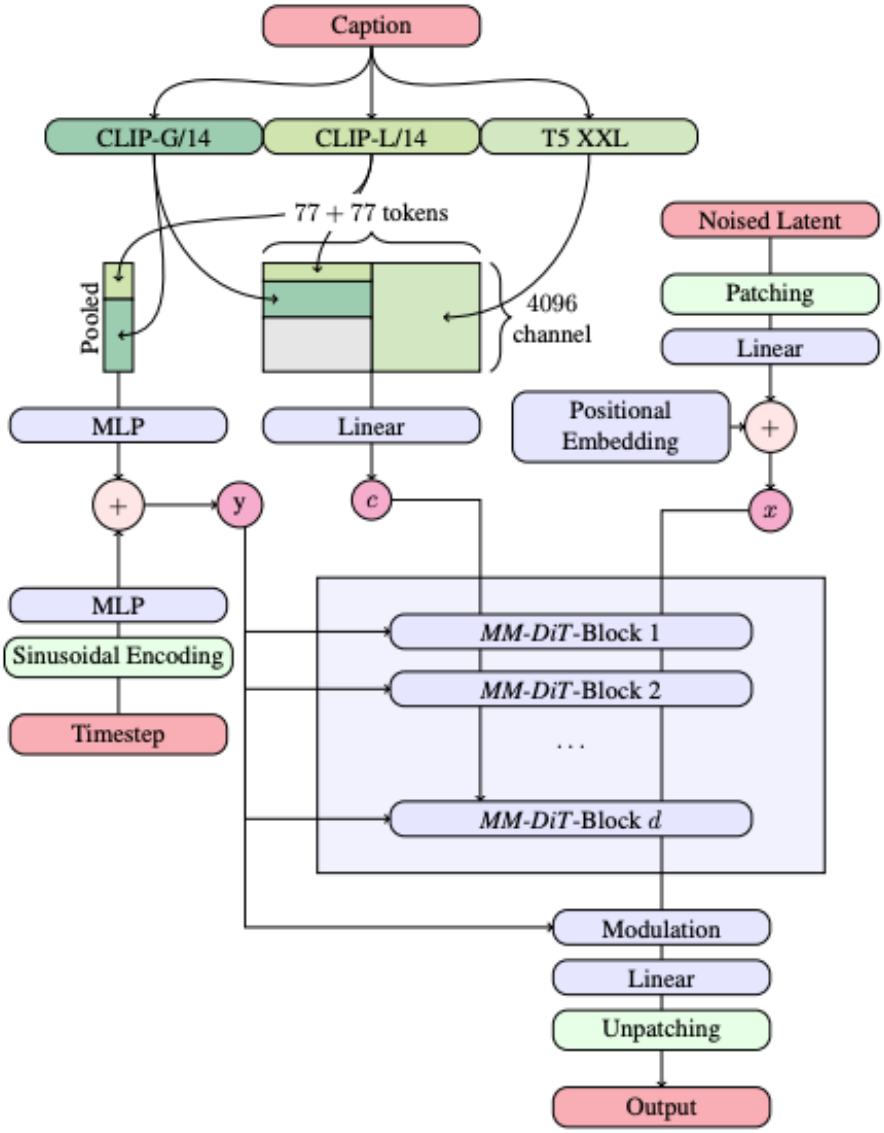


Рис. 1.15 Архітектура нових компонентів SD3 [65]

Stable Diffusion XL [68].

SDXL використовує втричі більшу основу UNet у порівнянні з попередніми версіями Stable Diffusion. Це досягнуто шляхом додавання більшої кількості блоків уваги та збільшення контексту крос-уваги за рахунок використання ще одного текстового енкодера.

Для покращення візуальної якості згенерованих зображень використовуються додаткові моделі рефайнінгу [69], які здійснюють процес зашумлення та розшумлення.

Замість того, щоб відкидати зображення меншої роздільної здатності, модель пристосовується на початкову роздільну здатність картинок, що дозволяє

використати більше даних під час навчання. Модель тренується на зображеннях різних розмірів і пропорцій, що дозволяє їй генерувати зображення у різних форматах (наприклад, ландшафтний або портретний).

Використання текстових підказок.

Великою перевагою Stable Diffusion моделей є можливість створення деталізованого зображення на основі текстового опису-підказки (prompt) [9]. Але ці підказки можна застосовувати і для інших завдань, наприклад, для замальовування, домальовування, перетворювання та редагування зображення.

Окрім текстового опису, модель використовує ще ряд інших параметрів, які суттєво впливають на результат. Щодо самого опису-підказки, то використання різних ключових слів в описі одного і того ж об’єкта також часто призводить до зовсім різних результатів генерування. На рисунку 1.16 показано як впливає додавання слів “photorealistic”, “highly detailed”, “4K”, “realistic” на створення зображень кота при фіксованих значеннях решти параметрів SD моделі.

Рис. 1.16 Приклад зміни результату генерування Stable Diffusion при зміні опису-підказки

Іншим важливим параметром у моделі Stable Diffusion є cfg_scale, при малих значеннях якого вхідний опис майже не впливає на результат, а при великих – результат стає більш нереалістичним. На рисунку 1.17 показано результат генерування з вхідним описом “cat” та значенням cfg_scale від одного до дев’ятнадцяти з кроком 2. Видно, що зображення зі значеннями від 7 до 11 мають найбільш реалістичний вигляд, а 15-19 – набувають рис твору образотворчого мистецтва.

Рис. 1.17 Результати генерування Stable Diffusion для різних значень параметра cfg_scale

1.2.4 Висновки щодо основних типів моделей генерування даних

Автокодувальники – це клас нейронних мереж, які використовуються для навчання ефективних кодових представлень даних. Вони складаються з двох основних компонентів: кодера та декодера. Основна мета автокодувальника полягає у тому, щоб відобразити вхідні дані в компактну внутрішню репрезентацію (зазвичай меншої розмірності), а потім відновити їх на виході з мінімальними втратами якості. Кодер стискає інформацію до латентного простору, що зберігає найважливіші характеристики даних, тоді як декодер намагається відновити початкові дані на основі цього стислого представлення.

Автокодувальники мають широкий спектр застосувань. Їх використовують для: стиснення даних, кодування та декодування, зменшення розмірності, видалення шуму з даних.

Попри їхню універсальність та простоту, класичні автокодувальники мають обмеження у здатності створювати високоякісні та різноманітні згенеровані дані. У цьому аспекті їх перевершують більш потужні генеративні моделі, що наведені далі.

Генеративні змагальні мережі – це клас нейронних мереж, що використовуються для генерування нових, реалістичних даних на основі вхідного розподілу реальних зображень. ГЗМ складаються з генератора та дискримінатора, які «змагаються» один з одним під час навчання. Генератор намагається створити нові зразки даних, які нагадують реальні, а дискримінатор намагається розділити згенеровані зразки від справжніх.

Однак генеративні змагальні мережі часто схильні до проблеми нестабільного навчання, де дисбаланс між генератором і дискримінатором може призвести до нерівномірного поліпшення результатів окремої моделі і привести до перенавчання, колапсу моди, поганій якості.

Дифузійні моделі – це клас генеративних моделей, що використовуються для поступової генерації нових зразків даних через процес оберненого дифузійного перетворення. В основі цих моделей лежить ідея поступового додавання шуму до даних, що зрештою перетворює їх у чистий шум. Потім модель навчається відновлювати дані шляхом зворотного процесу, усуваючи цей шум на кожному етапі, що дозволяє отримати нові реалістичні зразки.

Основною перевагою дифузійних моделей є їхня здатність генерувати високодеталізовані та реалістичні зображення, особливо в тих випадках, коли важлива плавність та послідовність результатів. В порівнянні з генеративними змагальними мережами, дифузійні моделі менш схильні до проблем нестабільності під час навчання. Однак ці моделі характеризуються складною архітектурою нейронних мереж та високими обчислювальними витратами як під час тренування, так і при застосуванні. Це зумовлено тим, що процес генерації вимагає багаторазових ітерацій для видалення шуму з вхідного зображення, а обчислення градієнтів відбувається на кожному етапі корекції шуму.

Кожен із розглянутих підходів до генерації даних має свої унікальні переваги та недоліки. Оскільки найкращі результати досягаються за допомогою генеративних змагальних мереж і дифузійних моделей, а для експериментів із дифузійними моделями потрібні дуже високі обчислювальні потужності як для навчання, так і для використання, було вирішено зосередитися на генеративних

змагальних мережах. Хоча вони мають певні недоліки, з ними можна проводити подальші експерименти з метою виправлення цих недоліків, що відкриває можливості для вдосконалення архітектури та методів навчання в рамках цього дослідження.

1.3 Огляд методів оцінювання результатів генерування

Оцінка генеративних мереж є непростим завданням, оскільки згенеровані дані можуть бути дуже різноманітними, а їх якість не завжди очевидна. З одного боку, важливо враховувати здатність моделі генерувати реалістичні та унікальні зразки, які не є просто копіями тренувальних даних. З іншого боку, необхідно забезпечити, щоб згенеровані дані відповідали певним критеріям, таким як консистентність стилю, різноманітність і коректність [5], [6].

Існує кілька підходів до оцінювання генеративних мереж, кожен з яких має свої переваги та обмеження. Вибір методів оцінювання залежить від конкретного завдання, типу згенерованих даних та вимог до них. Методи оцінювання можна розділити на дві категорії: автоматичні та із залученням людини.

1.3.1 Автоматичні методи оцінювання

Одна з найвідоміших метрик – Inception Score [70]. Її реалізація базується на однойменної нейронної мережі InceptionNet [71], що навчена на ImageNet наборі даних. В основі цієї метрики є два критерія: вірність, що відповідає за якість створених даних та різноманітність, що відповідає за те, на скільки різні дані може створювати модель.

Для оцінки згенерованих даних (у даному випадку тільки зображення) перш за все вони подаються на вхід до нейронної мережі InceptionNet. На виході отримуються ймовірності приналежності даних до одного з тисячі класів що представлені у ImageNet (див. рис. 6).

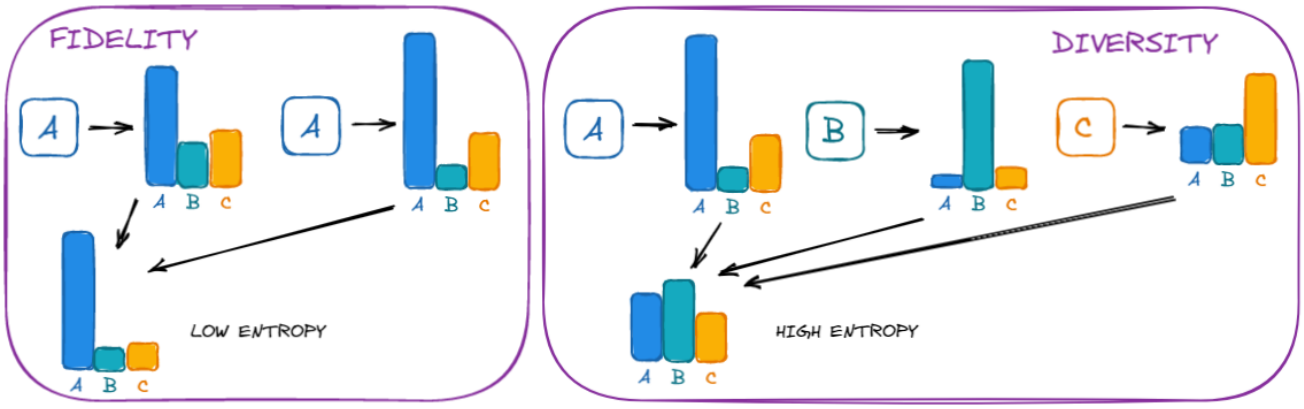


Рис. 1.18 Принцип роботи Inception Score

Щоб мати гарні показники вірності конкретні семпли згенерованих даних повинні належати тільки до одного вихідного класу (класифікаційна модель повинна чітко розуміти що зображено на картинці серед того що вона знає), тобто мати низьку ентропію. Для критерію різноманітності потрібно щоб набір згенерованих даних охоплював як можна більшу кількість класів, тобто мав високу ентропію по класам. Комбінація цих метрик утворює фінальну оцінку: чим більше буде різниця в цих розподілах тим вище буде Inseption Score (IS), це вимірюється за допомогою міра розходження Кульбака – Лейблера (див. рис. 1.19).

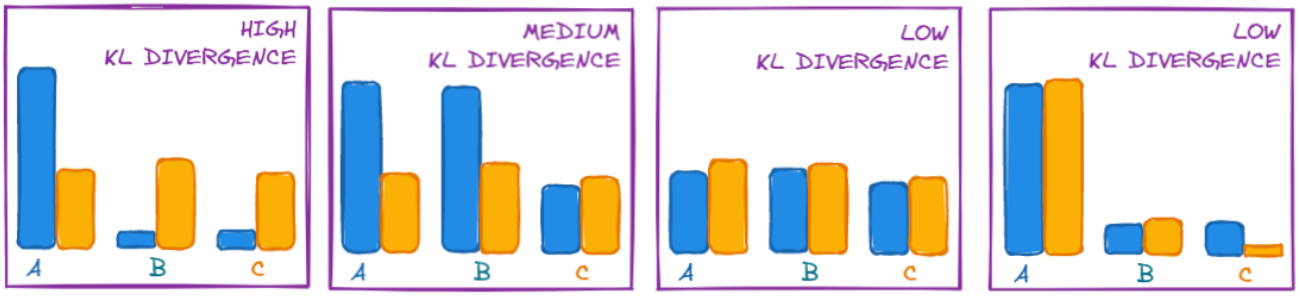


Рис. 1.19 Оцінка близькості розподілів

Одним із недоліків IS є невикористання інформації про реальних даних та їх розподілів. На цій ідеї побудована метрика Fréchet Inception Distance (FID) [72]. FID показує, наскільки розподіли реальних та згенерованих даних близькі між собою за відстанню Фреше.

Щоб порівнювати розподіли, кожне зображення має бути представлено у вигляді вектора. Оскільки не має сенсу порівнювати між собою пікселі картинок у підході, як і у IS, використовується нейронна мережа InceptionNet для отримання абстрактних векторів ознак з останніх шарів моделі, оскільки ці шари описують важливі з точки зору моделі ознаки вхідних зображень різних об’єктів. Для таких реальних та згенерованих векторів даних розраховуються значення середнього та коваріації. Далі отримані розподіли порівнюються між собою через відстань Фреше, що схематично показано на рисунку 1.20.

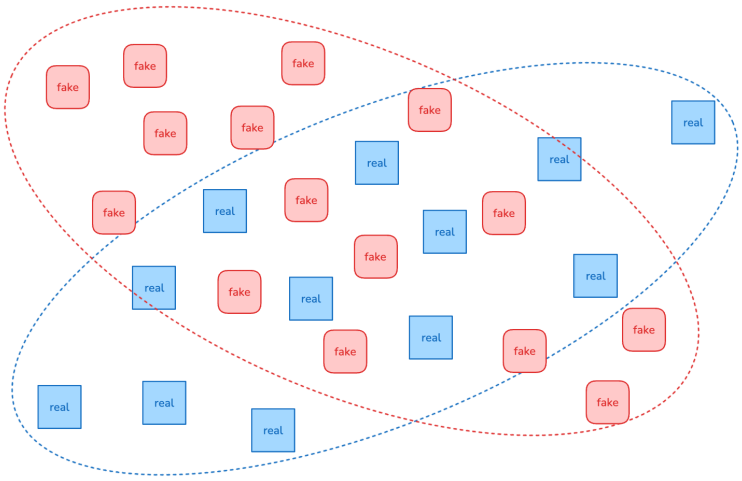


Рис. 1.20 Схематичне порівняння розподілів даних

Влучність (precision) та повнота (recall) [73] зазвичай застосовуються як вагомі метрики у задачах класифікації для підрахунку частки позитивних зразків серед знайдених та частки загального числа знайдених позитивних зразків відповідно. Влучність та повнота також можуть застосовуватись і при оцінці згенерованих даних.

На рисунку 1.21b показано, що влучність – ймовірність того, що згенеровані дані будуть у межах розподілу реальних даних, що є визначенням вірності. Якщо більшість згенерованих даних попадають у область, що займають реальні дані, це може свідчити про гарну якість зображень. Відповідно, якщо згенеровані дані знаходяться поза межами реальної області, це може свідчити про погану якість цих даних.

На рисунку 1.21c показана оцінка повноти, що відповідає ймовірності того, що реальні дані будуть входити у згенерований розподіл, за що відповідає різноманітність. Якщо всі реальні дані знаходяться у межах згенерованого розподілу, то це є свідченням того, що згенеровані дані мають дуже різні варіації. Якщо згенеровані охоплюють тільки маленьку частину реальних даних, то це свідчить про те, що такі дані практично не мають різноманітності та всі дуже схожі між собою, що також називається колапс мод (mode collapse) (див. рис. 1.22).

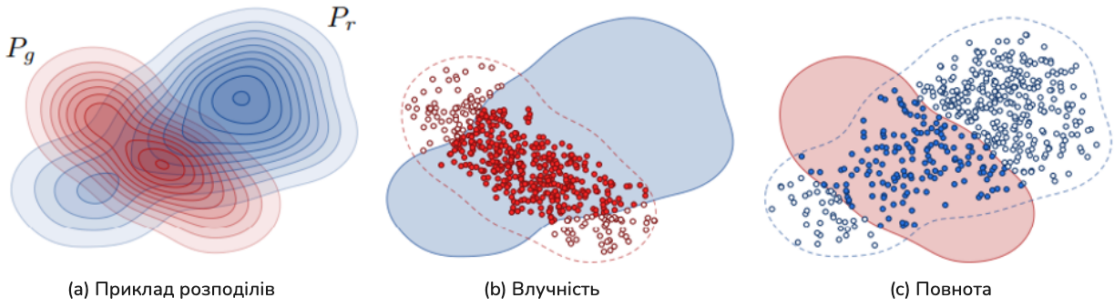


Рис. 1.21 Влучність та повнота. P_g – згенеровані дані, P_r – реальні дані

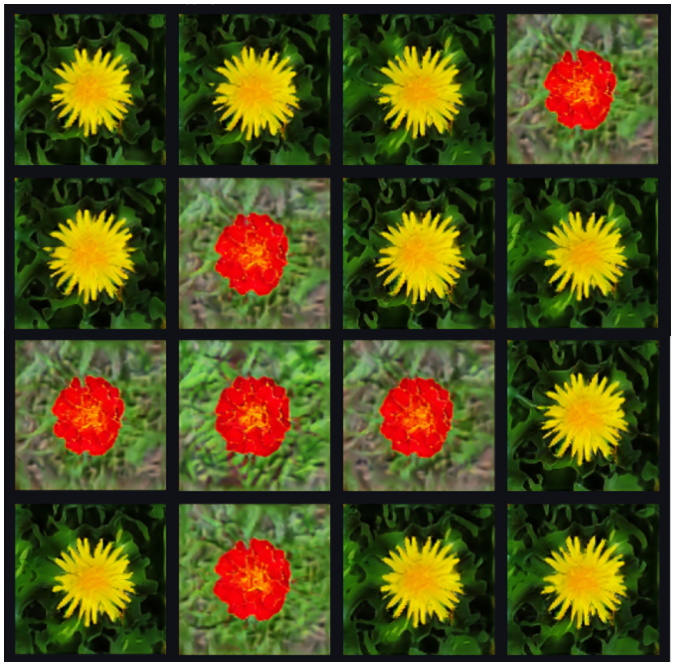


Рис. 1.22 Приклад явища mode collapse

Щільність (density) та покриття (coverage) це метрики, що є аналогами влучності та повноти, але вони вирішують ряд проблем що мають їх попередники, а саме: мають можливість ідентифікувати ідентичні реальний та згенерований розподіли показуючи значення близькі до 100 відсотків; є стійкими до аномалій як у згенерованих даних, так і у реальних; мають чіткий та простий алгоритм вибору гіперпараметрів оцінювання [74].

Порівняння щільності та покриття з влучністю та повнотою показано на рисунку 1.23. Із рисунку 1.23a видно, що якщо у реальних даних є аномальні семпли, то вони можуть дуже розширити покриття простору реальних даних та захватити погано згенеровані дані при цьому піднявши влучність до 100%. Щільність у такій ситуації є кращим показником, так як ця метрика враховує у скільки зон попадає згенеровані дані відносно кількості реальних даних у цій зоні (кількість реальних даних у кожній області є одним із гіперпараметров алгоритму, зони збільшуються поки кількість реальних даних не досягне заданого значення, на рисунку 1.23a це значення дорівнює двом). У прикладі чотири з п’яти фейкових семплів покривається лише однієї реальною областю, що є аномальною.

Ліва частина рисунку 1.23b показує, що якщо генеративна модель має погану якість, але гарну різноманітність, то згенеровані дані можуть дуже сильно захватувати простір та піднімати метрику повноти до 100 відсотків. У той час метрика покриття створює області на основі реальних семплів та підраховує який відсоток реальних семплів має як мінімум один фейковий семпл у своїй області. Із рисунку видно, що оцінка згенерованих даних вийшла коректною та низькою.

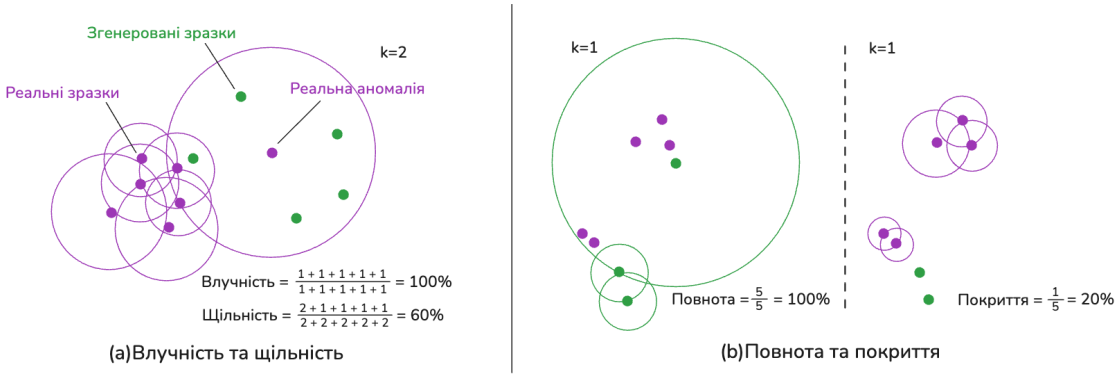


Рис. 1.23 Щільність та покриття

Метрики GAN-train та GAN-test [75] використовуються для вимірювання різноманітності та реалізму даних відповідно. В першому підході створюється умовна генеративна мережа на основі тренувальних даних, та завдяки моделі створюється новий набір синтетичних даних (з відомими для них класами). На цих даних тренується класифікатор. Останнім шагом є оцінка отриманого класифікатора на окремих реальних тестових даних. Якщо класифікатор має гарну точність по всім категоріям, то метрика різноманітності буде висока, оскільки генератору вдалось створити такі дані, що інша, натренована на цих даних модель, змогла чітко розрізняти реальні дані. Якщо згенеровані зразки будуть погано описувати реальні дані, то і класифікаційна модель не зможе з гарною точністю розрізняти тестовий набір. Схематично цей процес показано на рисунку 1.24.

Пайплайн GAN-test складається з похожих компонентів, але відрізняється процесами та зв’язками між ними. Спочатку створюється умовна ГМ та класифікатор на існуючих зразках тренувального набору даних. Далі генератором створюється набір синтетичних даних. Завдяки класифікатору ці синтетичні дані оцінюються – якщо мітка класифікатора збігається з високою ймовірністю з міткою, яку було передано моделі-генератору, то ці дані вважаються високоякісними, оскільки навчений на реальних даних класифікатор сприймає їх. Якщо ж модель генератор буде створювати дані поганої якості, то, ймовірно, класифікатор не зможе впізнати в них потрібній йому ознаки. Схематично цей процес показано на рисунку 1.25.

Ці два підходи зазвичай використовуються разом, тобто кроки створення ГМ та генерація даних робляться лише раз, а потім створюються вже класифікатори та проводиться оцінка відповідно до конкретної метрики.

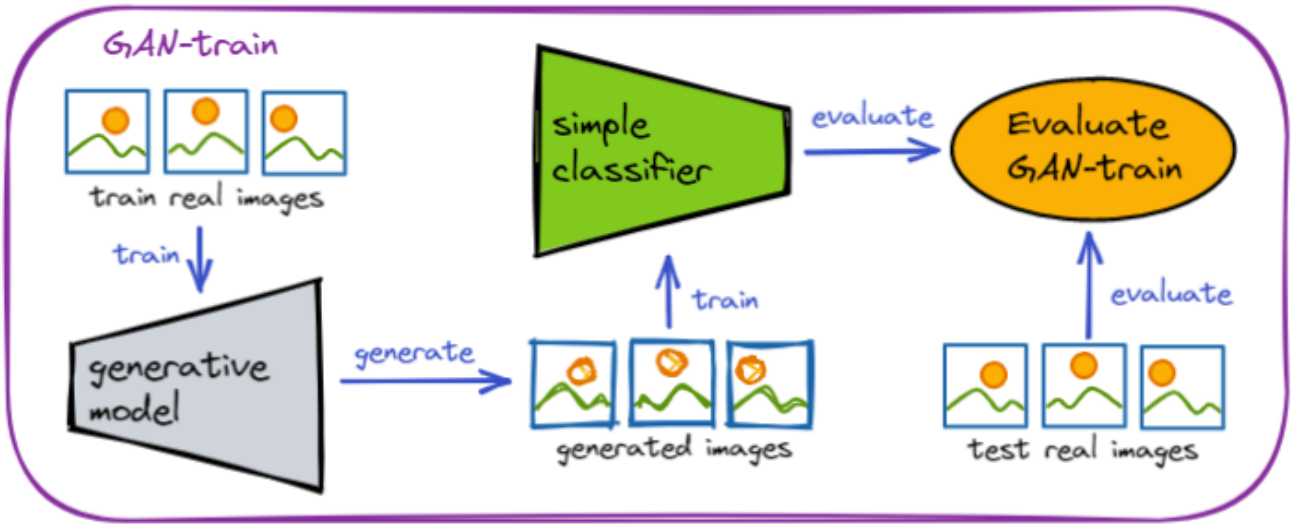


Рис. 1.24 Схема підходу GAN-train

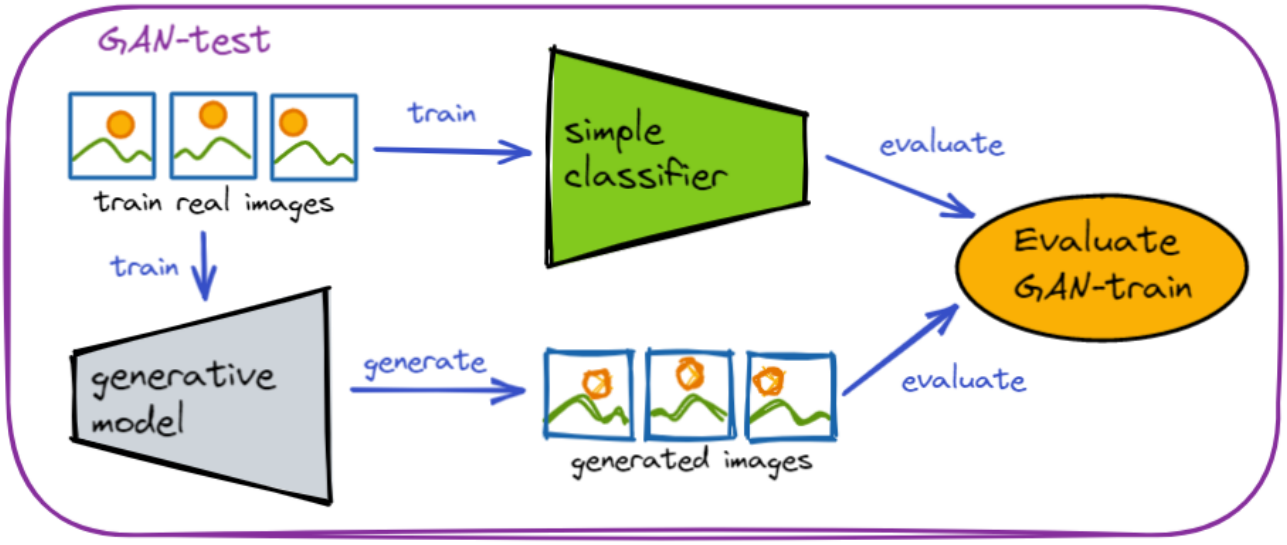


Рис. 1.25 Схема підходу GAN-test

Також дуже вагомою є проблема копіювання даних. Генеративна модель може просто запам’ятати частково або навіть повністю частини тренувальних даних та відтворювати їх. Непараметричний тест для виявлення копіювання даних [76] – це метод, що дозволяє визначати чи не перенавчилася генеративна модель на тренувальному наборі. Для цього методу потрібно: тренувальний, тестовий та згенерований набори даних. Якщо згенеровані дані знаходяться у просторі, наприклад, ознак моделі InceptionNet в середньому ближче до тренувального набору аніж тестові, то це може свідчити про проблему копіювання даних. Також варіація цього методу оцінювання передбачає ситуації, коли копіювання вийшло

тільки на частині тренувальних даних. Для цього простір ділиться на області, і на кожній з областей проводиться оцінювання ступеню копіювання.

Perceptual Path Length [57] дозволяє оцінити, наскільки складним є латентний простір після навчання генеративної моделі. Якщо зробити інтерполяцію у цьому просторі, то очікується, що у проміжних зображеннях буде схожий набір ознак, що наявний і у крайніх зображеннях, в ситуації, якщо простір є не викривленим. Але якщо простір є занадто заплутаним та складним, то у процесі інтерполяції можуть з'являтись дуже незвичайні ознаки на проміжних зображеннях. PPL дозволяє кількісно оцінити, наскільки змінюються ознаки даних у процесі інтерполювання у просторі.

Метрика лінійної роздільності [57], як і PPL, оцінює якість латентного простору. Якщо простір є не дуже складним в ньому може з'явитись вектори, що відповідають за присутність тієї чи іншої ознаки. Ця метрика дозволяє оцінити таку здібність простору, надаючи оцінку, наскільки добре точки прихованого простору можуть бути розділені на два кластери за допомогою лінійної гіперплощини, так щоб кожен кластер відповідав певному бінарному атрибуту даних.

1.3.2 Методи на основі людської оцінки

Методи, які передбачають залучення до оцінювання людини мають перевагу в тому, що людина краще може розуміти деякі нюанси предметної області та світу (наприклад, дуже легко впізнають неправильні риси обличчя, несумісності у згенерованому ландшафті тощо).

Часто для оцінювання моделей потрібно згенерувати деякий набір даних та віддати ці дані на оцінку групі експертів, які ретельно переглянуть дані та зроблять висновок про те, наскільки доброю є якість даних. Однак, цей процес повинен мати чіткі специфікації та критерії якості.

Особливості залучення людини до оцінювання згенерованих даних найкраще описані у методі-стандарті Human eYe Perceptual Evaluation [77]. В

основі цього стандарту є вимірювання реалістичності сприйняття результатів людиною, чітка послідовність дій при оцінюванні, забезпечення стабільності отримання результатів, а краудсорсингова платформа дозволяє ефективно використовувати грошові та часові ресурси..

Існує дві варіації стандарту HYPE: часовий ($HYPE_{time)}$ та необмежений у часі ($HYPE_{\infty}$).

Часова варіація полягає у тому, що вираховується час, який потрібний людині, щоб зрозуміти справжність чи несправжність зображення. Наприклад, оцінка в одну секунду показує, що людині потрібно не менше однієї секунди, щоб відрізнити реальне зображення від згенерованого.

Процедура оцінки базується на адаптивному методі сходинок [78]: зображення протягом деякого часу показується людини, потім людині потрібно відповісти на питання про справжність зображення, і якщо відповідь правильна, то наступного разу часу буде трохи менше, якщо неправильна, то часу буде більше. Зазвичай використовується часові надбавки у відношенні три до одного: за правильну відповідь віднімається у три разі більше часу, аніж додається при неправильній відповіді. Для кінцевої оцінки часу зазвичай обирається 150 зображень, з яких половина реальні, а половина згенеровані, а межі часової оцінки становлять від 100 мс до 1 с.

На основі $HYPE_{time}$ побудовано метод $HYPE_{\infty}$, який є простішим, швидшим та дешевшим. Цей стандарт відійшов від часових рамок та дозволяє людині вирішувати реальне чи згенероване зображення йому показано без прив’язки до часу. Зазвичай використовується 50 реальних та 50 згенерованих зображень. 10% $HYPE_{\infty}$ показує, що тільки 10% зображень є незрозумілими для експертів, коли 50% показує, що людина плутає половину зображень, іншими словами згенеровані зображення не можна відрізнити від справжніх (див. рис. 1.26). Більше 50% показує, що згенеровані зображення є гіпер-реалістичними – схожі на реальні більш ніж справжні. В такому випадку людина дає мітку справжніх зображень згенерованим.

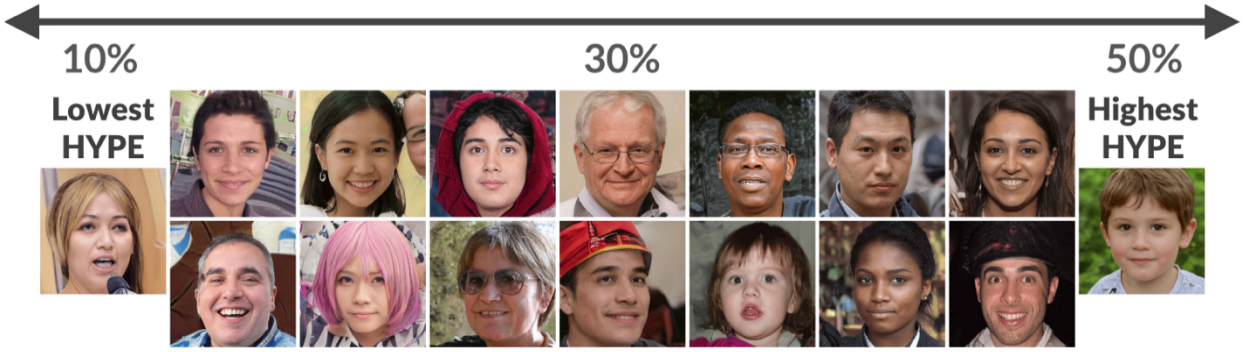


Рис. 1.26 Приклад згенерованих зображень та HYPE оцінки для них [77]

1.4 Аналіз структурних проблем на згенерованих зображеннях

ГЗМ може навчитися створювати досить гарні зображення, але при цьому може додати на них деякі артефакти. Одним із прикладів є артефакт шахової дошки [79], коли на зображенні виникає неприродний шаблон, що нагадує шахівницю. Інший відомий артефакт — краплеподібні артефакти (схожі на краплі води на поверхні зображення), які були присутні у першій версії моделі StyleGAN, але вирішені у StyleGAN2.

У цій роботі розглянуто ще один цікавий ефект, що можна назвати «артефактом межі» [7]. Цей специфічний шаблон виникає на межах згенерованого зображення і характеризується різкими переходами кольорів між сусідніми групами пікселів. На рисунку 1.27 показано приклад такого артефакту для умовної ГЗМ, яка генерує зображення кульбабок та чорнобривців: на лівому зображенні в нижній частині видно коричнево-зелену смугу, а на правому — смугу високої контрастності в правому верхньому куті, яка помітно відрізняється від кольорів нижче неї.

Рис. 1.27 Артефакт межі у ГЗМ

Згадана раніше модель StyleGAN2, яка демонструє високий рівень майстерності в генеруванні облич неіснуючих людей [80], також стикається з проблемою артефактів на межах зображень. Як показано на рисунку 1.28, на лівому зображенні два нижніх ряди пікселів мають світліший відтінок, що створює контрастну лінію вздовж нижньої межі. На правому зображенні в лівому верхньому куті також видно смуги пікселів, які суттєво відрізняються за кольором від сусідніх областей, що спричиняє порушення візуальної цілісності зображення.

Рис. 1.28 Артефакт межі у моделі StyleGAN2

1.5 Проблема створення даних для задач сегментації

Сегментація є однією з ключових задач у галузі CV, зокрема у сферах, що потребують точного аналізу зображень, таких як медична діагностика (див. рис. 1.29), промисловий контроль якості або наукові дослідження. Основна мета сегментації полягає у поділі зображення на значущі регіони, кожен з яких відповідає певному об'єкту, структурі або класу. Це дозволяє виділити важливі елементи, що можуть бути використані для подальшої обробки, аналізу або прийняття рішень.

Попри значний прогрес у цій галузі, процес сегментації часто супроводжується низкою викликів, пов'язаних з поганою якістю та маленьким обсягом вхідних даних. Наприклад, недостатня кількість високоякісних маркованих зображень може суттєво вплинути на точність моделей. Крім того, у випадку використання згенерованих або синтетичних даних для навчання моделей сегментації виникає ще одна важлива проблема – створення точних масок для цих даних. Маски повинні адекватно відображати об'єкти на зображенні, щоб модель змогла навчитися коректно розпізнавати й розділяти різні класи на реальних зображеннях.

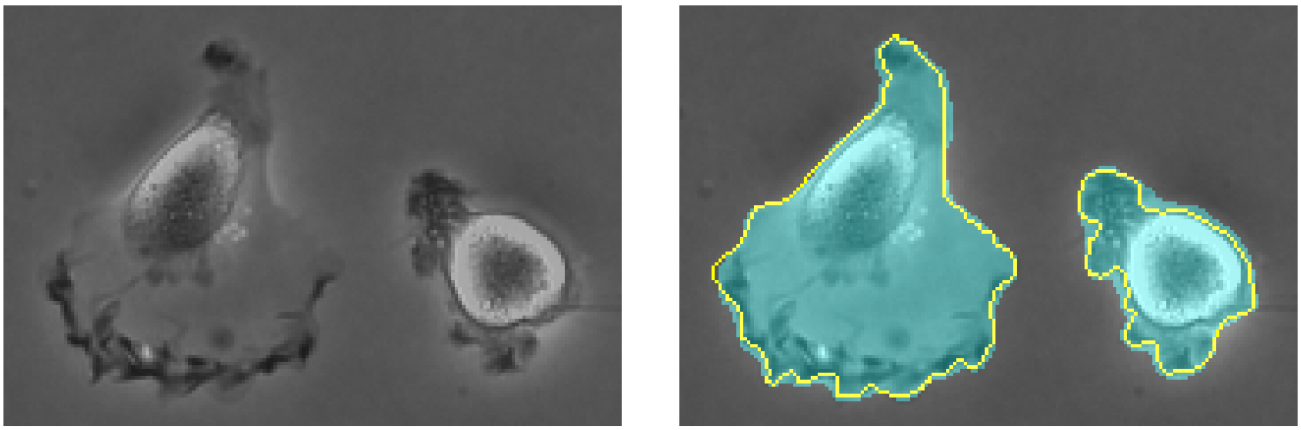


Рис. 1.29 Приклад роботи сегментації зображень [60]

1.5.1 Проблема створення масок

Маски є важливим компонентом у задачах сегментації, оскільки саме вони визначають, які частини зображення належать до певних об'єктів або класів. Без масок модель не може навчатися, адже вони виконують роль таргетних міток і використовуються для обчислення функції втрат, що безпосередньо впливає на оновлення параметрів моделі. Процес створення масок для згенерованих даних може бути трудомістким і ресурсозатратним, оскільки зазвичай вимагає ручної розмітки зображень.

Ручне створення масок – це не лише виснажливий, але й дорогий процес, який може суттєво збільшити витрати на підготовку даних. У зв'язку з цим критично важливим стає розвиток нових методів автоматизованого або напівавтоматичного створення масок [81]. Такі методи дозволяють значно скоротити витрати, прискорити процес підготовки даних і забезпечують можливість обробки великих обсягів зображень, що робить їх незамінними в умовах масштабного навчання моделей.

1.5.2 Проблема дисбалансу кількості даних

Часто у задачах сегментації зустрічається проблема незбалансованих даних [8]. Зокрема, дані можуть містити велику кількість зображень нульового класу (фон або відсутність об'єкта) та обмежену кількість зображень першого класу (цільові або основні об'єкти). Це створює додаткові перешкоди у навчанні моделей, оскільки більшість алгоритмів можуть зміщувати прогнози в бік класу, у якого кількість зразків більша [82].

Незбалансованість даних може призводити до того, що моделі не здатні ефективно виявляти цільові об'єкти. Однак, наявність даних нульового класу може бути використана для генерації нових прикладів цільового класу. Це б дозволило збалансувати набір даних та поліпшити умови для навчання моделі. Наприклад, методи синтетичного збільшення даних можуть бути використані для

перетворення зображень нульового класу у зображення першого класу, що допомагає створити більш збалансований та репрезентативний набір даних.

1.5.3 Аугментація фрагментів зображень

Аугментація даних [10] є одним із ключових етапів під час створення моделей машинного та глибокого навчання. У контексті обробки зображень, аугментація означає генерацію нових зображень на основі наявних шляхом застосування різних операцій трансформації. В залежності від характеру трансформацій і їхнього впливу на зображення, аугментацію можна поділити на кілька типів:

Геометрична аугментація включає в себе операції, такі як зміна масштабу, обертання, зсув тощо. Ці трансформації змінюють просторове розташування об'єктів, їхній розмір та орієнтацію.

Кольорова аугментація впливає на колірні характеристики пікселів зображення, включаючи зміну яскравості, контрастності, насиченості та відтінку, а також додавання шуму або застосування фільтрів кольорової корекції.

Текстурна аугментація змінює текстурні властивості зображення шляхом розмивання, додавання нових текстур за допомогою фільтрів або внесення артефактів та шуму, що може допомогти моделі в розпізнаванні об'єктів в умовах реального світу.

Аугментація застосовується не лише до основних зображень, але й до масок чи рамок об'єктів, коли вирішується задача сегментації або детектування.

Аугментації також можна поділити на ті, що змінюють цільовий клас об'єкта, і ті, що цього не роблять. Наприклад, при дзеркальному відображенні цифри «6» по горизонталі, а потім по вертикалі, утворюється цифра «9», що потребує зміни класу для цього зображення.

Окремою важливою проблемою в сегментації є розмір об'єктів, які часто є значно меншими порівняно з повним зображенням. Використання фрагментів зображення замість повного може бути ефективним способом оптимізації ресурсів

і підвищення якості аугментації. Цей підхід дозволяє здійснювати трансформації на окремих частинах зображення, а потім інтегрувати їх назад у повне зображення.

Такий метод збільшує кількість цільових об'єктів у навчальному наборі даних і підвищує точність моделі. Аугментація фрагментів зображень може включати як класичні трансформації, так і застосування генеративних моделей, що може більше покращити різноманітність і якість навчального набору.

1.6 Висновки до розділу та постановка задач дослідження

У першому розділі дослідження охарактеризовано основні задачі генерування даних: генерування реалістичних зображень, покращення роздільної здатності зображень, зміна атрибутів об’єктів, створення зображень за текстовим описом, перетворення одного зображення в інше, генерація музики, мови, звуків, перетворення тексту в аудіо формат, перетворення стилю аудіозаписів, застосування методів до відео, генерування та реферування текстів, фільтрація аномальних даних та кластеризація. Наведено приклади успішного застосування генеративних моделей, що показує актуальність і затребуваність таких методів на сьогоднішній день.

Здійснено огляд відомих моделей що використовуються для задач генерації даних:

- Автокодувальники: звичайні та варіаційні,
- Генеративні змагальні мережі: DCGAN, WGAN, WGAN-GP, BigGAN, ProGAN, StyleGAN, StyleGAN2.
- Дифузійні моделі: SD, SD2, SD3, SDXL.

Охарактеризовано переваги, недоліки, обмеження щодо перспектив використання цих моделей. Виокремлено генеративні змагальні мережі як тип моделей що мають достатньо високу якість і потребують помірних ресурсів для створення, тренування та використання у експериментах.

Наведено методи та метрики оцінювання згенерованих даних:

- Методи автоматичної оцінки, такі як Inception Score, Fréchet Inception Distance, влучність, точність, покриття, щільність, GAN-train та GAN-test, непараметричний тест для виявлення копіювання даних, Perceptual Path Length, метрика лінійної роздільності.

- Методи на основі людської оцінки такі як HYPE – часовий ($HYPE_{time}$) та необмежений у часі ($HYPE_{\infty}$).

Показано такі артефакти як ефект шахівниці та артефакт межі що можуть виникати при генеруванні зображень.

Охарактеризовано проблему генерування даних для задач сегментації, а саме виділено проблему створення масок сегментації на ряду з генерацією зображення, проблему незбалансованих наборів даних, коли йдеться про набори даних в яких багато фону та мало об’єктів цільового класу, роботу з маленькими частинами зображень, так як поширена ситуація, коли більшість об’єктів займають лише маленьку частину повного зображення.

Отже, для досягнення поставленої у дисертації мети щодо вдосконалення архітектури генеративних змагальних мереж, завдяки введення компонента самовалідування, та створення методу оцінки часткового згенерованих даних необхідно розв’язати наступний перелік задач:

1. Здійснити детальний аналіз сучасних підходів до генерування даних за допомогою моделей глибокого навчання. Особливу увагу приділити автокодувальникам, генеративним змагальним мережам та дифузійним моделям, розглянувши їх ключові принципи, переваги та недоліки.

2. Проаналізувати ключові проблеми, пов'язані з оцінюванням результатів генерування даних. Зокрема, визначити основні виклики, які виникають під час оцінки якості згенерованих даних, а також дослідити обмеження існуючих метрик оцінювання та їх вплив на точність результатів.

3. Створити нові підходи, методи та алгоритми, які сприятимуть покращенню якості згенерованих даних, оптимізації процесу генерування та зменшенню часу, необхідного для вибору найкращих моделей, забезпечуючи при цьому якість роботи мереж.

4. Удосконалити методи оцінки результатів генерування: розробити нові технології та інструменти для оцінювання якості частково згенерованих даних, які будуть боротись з недоліками існуючих методів і дозволять об'єктивно оцінювати результати.

5. Розробити структуру нової інформаційної технології самовалідованих генеративних змагальних мереж, визначивши її архітектуру та ключові компоненти з використанням UML-діаграм. Описати функціональні можливості кожного компонента та їх взаємодію, що мають забезпечувати ефективність і узгодженість роботи всієї системи.

6. Розробити комплекс програмних модулів на мові програмування Python для впровадження та використання інформаційної технології самовалідованих генеративних змагальних мереж. Цей набір модулів має забезпечити ефективне та універсальне вирішення практичних задач, дозволяючи легко адаптувати технологію до різних сценаріїв застосування.

7. Провести апробацію та практичне застосування розробленої інформаційної технології самовалідованих генеративних змагальних мереж на низці реальних задач. Оцінити ефективність запропонованих рішень та їхню доцільність для використання у конкретних прикладних сценаріях.

Результати розділу опубліковано в таких роботах автора: [1], [5], [6], [7], [8], [9].

РОЗДІЛ 2
РОЗРОБКА САМОВАЛІДОВАНИХ ГЕНЕРАТИВНИХ ЗМАГАЛЬНИХ МЕРЕЖ ДЛЯ КОНТРОЛЮ ГЕНЕРУВАННЯ ШТУЧНИХ ДАНИХ

2.1 Розробка базової версії ГЗМ Васерштейна

Спершу розглянемо задачу генерації зображень за допомогою генеративно-змагальних мереж для двох популярних квітів, поширених в Україні: кульбаби та чорнобривців [1]. Для виконання цього завдання було сформовано датасет [83], який складається з 900 знімків реальних квітів – по 450 зображень на кожен з класів. Кожне зображення являє собою кольоровий RGB файл з трьома каналами розміром 512x512 пікселів у форматі PNG (див. рис. 2.1).

В усіх наступних експериментах для цих зображень застосовувалася така процедура аугментації даних:

- перевертання по горизонталі;
- перевертання по вертикалі;
- транспонування;
- поворот на кут в діапазоні значень від –90 до 90 градусів.

Рис. 2.1 Приклади реальних зображень класів, обраних для генерації

Під час вирішення поставленого завдання було реалізовано та досліджено різні архітектурні підходи до проєктування генераторів і дискримінаторів у контексті генеративно-змагальних мереж.

В якості основи було обрано одну з найпопулярніших та найефективніших моделей – WGAN-GP.

У першому експерименті вирішено використовувати невеликий розмір зображень (28x28 пікселів), щоб оцінити, наскільки WGAN підходить для генерації цього типу зображень. Важливо зазначити, що класифікація на два типи квітів (чорнобривці та кульбаби) в цьому експерименті не проводилась, метою генератора було створення будь-якої квітки, яка б нагадувала реальну. Генератор було реалізовано як нейронну мережу, що складається з чотирьох блоків. Перші три блоки включають наступні компоненти:

Шар транспонованої згортки [84]. Цей шар формує необхідні ознаки та збільшує вихідний розмір, що показано на рисунку 2.2. Принцип дії шару транспонованої згортки: у звичайному згортковому шарі ядро (фільтр) ковзає по вхідному зображенню, обчислюючи скалярний добуток між ядром і областю вхідного зображення, що призводить до зменшення розмірів вихідного зображення. У транспонованій згортці цей процес інвертується. Замість того, щоб зменшувати розміри, шар транспонованої згортки збільшує їх. Це досягається шляхом розміщення ядра на кожному значенні вхідного тензора і додавання результатів. Якщо крок транспонованої згортки дорівнює одиниці, обчислення в деяких областях можуть перекриватися, що призводить до того, що значення додаються разом у цих місцях, збільшуючи значення пікселів у вихідному тензорі. Це перекриття обчислень дозволяє генерувати складніші та більш деталізовані карти ознак.

Розмір вихідного тензора залежить від розміру вхідного тензора, розміру ядра, кількості нульових інтервалів (які також називаються кроком) і параметрів відступу (padding). Формула для обчислення розміру виходу:

$$H_{out} = (H_{in} - 1) * stride - 2 * padding + kernal_size, \quad (2.1)$$

де H_{out} – висота вихідного тензора, H_{in} – висота вхідного тензора, $stride$ – крок згортки, $padding$ – кількість пікселів, доданих до кожного краю вхідного тензора перед застосуванням згортки, $kernal_size$ – розмір ядра згортки.

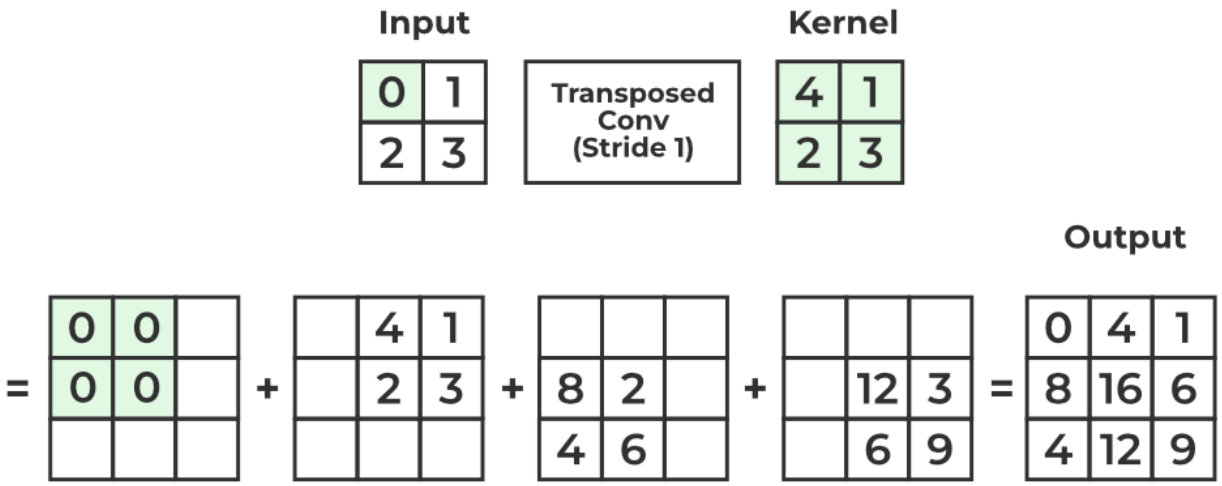


Рис. 2.2 Приклад роботи шару транспонованої згортки [85]

Пакетна нормалізація – це метод, який використовується для прискорення та стабілізації навчання нейронних мереж. Принцип дії пакетної нормалізації полягає в нормалізації шарів нейронної мережі по міні-пакетах даних. Ось основні принципи пакетної нормалізації:

Під час навчання модель працює з міні-пакетами даних (зазвичай це невеликі підмножини даних навчання), які проходять через нейронну мережу. Для кожного шару мережі, на який застосовується нормалізації, обчислюється середнє значення та дисперсія активацій цього шару в межах міні-пакета:

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i, \tag{2.2}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2, \tag{2.3}$$

де m – розмір міні-пакета, x_i – активації на виході нейронів.

Активації нормалізуються шляхом віднімання середнього значення та ділення на стандартне відхилення:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \quad (2.4)$$

де ϵ – це невелике значення, яке додається для запобігання діленню на нуль.

Після нормалізації додається додаткова параметризація, щоб модель могла навчитися оптимальним масштабам і зсувам активацій:

$$y_i = \gamma \hat{x}_i + \beta, \quad (2.5)$$

де γ і β – це параметри, які навчаються разом з іншими параметрами моделі. Вони дозволяють моделі адаптуватися до нормалізованих активацій, що дозволяє зберегти оригінальні характеристики розподілу, якщо це необхідно.

Параметри γ і β навчаються за допомогою зворотного поширення помилки, як і інші ваги моделі.

Зрізаний лінійний вузол (ReLU) [86] – це функція активації, яка широко використовується в нейронних мережах, особливо в глибокому навчанні. Принцип дії ReLU досить простий: вона залишає позитивні значення без змін, а всі негативні значення прирівнює до нуля (див. рис 2.3). Це робить її лінійною для всіх позитивних значень і нелінійною для негативних значень.

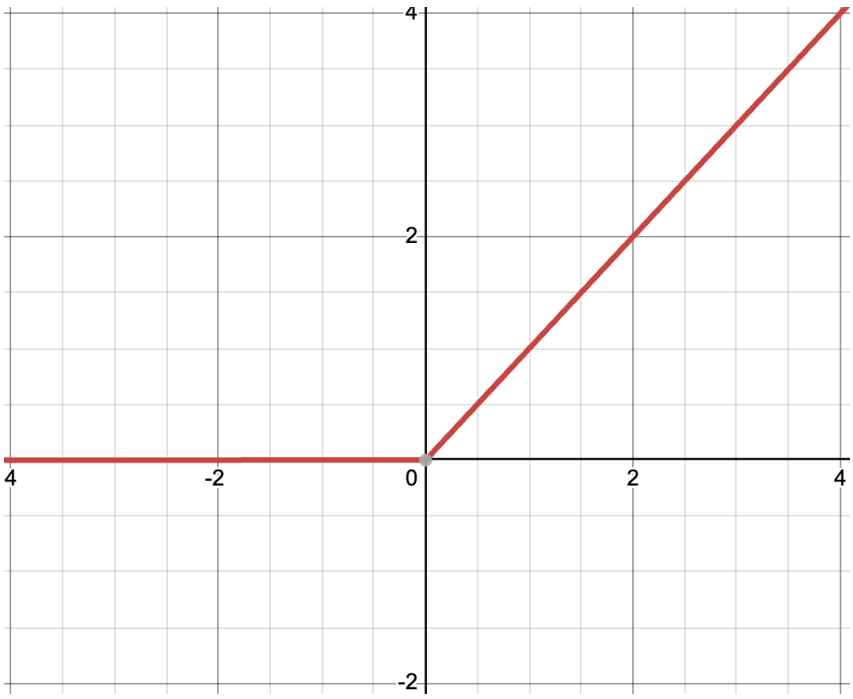


Рис. 2.3 Принцип роботи ReLU

Функція ReLU визначається так:

$$f(x) = \max(0, x), \tag{2.6}$$

де x – вхідне значення (підсумовані зважені входи нейрона).

Хоча ReLU є частково лінійною (на відміну від сигмоїдної або гіперболічної тангенс функцій), вона все ще забезпечує нелінійність, що необхідно для моделювання складних функцій у глибоких нейронних мережах. ReLU проста у реалізації та має низьку обчислювальну вартість, оскільки тільки порівнює значення з нулем і повертає результат. Це робить її більш ефективною для великих нейронних мереж порівняно з іншими активаційними функціями.

Крайній блок містить:

- шар транспонованої згортки, для формування корисних ознак;
- функція активації – гіперболічний тангенс [87]. Функція гіперболічного тангенсу є нелінійною і трансформує вхідні дані в діапазон від -1 до 1, що робить її корисною для задач, де необхідно нормалізувати дані або моделювати складні залежності. Функція гіперболічного тангенсу визначається як:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{2.7}$$

де x – вхідне значення (підсумовані зважені входи нейрона).

Графік tanh має S-подібну форму, схожу на сигмоїду, але з діапазоном від -1 до 1 і симетрією щодо початку координат, що показано на рисунку 2.4.

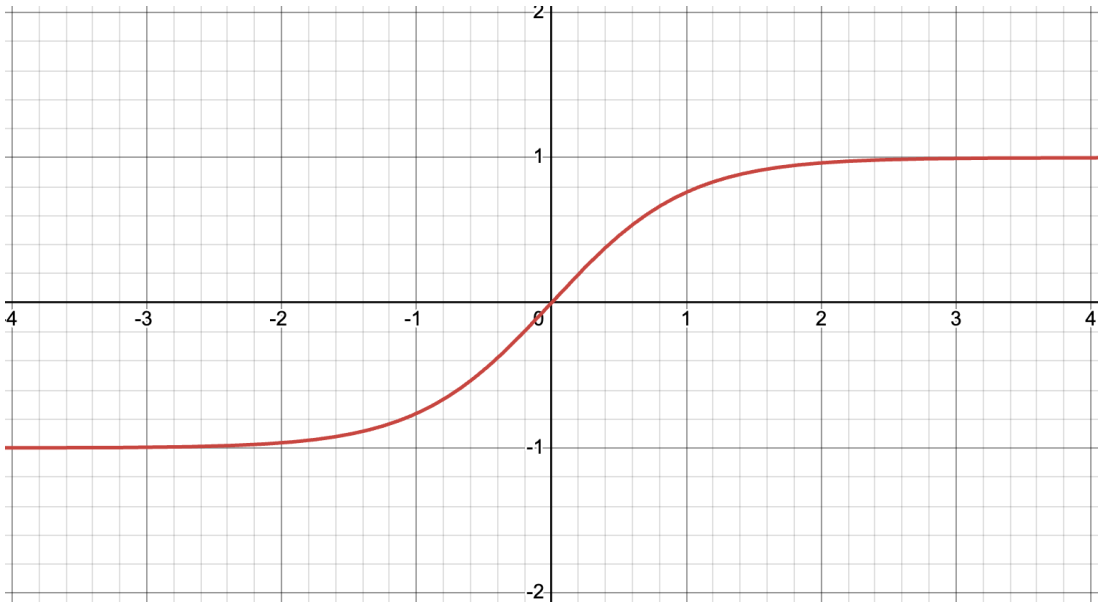


Рис. 2.4 Принцип роботи гіперболічного тангенсу

Критик складається з наступних компонентів:

- шар згортки, для формування корисних ознак та зменшення розміру;

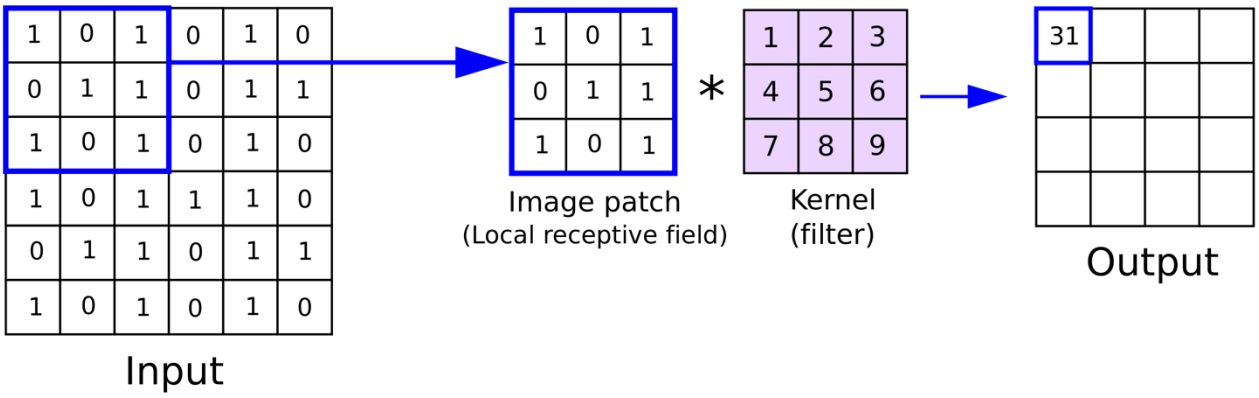


Рис. 2.5 Приклад роботи шару згортки [88]

- пакетна нормалізація;
- функція активації – нещільний зрізаний лінійний вузол (англ.: Leaky ReLU [89]); Ця варіація активаційної функції ReLU, яка вирішує проблему так званих "вмираючих" нейронів, характерну для стандартного ReLU. У стандартному ReLU всі негативні значення переводяться в нуль, що може

призвести до того, що деякі нейрони ніколи не будуть активними, і, як наслідок, не будуть навчатися. Leaky ReLU вирішує цю проблему, вводячи невеликий негативний нахил для від'ємних значень (див. рис. 2.6), що дозволяє деякій частині градієнта все ж проходити. Функція Leaky ReLU визначається так:

$$f(x) = \begin{cases} x, & if\ x > 0 \\ \alpha x, & if\ x \leq 0' \end{cases} \tag{2.8}$$

де α – це невеликий додатний коефіцієнт нахилу для негативних значень x. Зазвичай α обирається на рівні 0.01 або подібного невеликого значення.

Як і стандартний ReLU, Leaky ReLU є нелінійною функцією, що дозволяє моделі вчитися складним залежностям у даних. Але на відміну від ReLU, ця функція не "відсікає" всі негативні значення, а дозволяє їм частково проходити через функцію. Значення α можна налаштовувати або вибирати експериментально.

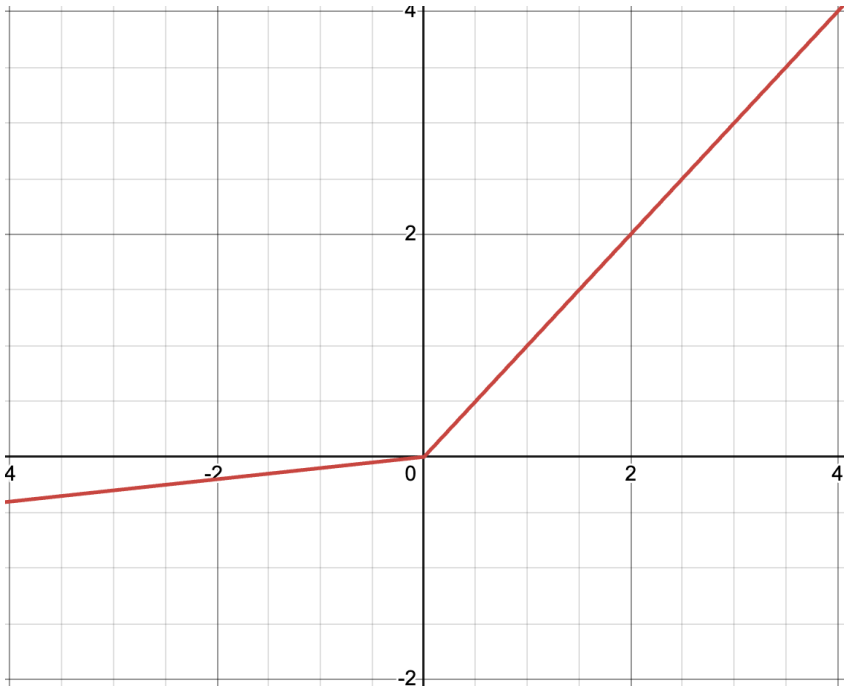


Рис. 2.6 Принцип роботи Leaky ReLU

Вихідний згортковий шар дискримінатору повертає лише одне числове значення (одноканальне зображення розміром 1x1), яке слугує оцінкою критика –

чим більше це значення, тим більша його впевненість у тому, що зображення є справжнім.

На рисунку 2.7 представлено результати генерації після початкового етапу, де можна побачити, що модель створює цікаві приклади, проте вони мають дуже маленьку роздільну здатність. Також добре помітно, що згенеровані зображення не демонструють чіткого поділу об’єктів на вхідні класи. Це ілюструється на прикладі з першого ряду та четвертої колонки рисунку 2.7 – одна половина квітки має червоне забарвлення, а інша – жовте.

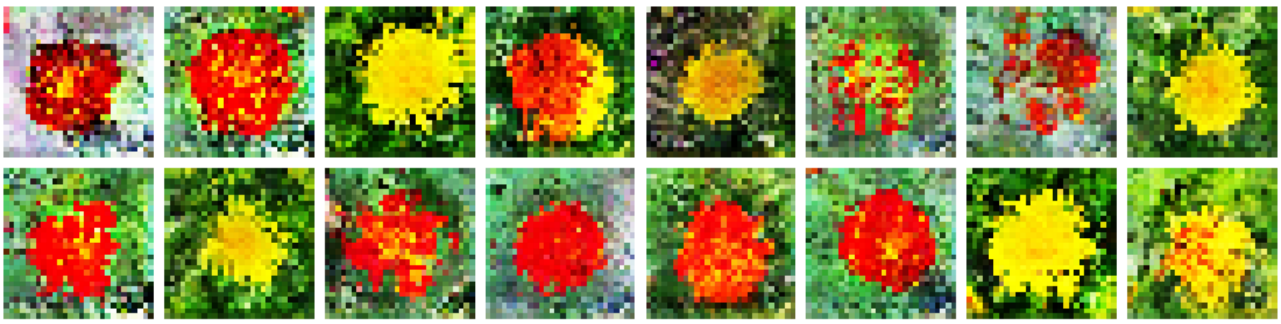


Рис. 2.7 Результат роботи WGAN-GP_28 у першому етапі

На другому етапі здійснюється перехід до розділення результатів генерації за класами. Для цього використовується умовна генерація, коли генератор, окрім шумового вектора, отримує також вектор унітарного коду, розмір якого відповідає кількості класів. Усі елементи цього вектора є нульовими, а клас, який потрібно згенерувати, позначається одиницею. У нашому випадку генерації квітів цей вектор містить два елементи (для двох видів квітів), тому після конкатенації з шумовим вектором розміром 64, вхідний вектор матиме 66 елементів. Критик, зі свого боку, отримує додаткові два канали унітарного коду, щоб визначати, якому класу відповідає зображення. Таким чином, на вхід критику надходитиме матриця розміром 28x28x5 замість попередніх трьох каналів (RGB).

Результати другого етапу генерації (умовна ГЗМ) представлені на рисунку 2.8, де чітко видно, що, окрім вирішення задачі поділу на класи і можливості генерувати зображення певного класу, покращилася й загальна якість зображень.

Рис. 2.8 Результат роботи CWGAN-GP_28 у другому етапі

На наступному етапі збільшимо роздільну здатність зображень до 64x64 пікселів. Для цього використовуються блоки, що й раніше, зокрема чотири основні блоки та один кінцевий для генератора, але з однією важливою зміною – у шарах транспонованої згортки фільтри будуть розміром три замість чотирьох, а крок залишиться незмінним і буде дорівнювати двом. Це дозволить отримати вихідні зображення розміром 64x64 пікселів. Для критика також використаємо чотири основні блоки та кінцевий шар для перетворення вихідного результату в фінальне значення.

Проте, через проблему того, що розмір фільтра не ділиться рівно на крок, виникає проблема – деякі пікселі отримують більше уваги, що призводить до появи так званого «ефекту шахової дошки». Це можна помітити на результатах генерації, показаних на рисунку 2.9: майже на кожному зображенні в лівому нижньому куті спостерігається патерн, що нагадує чорно-білі клітинки шахової дошки.

Рис. 2.9 Результат роботи CWGAN-GP_64 з ефектом шахової дошки

Щоб подолати ефект шахової дошки, у наступному експерименті було вирішено повернути розмір фільтрів до чотирьох, а досягти вихідного розміру зображення (64x64) за рахунок іншого гіперпараметра – доповнення у згортці. Однак, це не дозволило повністю позбутися цього ефекту, хоча він став менш вираженим. На рисунку 2.10 можна побачити, що текстура зображення залишається злегка нерівною, нагадуючи повторювану сітку, подібну до шахової дошки, хоча цей ефект і став менш помітним.

Рис. 2.10 Результат роботи CWGAN-GP_64 на четвертому етапі експерименту з іншим проявом ефекту шахової дошки

Для остаточного усунення цього ефекту (п'ятий етап експерименту) було змінено архітектуру блоків генератора. Було прийнято рішення відмовитися від транспонованих згорток, які відповідали як за збільшення розміру зображення, так і за його коректність. Тепер основні блоки мають наступну структуру:

- шар звичайної згортки, що покращує якості зображень та виявляє важливі ознаки;
- пакетна нормалізація;
- функція активації – зрізаний лінійний вузол.

Кінцевий компонент генератора теж зазнав змін:

- шар звичайної згортки;
- функція активації – гіперболічний тангенс.

Після кожного основного блоку застосовувався алгоритм білінійного підвищення дискретизації [90] для збільшення роздільної здатності зображення у два рази, що раніше виконувала транспонована згортка. На відміну від транспонованої згортки, білінійне підвищення дискретизації не є шаром що навчається, а базується на інтерполяції. Це означає, що значення пікселів у вихідному зображенні обчислюються на основі вагових середніх значень сусідніх пікселів із вхідного зображення, що показано на рисунку 2.11. Для обчислення значення пікселя $P(x', y')$ у збільшеному зображенні використовується наступна формула:

$$P(x', y') = (1 - \Delta x) * (1 - \Delta y) * P(x_1, y_1) + \Delta x * (1 - \Delta y) * P(x_2, y_2) + (1 - \Delta x) * \Delta y * P(x_3, y_3) + \Delta x * \Delta y * P(x_4, y_4), \quad (2.9)$$

де x' і y' – координати пікселя у збільшеному зображенні, x_i і y_i – координати чотирьох сусідніх пікселів у вихідному зображенні, Δx і Δy – відстані від координат x' і y' до найближчих сусідніх пікселів у вихідному зображенні по осі X і Y відповідно.

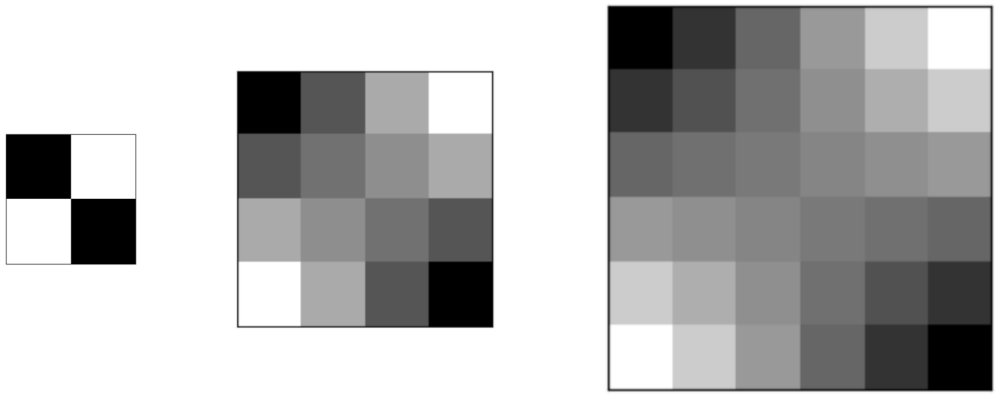


Рис. 2.11 Приклад роботи білінійного підвищення дискретизації

Фінальна архітектура генератора показана на рисунку 2.12.

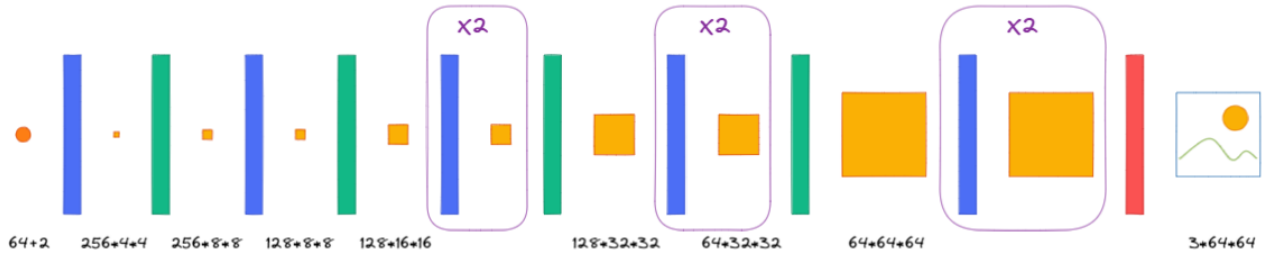


Рис. 2.12 Фінальна архітектура CWGAN-GP_64, де сині прямокутники відповідають за формування ознак, зелені блоки – за підвищення дискретизації, а червоний блок – за формування вихідного зображення

Використовуючі описаний підхід вдалося значно покращити якість згенерованих зображень (див. рис. 2.13). Для репрезентативності аналізу на рисунку 2.14 представлено збільшені зображення, створені моделлю. Чітко видно, що вони мають більш рівномірну текстуру і не містять артефактів, характерних для результатів моделей, де застосовувалися транспоновані згортки.

Рис. 2.13 Результат роботи моделі CWGAN-GP_64 з оновленою архітектурою

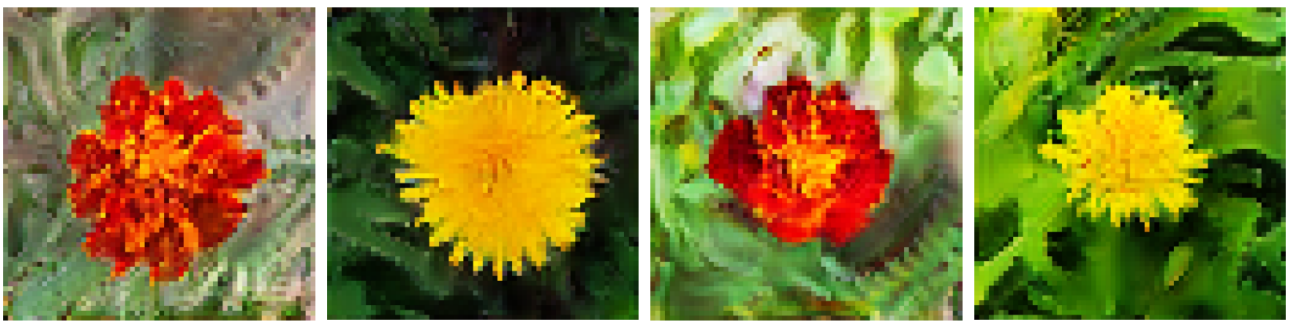


Рис. 2.14 Приклад відсутності ефекту шахової дошки у згенерованих зображеннях

На заключному, шостому етапі було прийнято рішення ще більше збільшити розмір вихідних зображень (вдвічі). Для цього до описаної раніше структури було додано ще один шар підвищення дискретизації та два блоки зі згорткою (див. рис. 2.15). Це дозволило отримати більш деталізовані зображення розміром 128x128 пікселів (див. рис. 2.16).

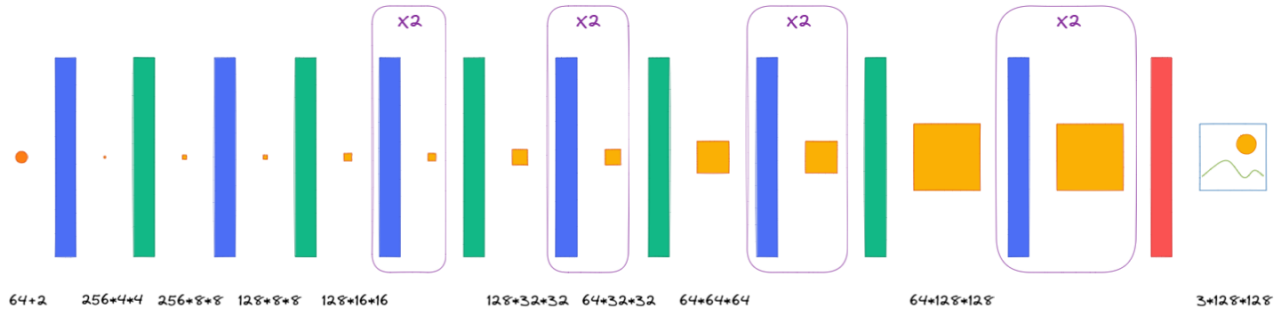


Рис. 2.15 Архітектура моделі CWGAN-GP_128 на останньому етапі експерименту

Рис. 2.16 Результат генерування моделі CWGAN-GP_128 на останньому етапі

На рисунку 2.17 наведено збільшені зображення, що отримані моделлю CWGAN-GP_128, з яких чудово видно, що вони мають гладку текстуру, а небажаних патерни відсутні.

Рис. 2.17 Збільшені приклади зображень розміром 128x128 пікселів

На рисунку 2.18 показано процес зміни вихідних зображень генератора (зліва – початкове, праворуч – кінцеве) з інтервалом у 500 епох для кожного з класів при ідентичному вектору шуму та унітарному коду, протягом навчання фінальної моделі. Можна побачити, як генератор поступово покращує якість та деталізацію зображень, зберігаючи основні особливості, такі як розташування квітки, розмір, напрямок росту квітів, а також фон.

Рис. 2.18 Зміна результатів генерування протягом навчання з кроком 500 епох

На рисунку 2.19 наведено приклади використання методу трюку відсікання [91] для фінальної моделі генерації, що дозволяє досягти компромісу між точністю та різноманітністю згенерованих зображень.

У першому рядку використано сильне відсікання, що забезпечує високу точність: усі квітки розташовані в центрі, мають чіткі межі та реалістичні деталі, наприклад, характерні пелюстки. Однак, різноманітність відсутня – всі зображення майже ідентичні.

У третьому рядку відсікання шуму не застосовується, тому деякі зображення мають нечіткі межі та деталі, не завжди розташовані по центру, але вони демонструють більше різноманіття, зокрема, у третьому прикладі видно незвичайний рожевий фон.

Другий рядок ілюструє проміжний варіант відсікання, який забезпечує баланс між точністю та різноманітністю.

Рис. 2.19 Використання методу трюку відсікання шуму: перший рядок – відсікання 0.01; другий рядок – відсікання 1.5; третій рядок – без

2.2 Використання розробленої ГЗМ Васерштейна для збільшення набору даних для оптичного розпізнавання символів

Оптичне розпізнавання символів [92] – це технологія, що дозволяє зчитувати та перетворювати текст із зображень, сканованих документів, фотографій або будь-яких інших візуальних форматів у цифровий текстовий формат. Основна мета OCR полягає в автоматичному розпізнаванні символів на зображенні та перетворенні їх у текст, який можна редагувати, шукати або аналізувати. Задачі OCR включають:

1. Попередня обробка зображення: Покращення якості зображення для точнішого розпізнавання, що може включати корекцію яскравості, контрасту, вирівнювання, зменшення шумів та генерування нових зразків даних.

2. Сегментація зображення: Виділення окремих символів, слів або рядків з усього зображення.

3. Розпізнавання символів: Ідентифікація кожного символу та його відповідність певному символу в алфавіті або набору символів.

4. Після-обробка: Коригування та покращення розпізнаного тексту, наприклад, перевірка орфографії, контекстне визначення слів і усунення помилок розпізнавання.

У випадках, коли даних недостатньо для тренування, оцінки, або перевірки на новому домені, GAN можуть бути використані для генерації додаткових синтетичних зображень, що відповідають характеристикам наявних даних. Основна ідея полягає в тому, щоб навчити генеративну модель створювати чи перетворювати нові зразки зображень, які є реалістичними та можуть підвищити об’єм набору для OCR.

В дослідженні розглянута проблема ідентифікації інформації на ID картках китайських паспортів. Основною проблемою є обмеженість набору даних, який містить 10 зображень різної якості. Задачею є вибір найкращого OCR методу серед існуючих [2].

Для цього розглянуто найпопулярніші методи для ORC та оцінена їх якість на цих десяти зображеннях. Окрім цього додатково створено ГЗМ що здатна генерувати цифри номеру ID картки для збільшення варіативності даних.

Для вирішення цього із існуючих даних було створено набір даних цифр паспортів, що вміщав в собі 52 зображення від 10 класів. Розподіл по класам дуже нерівномірний та показано на рисунку 2.20. Так для цифри «3» в наборі даних є лише 1 екземпляр, а для цифри «1» 12 штук.

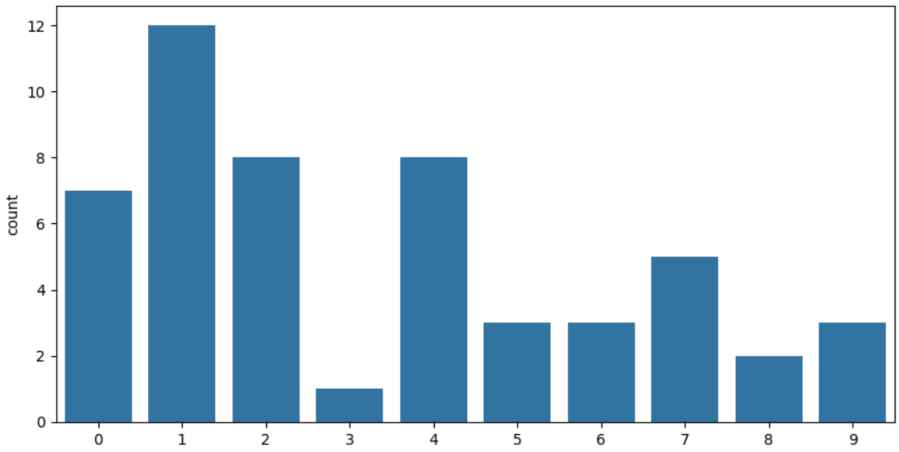


Рис. 2.20 Розподіл цифр для тренування

Моделі генератору та дискримінатору взяти з минулого розділу, а саме з експерименту CWGAN-GP_64, де вихідна розмірність дорівнює 64 на 64 пікселі, що є цілком достатнім для їх вставки в основне зображення. Окрім цього тренування характеризувалось наступними гіперпараметрами:

- кількість епох: 2000;
- розмір пакету: 52;
- кількість повторів критика: 1;
- коефіцієнт лямбда для GP: 10;
- однакові оптимізатори для генератора та критика: Adam, з швидкістю навчання 0.0002.

Приклади згенерованих зображень після останньої епохи тренування показані на рисунку 2.21.

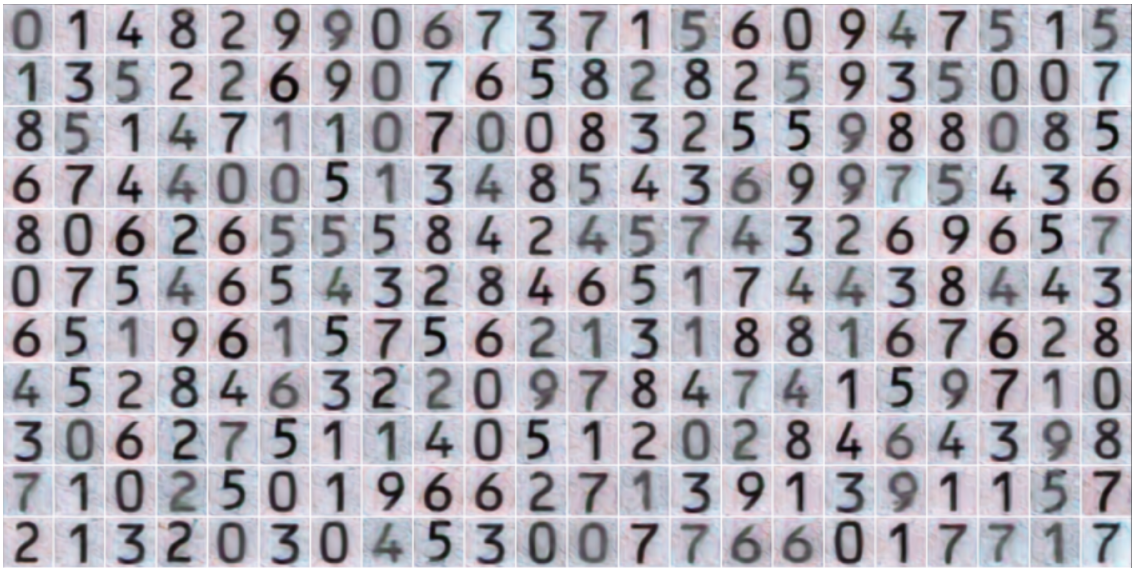


Рис. 2.21 Приклади згенерованих зображень цифр

Ці результати використані для збагачення набору даних шляхом вставки їх замість існуючих цифр в номерах ID карт. На рисунку 2.22 наведено анонімізована ID картка, де останні 7 цифр залишені відкритими. Зліва на рисунку показана не трансформована версія паспорту, з правої частини рисунку показана трансформація трьох цифр:

- Цифра на позиції 1 змінила своє значення з «5» на «3»;
- Цифра на позиції 3 змінила своє значення з «2» на «7»;
- Цифра на позиції 6 змінила своє значення з «2» на «1»;

Рис. 2.22 Приклад трансформації цифр для задачі ідентифікації тексту

При детальному аналізі цього зображення артефакти «вставки» є досить помітними, але, це не заважає використовувати ці дані в моделі. На рисунку 2.23

показано результати роботи моделі Paddle OCR [93] для цього зображення. Всі 7 цифр, включаючи 3 трансформовані були правильно оброблені цією моделлю.

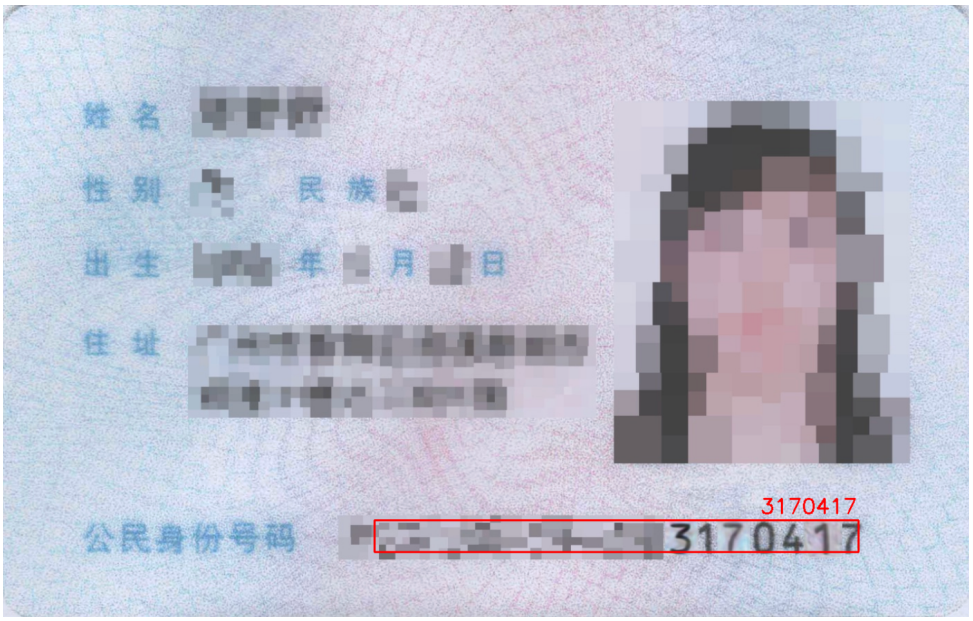


Рис. 2.23 Приклад роботи екстрагування тексту

2.3 Удосконалення архітектури ГЗМ, для автоматичного створення сегментаційної маски, завдяки використанню U-генератору

У цьому розділі детально описано метод перетворення зображень одного класу (наприклад, здорові зуби, легені або клітини) на зображення іншого класу з додатковим створенням маски, яка чітко визначає об'єкт (наприклад, пломбу на зубах, запалення в легенях або аномалії в клітинах). Такий підхід є корисним у випадках, коли необхідно створити модель за умов суттєвого дисбалансу класів, що є досить поширеною проблемою.

Архітектуру розробленого методу [3] показано на рисунку 2.24. Датасет поділяється на дві вибірки за класами: дані фонового класу та дані цільового класу, які містять відповідну сегментаційну маску. До зображень класу 0 додається випадковий шум, розмір якого відповідає розмірам вхідного зображення. Створюється вектор випадкових значень, який за шириною відповідає зображенню, і дублюється по висоті, утворюючи квадратну матрицю з градаціями

сірого. Цей шум додається як додатковий канал до вхідного зображення, надаючи моделі можливість створювати при кожній генерації унікальні характеристики (наприклад, різні розташування чи форму пломби), що зумовлено введенням випадкових даних на етапі підготовки. Для зображень у відтінках сірого цей канал є другим, для кольорових RGB-зображень – четвертим.

Генератор має класичну архітектуру U-Net (надалі – U-генератор) [94] і приймає на вхід зображення з каналом шуму. На виході йому потрібно відтворити вхідне зображення, додавши до нього ознаки іншого класу, та згенерувати маску як окремий канал для позначення місця цього нового об’єкта на зображенні.

Критик має на меті визначити, чи є комбінація зображення і маски згенерованою або реальною. Він також отримує на вхід зображення та маску, яка додається як додатковий канал.

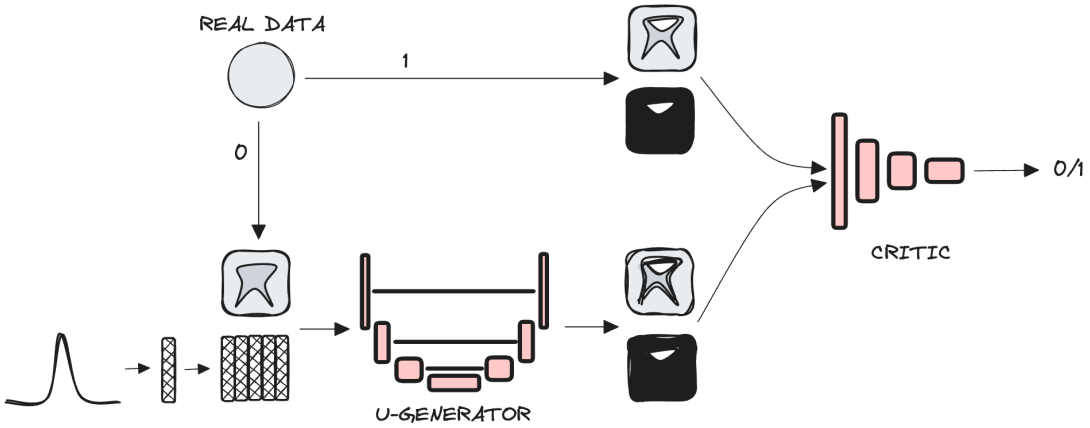


Рис. 2.24 Схема тренування моделі

З такими особливостями тренування генератор здатний трансформувати зображення одного класу у зображення іншого, одночасно створюючи сегментаційну маску цільового об'єкта, що додає контрольовану аугментацію даних під час навчання.

На рисунку 2.25 показано етапи перетворення зображень. Спочатку, подібно до процесу навчання, вибирається зображення без сегментаційного класу, до якого додається шум. Ця комбінована структура подається в U-генератор, який намагається додати ознаки нового класу та згенерувати сегментаційну маску

цільового об’єкта. Однак, створене зображення може бути недостатньо якісним або відрізнятися від оригінального (через наявність шумів). Тому наступним кроком є операція екстракції згенерованих ознак із отриманого зображення (з використанням маски) та їх додавання на вихідне зображення (див. рис. 2.26). Незважаючи на можливі недоліки, цей підхід дозволяє точно позиціонувати маску та надавати їй відповідний вигляд.

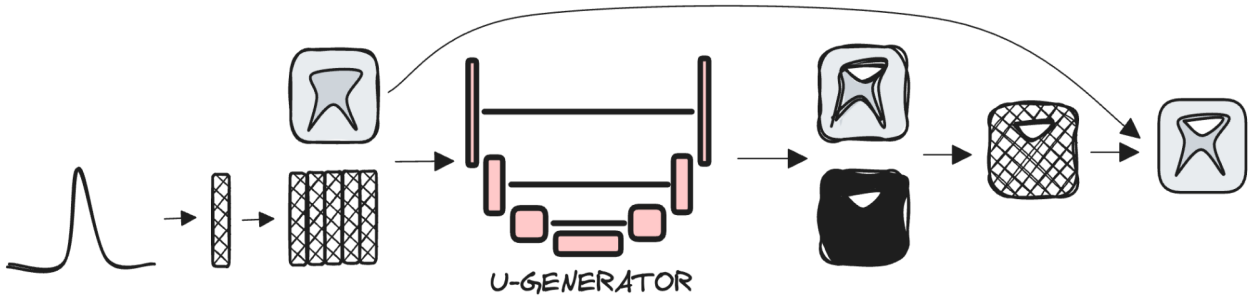


Рис. 2.25 Схема використання моделі

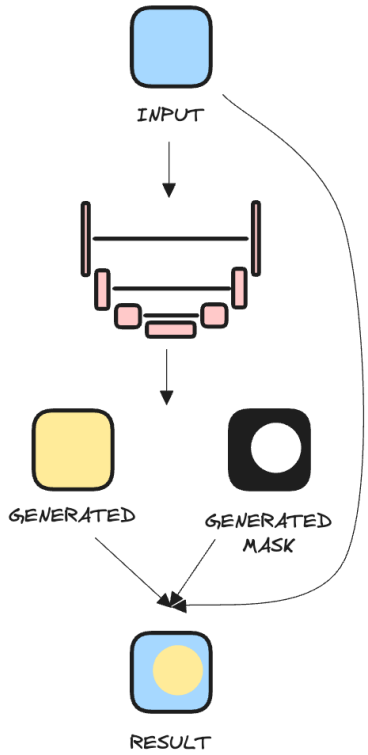


Рис. 2.26 Схема агрегації вхідного, згенерованного зображення та маски

Така післяобробка забезпечує кращу якість результатів та більшу узгодженість у збереженні основного зображення. Це особливо корисно в ситуаціях, як показано на рисунку 2.27: коли необхідно виконати аугментацію окремих частин зображення, наприклад, окремих зубів на панорамному знімку. Можна використовувати звичайний детектор об’єктів для вирізання необхідних зубів, передавати їх у модель для аугментації, а потім вставляти назад у початкову позицію на зображенні та відповідну сегментаційну маску. Нове зображення повинно залишатися максимально незмінним (окрім змін у доданому об'єкті), щоб правильно інтегрувати його в навколишній контекст.

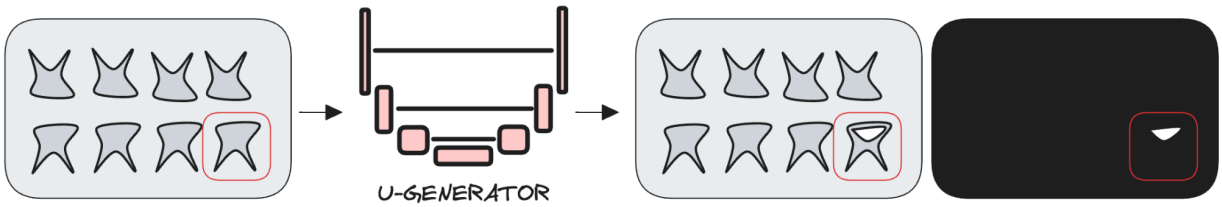


Рис. 2.27 Приклад часткової генерації

Було проведено експеримент із тренування генеративної моделі для перевірки ефективності цього підходу. Із набору даних панорамних знімків щелепи [95] було вибрано 128 зображень окремих зубів без пломб. Цим зображенням було присвоєно нульовий клас та порожню сегментаційну маску, відповідні приклади наведено на рисунку 2.28.

З цього ж набору даних було обрано 148 зображень зубів із пломбами, яким було присвоєно клас номер 1 (цільовий). Пломби на цих зубах були розмічені бінарною маскою такого ж розміру, де нуль (чорний колір) означає відсутність пломби, а одиниця (білий колір) позначає її наявність, як показано на рисунку 2.29.

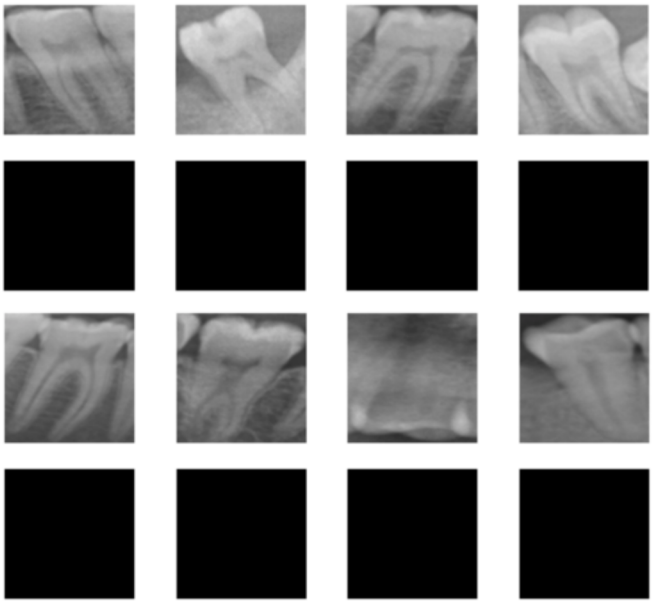


Рис. 2.28 Дані нульового класу

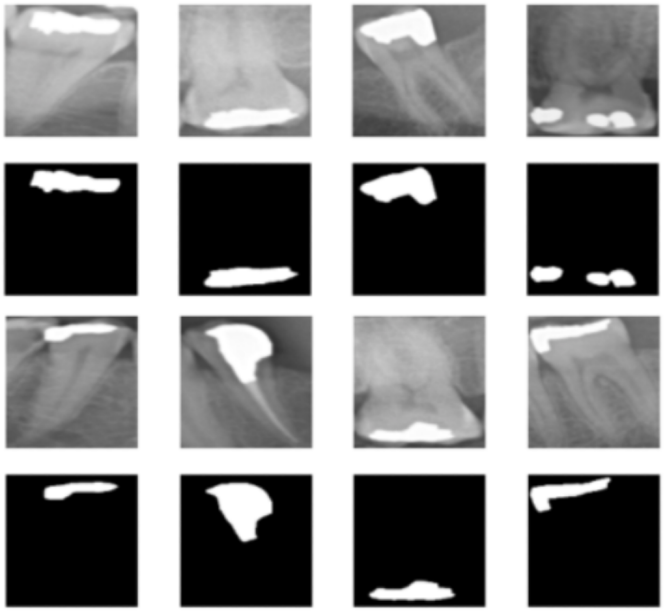


Рис. 2.29 Дані першого класу

З набору першого класу було відібрано 20 зображень зубів, для формування тестового набору для фінальної оцінки моделей (див. рис. 2.30). Варто зазначити що панорамні знімки КТ, з яких взяті ці зображення, використовувалися виключно для тестування. Ці зображення не використовувалися під час тренування, налаштування, валідації чи вибору моделей. В основному наборі залишилося по 128 зображень для нульового та першого класів.

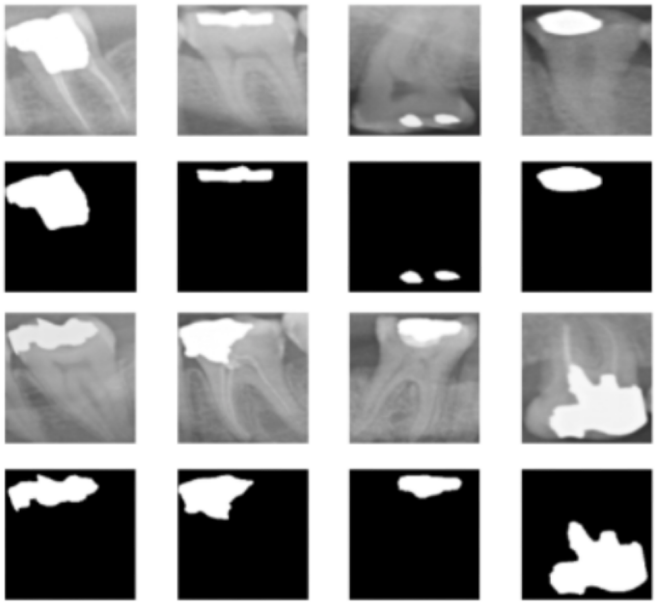


Рис. 2.30 Тестувальні дані

Зазначені вище зображення разом із відповідними сегментаційними масками використовувались для тренування моделі були. Перед поданням у модель зображення було перетворено до розміру 128x128 пікселів. Генератором слугувала архітектура U-Генератору, а дискримінатором – звичайна згорткова нейронна мережа. В якості функції втрат було обрано відстань Васерштейна.

Для тренування використовувались наступні гіперпараметри:

- кількість епох: 300;
- розмір пакету: 16;
- оптимізатор (для обох моделей): Adam (beta0: 0, beta1: 0.999);
- швидкість навчання (learning rate): 0.0002.

Кожні 10 епох проводився огляд результатів генерації. Найкраща епоха, визначалась на основі візуальної оцінки згенерованих зображень, і мала номер 210.

На рисунку 2.31 наведено приклад даних, створених натренованою моделлю, та поетапний (зверху вниз) процес переходу від вхідного зображення нульового класу до зображення першого класу. У першому рядку представлено 8 вхідних зображень, які подаються до генератора. Генератор трансформує їх у

зображення, показані у другому рядку, з відповідними масками в третьому рядку. Як можна побачити, створені зображення є дещо розмитими та нечіткими. Для покращення результату із зображень вирізаються лише ті фрагменти, які відповідають масці, і ці фрагменти накладаються на вхідні зображення, формуючи фінальний результат, представлений у четвертому рядку.

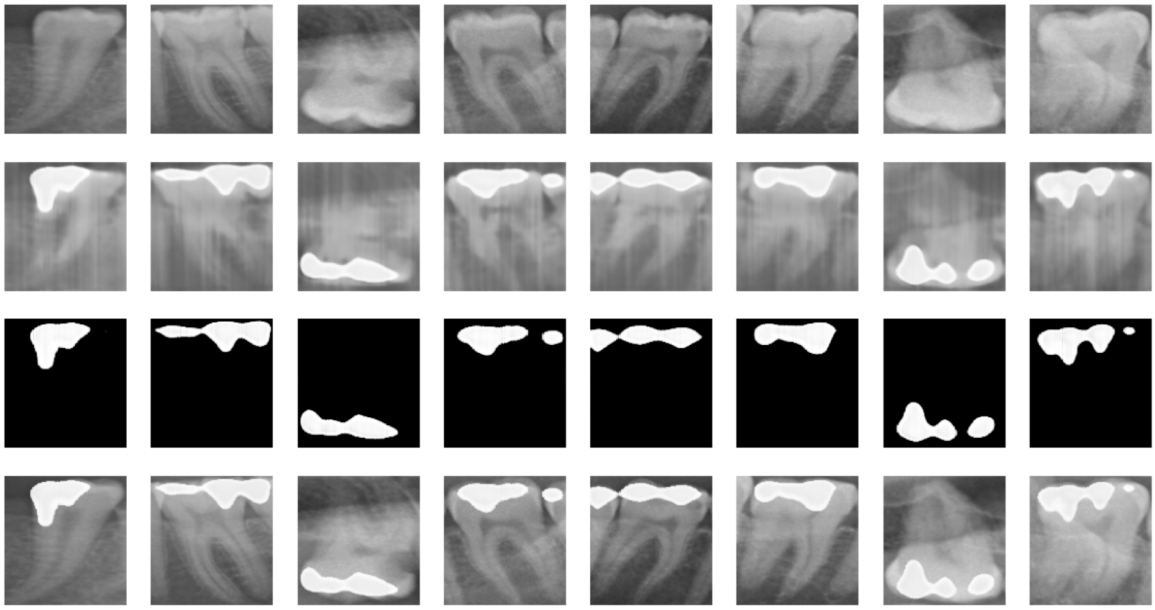


Рис. 2.31 Приклад створення зображень

Для оцінки впливу вказаної аугментації на ефективність тренування моделі було проведено експеримент із використанням класичної архітектури U-Net для сегментації пломб. Було створено два окремі набори даних для тренування моделей:

1. Тренування на основі початкового набору: відібрано 128 зображень першого класу, які було поділено на тренувальний та валідаційний набори у співвідношенні 80% до 20% (102 зображення для тренування і 26 для валідації).

2. Тренування з розширеним набором: до початкового тренувального набору було додано 64 згенеровані зображення, створені U-генератором на 210-й епосі. Кращі зразки серед згенерованих зображень було відібрано вручну (приклади цих зображень представлені на рисунку 2.32).

Такий підхід дав можливість оцінити вплив аугментації на продуктивність моделі та порівняти результати з моделлю, навченою виключно на реальних даних. Якщо використання аугментованих даних приведе до підвищення точності, цей метод можна буде інтегрувати у процес налаштування гіперпараметрів під час навчання моделі та комбінувати з іншими техніками для подальшого покращення її результатів.

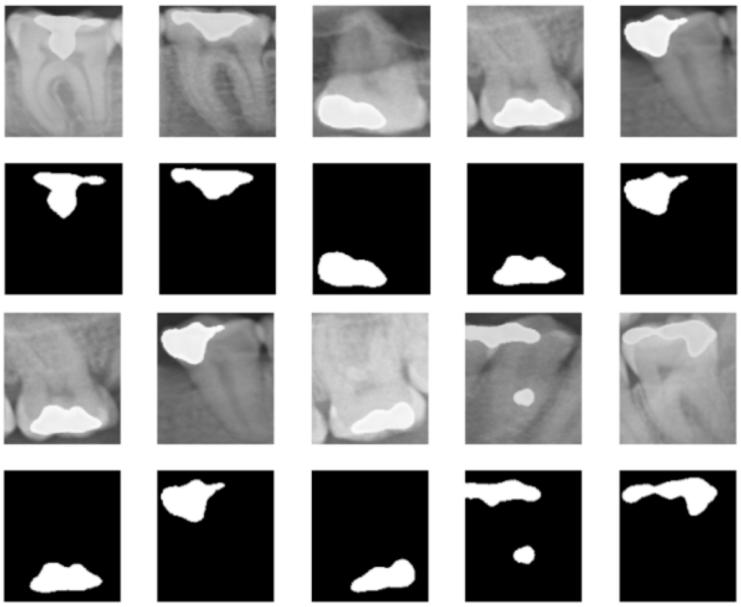


Рис. 2.32 Додані дані

В тренування не додавались картинки нульового класу, оскільки вони погіршують точність мітрик для всіх моделей в всіх експериментах.

Проведено п’ятдесят сессій тренуваня для кожного з двох експериментів, використовуючи різний випадковий стан починаючі від розділення даних на тренувальну та валідаційну вибірку. Використані наступні гіперпараметри для кожної з сессій тренування:

- модель: U-Net;
- основа моделі: ResNet-34;
- метрика: коефіцієнт Жаккара, який розраховується за формулою 2.10;
- функція втрат: функція втрат Жаккара;
- кількість епох: 50;

- зупинка через 5 епох, якщо немає покращення валідаційної метрики;
- розмір пакету: 32;
- оптимізатор: Adam;
- швидкості навчання: 0.001.

$$IoU(A,B) = \frac{|A \cap B|}{|A \cup B|}, \quad (2.10)$$

де A – згенерована сегментаційна маска, B – передбачена сегментаційна рамка.

Оцінка моделей проводилася на 20 зображеннях, залишених для тестування. Тренувальні, валідаційні та тестові показники для обох наборів даних наведені у таблиці 2.1. Розподіл тестових метрик (коефіцієнтів Жаккара), представлений на рисунку 2.33. Середнє значення метрики тестового коефіцієнту Жаккара для експерименту тільки з базовими даними склало 94.2, а для експерименту з базовими та доданими згенерованими даними 96.1. Різниця складає майже 2 відсотки.

Таблиця 2.1 Результати сегментаційної моделі для різних випадкових станів

Seed	**Базові**			**Базові + 64 Згенеровані**		
	Тест	**Валідація**	**Тренування**	**Тест**	**Валідація**	**Тренування**
0	0.941	0.955	0.937	0.940	0.956	0.948
1	0.944	0.958	0.949	0.965	0.962	0.967
2	0.957	0.957	0.949	0.948	0.942	0.948
3	0.947	0.951	0.956	0.971	0.952	0.977
4	0.966	0.961	0.961	0.961	0.958	0.956
5	0.956	0.948	0.951	0.958	0.948	0.948
6	0.956	0.961	0.965	0.955	0.952	0.950
7	0.955	0.949	0.943	0.970	0.957	0.979

Продовження до табл. 2.1

Seed	Базові			Базові + 64 Згенеровані		
	Тест	Валідація	Тренування	Тест	Валідація	Тренування
8	0.947	0.955	0.931	0.962	0.956	0.953
9	0.957	0.948	0.957	0.968	0.958	0.972
10	0.955	0.953	0.948	0.966	0.954	0.969
11	0.961	0.953	0.947	0.970	0.957	0.983
12	0.950	0.942	0.954	0.955	0.934	0.949
13	0.958	0.951	0.959	0.966	0.952	0.966
14	0.944	0.964	0.954	0.966	0.966	0.968
15	0.953	0.951	0.954	0.949	0.946	0.936
16	0.951	0.955	0.962	0.968	0.959	0.971
17	0.952	0.956	0.956	0.971	0.963	0.971
18	0.955	0.946	0.936	0.968	0.959	0.969
19	0.956	0.961	0.957	0.966	0.964	0.967
20	0.918	0.930	0.918	0.946	0.942	0.931
21	0.945	0.957	0.944	0.966	0.959	0.964
22	0.957	0.963	0.950	0.969	0.965	0.971
23	0.949	0.954	0.927	0.960	0.955	0.948
24	0.954	0.961	0.890	0.966	0.969	0.968
25	0.908	0.912	0.943	0.971	0.960	0.968
26	0.960	0.966	0.958	0.931	0.940	0.871
27	0.938	0.945	0.940	0.967	0.966	0.967
28	0.951	0.960	0.959	0.966	0.964	0.963
29	0.794	0.793	0.842	0.965	0.953	0.964
30	0.952	0.956	0.881	0.961	0.954	0.952

Продовження до табл. 2.1

Seed	Базові			Базові + 64 Згенеровані		
	Тест	**Валідація**	**Тренування**	**Тест**	**Валідація**	**Тренування**
31	0.957	0.959	0.961	0.965	0.962	0.965
32	0.835	0.840	0.914	0.966	0.957	0.966
33	0.942	0.947	0.939	0.960	0.952	0.958
34	0.946	0.946	0.948	0.969	0.959	0.972
35	0.928	0.923	0.928	0.959	0.929	0.950
36	0.956	0.957	0.957	0.965	0.956	0.964
37	0.957	0.943	0.959	0.956	0.934	0.954
38	0.948	0.952	0.951	0.964	0.954	0.955
39	0.959	0.948	0.957	0.964	0.949	0.960
40	0.924	0.934	0.917	0.958	0.954	0.955
41	0.945	0.957	0.944	0.963	0.960	0.955
42	0.950	0.941	0.868	0.969	0.953	0.968
43	0.945	0.951	0.941	0.959	0.961	0.947
44	0.968	0.958	0.969	0.962	0.953	0.965
45	0.947	0.943	0.945	0.963	0.948	0.963
46	0.955	0.958	0.955	0.969	0.964	0.971
47	0.955	0.959	0.963	0.966	0.960	0.965
48	0.947	0.948	0.941	0.969	0.958	0.971
49	0.953	0.956	0.956	0.960	0.954	0.954

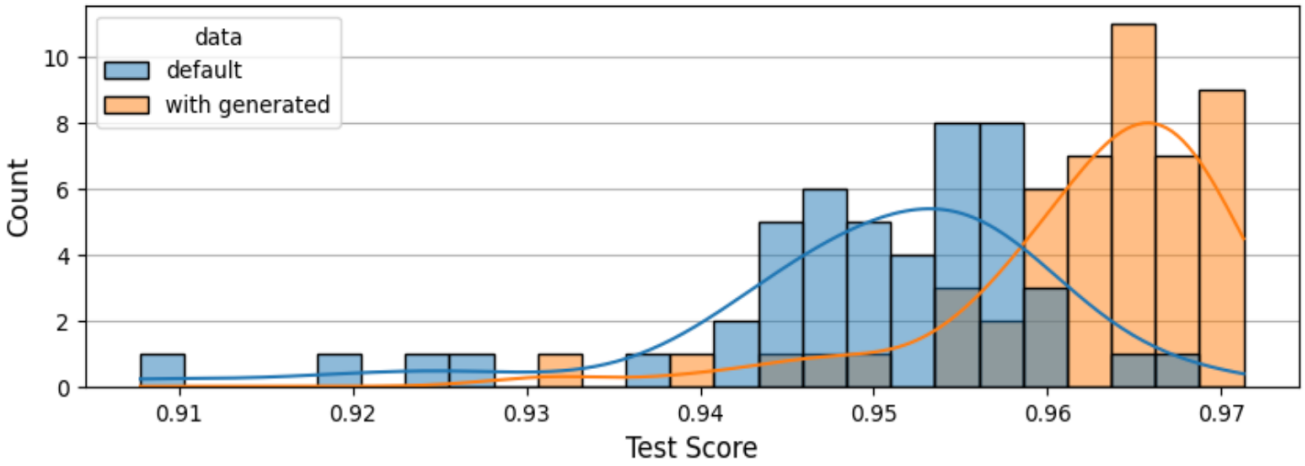


Рис. 2.33 Розподіл метрики сегментації на тестових даних

Також було проведено дослідження ефекту кількості згенерованих даних на якість моделей. В якості методу дослідження було обрано поступове збільшення кількості згенерованих зображень від 16 до 128, із кроком 16 зображень. Для кожної кількості було натреновано 50 моделей, що дало можливість оцінити вплив випадкових варіацій на результати та зменшити вплив випадковості на остаточні висновки.

На рисунку 2.34 показано результати цього експерименту, де відображено середні значення продуктивності моделей та їх стандартні відхилення на тестовому наборі даних.

З аналізу графіку можна зробити декілька важливих висновків:

1. Оптимальна кількість згенерованих зображень: Найбільше покращення продуктивності моделей спостерігається при додаванні від 64 до 80 згенерованих зображень. Це свідчить про те, що ця кількість є оптимальною для тренування моделей у даних умовах.

2. Зменшення ефективності з подальшим збільшенням: кількості Після додавання 80 зображень подальше збільшення кількості згенерованих даних не призводить до значного покращення якості моделей. Це може свідчити про те, що модель досягає своєї межі насичення, і додаткові дані не надають додаткової інформації, яка б покращувала результати.

3. Ресурсна ефективність: З кожним додаванням даних тренування моделей стає більш ресурсномістким, і після досягнення оптимальної кількості (64-80 зображень) подальше збільшення ресурсів на тренування не виправдовується, оскільки не приносить значного покращення результатів.

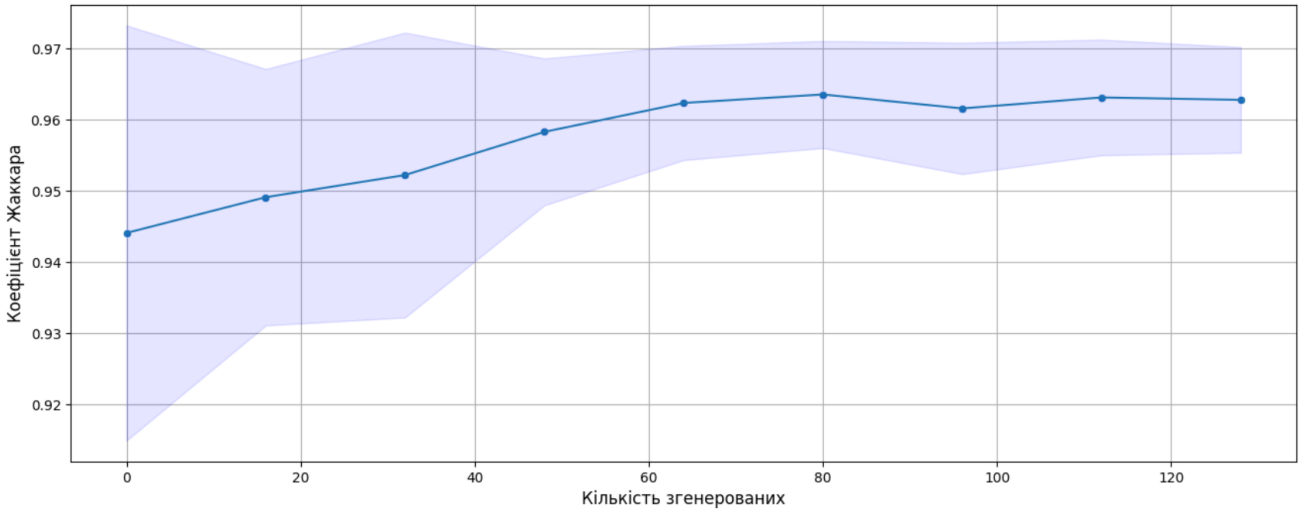


Рис. 2.34 Розподіл метрики сегментації на тестових даних для різної кількості згенерованих даних

2.4 Удосконалення архітектури ГЗМ для мінімізації змін вхідного зображення шляхом інтеграції компонента самовалідації

Незважаючи на позитивні результати моделі та підвищення точності в задачі сегментації пломб на окремих зображеннях зубів, попередній підхід має певні недоліки, продемонстровані на рисунку 2.35. Зокрема, зображення ліворуч (верхня щелепа, другий зуб зліва — 16 за Міжнародною двоцифровою системою нумерації зубів [96], і перший зуб справа — 27) та зображення праворуч (перший зуб зліва на верхній щелепі — 17 та нижній — 47) ілюструють проблему. Аугментовані зображення могли змінити контрастність, насититися артефактами або шумом. Для вирішення цього питання було запроваджено додатковий підхід, який передбачає агрегацію оригінального зображення з трансформованим по

місцях. Однак, цей алгоритм має недолік, оскільки може не враховувати окремі змінені деталі нових зображень, які не потрапляють під маску.

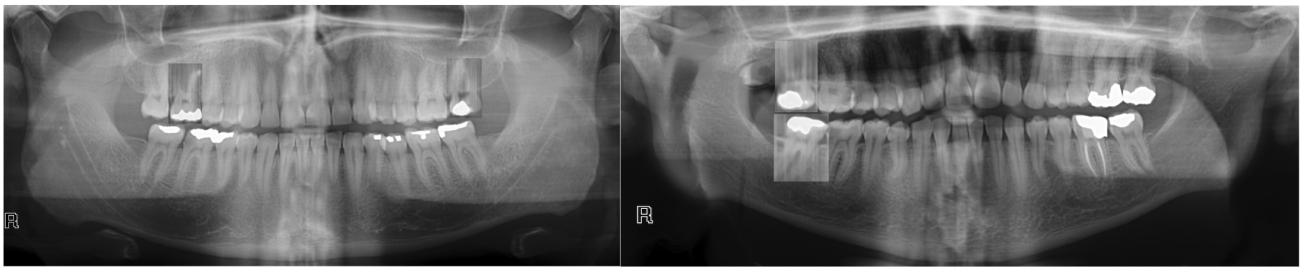


Рис. 2.35 Приклад недоліків генерування попереднього підходу

В цьому розділі пропонується покращити це рішення, шляхом додавання нового компонента контролю змін на зображенні. Удосконалена архітектура рішення [4] показана на рисунку 2.36.

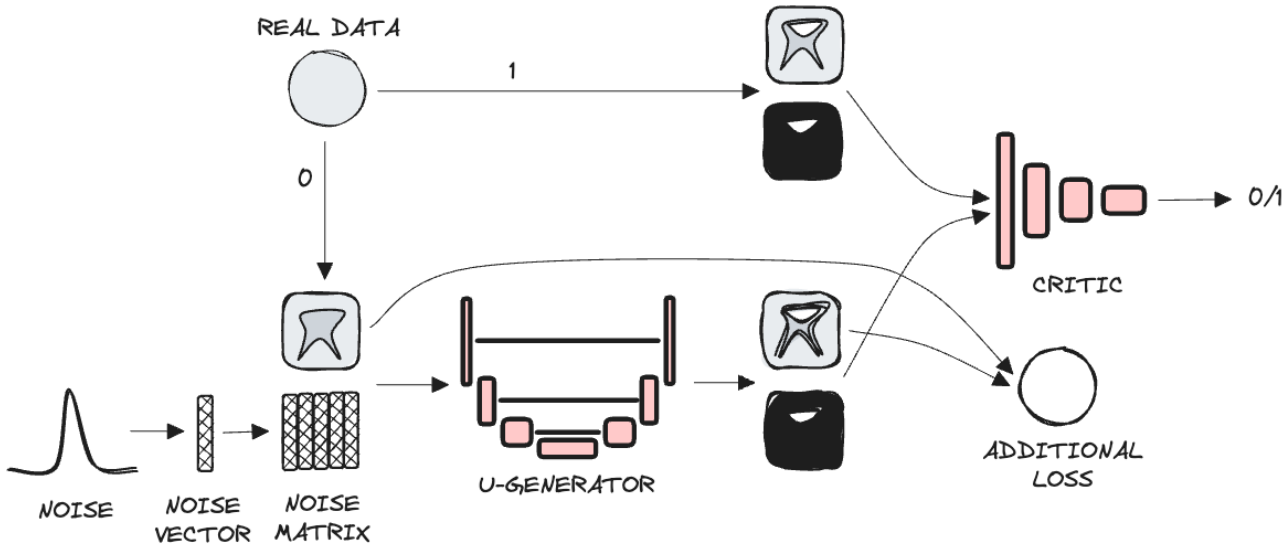


Рис. 2.36 Схема вдосконаленої трансформації цільового класу сегментації

Додатковий компонент, що зображено на рисунку під назвою Additional Loss (додаткова втрата) було впроваджено з метою покращення стабільності та точності генерації зображень. Його основна задача полягає у введенні додаткового штрафу за надмірні зміни, які можуть виникати під час трансформації зображення. Це відрізняє цей компонент від стандартного критика, який оцінює якість

згенерованого зображення лише з точки зору його відповідності розподілу реальних зображень.

Основна проблема, яку намагається вирішити цей компонент, стосується надмірної трансформації зображень під час переходу від фонового класу до цільового. Іншими словами, коли генератор змінює зображення занадто сильно, це може призводити до втрати важливих характеристик оригіналу. Щоб уникнути цього, Additional Loss додає штраф рівноцінний тому, наскільки зображення після обробки відрізняється від вхідного.

Це рішення допомагає не тільки зберігати первісні характеристики зображення, але й скорочує час, необхідний для навчання моделі. Завдяки тому, що цей компонент мінімізує небажані зміни, генератор може швидше досягати реалістичних результатів, зменшуючи кількість епох, необхідних для досягнення високої якості зображень. Таким чином, компонент Additional Loss забезпечує більш ефективне та точне генерування зображень, зберігаючи при цьому їхню оригінальну сутність.

Далі описано гіперпараметри цього компонента.

Функція втрат. Може мати такі види:

Середньоквадратичне відхилення (Mean Squared Error, MSE) [97] – це міра середнього квадратичного відхилення прогнозованих значень від фактичних значень, яка часто використовується для оцінки точності регресійних моделей. Описується наступною формулою:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2, \tag{2.11}$$

де n – кількість спостережень, y_i – фактичне значення для і-го спостереження, y_i' – прогнозоване значення для і-го спостереження.

MSE завжди не від’ємне, оскільки квадрати відхилень завжди додатні або рівні нулю. MSE чутливе до великих відхилень, оскільки квадрати відхилень підсилюють вплив великих помилок.

Кореневе середньоквадратичне відхилення (Root Mean Squared Error, RMSE) [98] – це корінь квадратний з середньоквадратичного відхилення. Використовується для інтерпретації MSE в тих же одиницях, що і вихідні дані:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2}, \quad (2.12)$$

RMSE виражається в тих же одиницях, що і фактичні значення y, що робить його більш інтерпретованим у контексті конкретних даних. Як і MSE, RMSE також чутливе до великих відхилень.

Абсолютне відхилення (Mean Absolute Error, MAE) [99] – це середнє значення абсолютних відхилень між фактичними та прогнозованими значеннями. MAE показує середню величину помилки без врахування її знаку:

$$MAE = |y_i - y_i'|, \quad (2.13)$$

MAE дає більш зрозумілу оцінку середньої помилки, оскільки не підсилює вплив великих відхилень, як це робить MSE. MAE не так сильно реагує на великі відхилення, що робить його більш стійким до викидів у даних.

Середнє абсолютне процентне відхилення (Mean Absolute Percentage Error, MAPE) [100] – це середнє значення абсолютних процентних помилок між прогнозованими і фактичними значеннями. Ця метрика часто використовується, коли важливо знати відносну похибку в процентах:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{|y_i - y_i'|}{y_i}, \quad (2.14)$$

MAPE виражає помилку у відсотках, що робить її зручною для інтерпретації в багатьох прикладних задачах. MAPE не підходить для випадків, коли y може бути нулем, оскільки це призводить до невизначеності в обрахунку.

Коефіцієнт впливу. Цей коефіцієнт контролює вплив описаної раніше функції на загальний штраф. Може бути більше одиниці (в якості основної втрати) або менше одиниці (в якості додаткової втрата), але більшим за 0. Сила впливу контролюється вибраною математичною моделлю і може зменшуватись в залежності від номеру поточної ітерації чи епохи.

Наприклад, спочатку визначати великий штраф за різницю з вхідним зображенням, щоб генератор зрозумів ідею та концепт, а потім зменшувати його, даючі можливість моделі більше вчитися від критика, який «знає», як саме має виглядати об’єкт класу. Функція може бути будь-якою: мати лінійний характер, вигляд сигмоїди, експоненційний, бути константною.

Тепер функцію втрат підходу можна виразити у наступному вигляді:

$$TotalLoss = CriticLoss + X * SimilarLoss, \quad (2.15)$$

де $CriticLoss$ – штраф критика, $SimilarLoss$ – додаткова функція втрат, X – коефіцієнт впливу розробленого компонента.

У випадку використання MSE як додаткової функції втрат, формула 2.15 набуває такого вигляду:

$$TotalLoss = CriticLoss + X * \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2, \quad (2.16)$$

2.5 Удосконалення архітектури ГЗМ для оптимізації вибору найкращого стану генератора через інтеграцію компонента самовалідації

Залишається проблема автоматичного контролю над вибором кращої епохи та ранньої зупинки, якщо модель перенавчилася або увійшла у колапс мод. Для рішення цієї проблеми додано компонент самооцінки, що описано у цьому розділі. Удосконалена архітектура рішення [4] показана на рисунку 2.37.

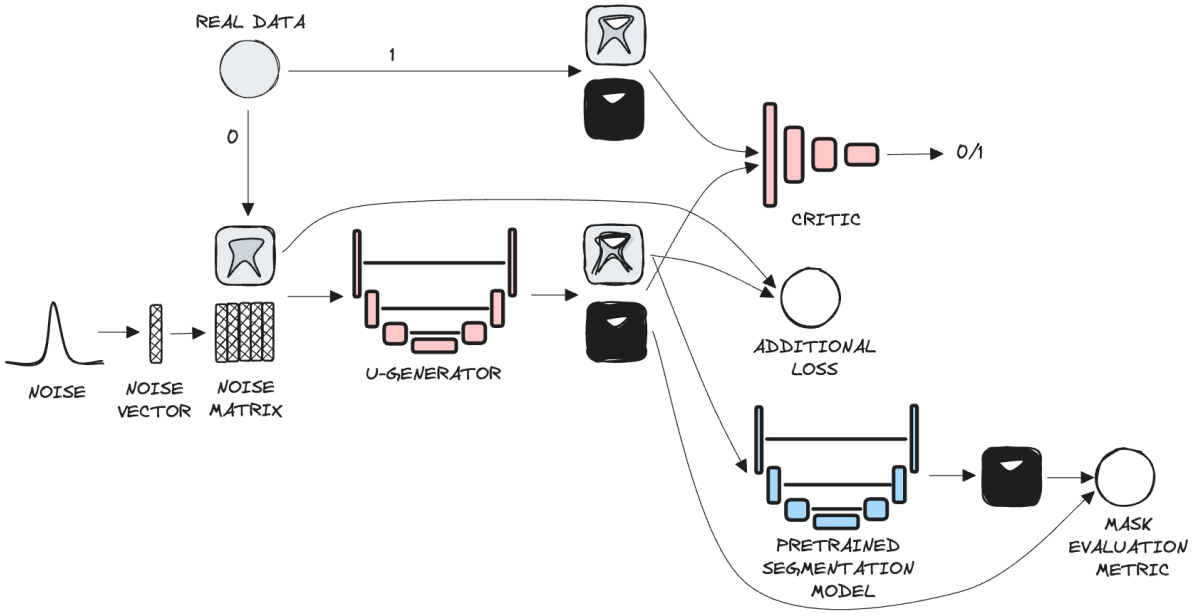


Рис. 2.37 Схема вдосконаленої трансформації цільового класу сегментації

Pretrained Segmentation Model (претренована сегментаційна модель) є важливим компонентом як у цьому підході, так і у задачах комп'ютерного зору,

зокрема для сегментації об'єктів на зображеннях [101]. Ця модель спочатку тренується на доступних даних, що належать до цільового класу, наприклад, об'єктів певного типу як пломби. Основна задача цієї моделі полягає в тому, щоб на основі вхідного зображення, створеного генеративною моделлю, побудувати маску, яка відокремлює цільовий об'єкт від фону.

Ця модель виконує роль додаткової моделі, яка навчається на зображеннях, отриманих від генератора, з метою знайти точну маску для цільового класу. Це важливо для того, щоб зрозуміти, наскільки генератор здатний створювати реалістичні зображення, які можуть бути коректно сегментовані. Як приклад, на рисунку 2.38 демонструється робота цієї моделі на зображеннях, отриманих на ранніх етапах навчання. Перший рядок на рисунку показує вхідне зображення разом із згенерованою маскою, яка була створена під час генерації зображення. Другий рядок демонструє те ж саме зображення, але вже з маскою сегментації, яку побудувала претренована модель.

Важливо зазначити, що на ранніх етапах навчання генеративної моделі на зображеннях часто можуть з’являтися артефакти, які ускладнюють процес формування якісної сегментаційної маски. Це, в свою чергу, впливає на точність оцінювання якості маски, оскільки модель може некоректно сегментувати об'єкт, що призводить до зниження ефективності метрик оцінки якості сегментації.

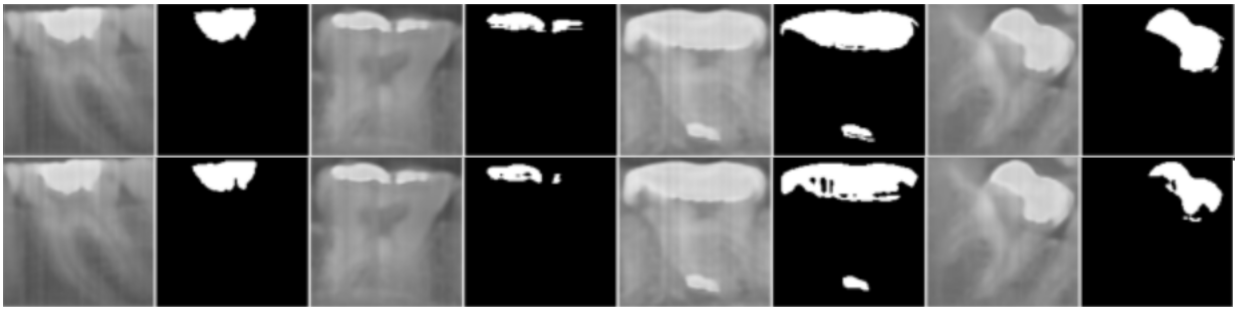


Рис. 2.38 Приклад генерування зображення та маски (перший ряд) та результат від сегментаційної моделі (другий ряд) на етапі навчання ГЗМ

Mask Evaluation Metric (метрика оцінки маски) є ключовим інструментом для визначення точності та якості роботи самовалідованих генеративних

змагальних мереж для задач сегментації картинок. Ця метрика призначена для порівняння маски, яка була згенерована генеративною моделлю, з еталонною маскою, яку формує сегментаційна модель. Високе значення метрики свідчить про те, що згенерована маска добре відповідає реальному об'єкту на зображенні, і претренована модель здатна чітко виділити цей об'єкт, що підтверджує точність генерації.

Метрика оцінки маски слугує критерієм порівняння між двома масками: згенерованою генератором та створеною моделлю сегментації. Чим більша схожість між цими масками, тим вище значення метрики, що свідчить про успішність генеративного процесу. Однією з найпоширеніших метрик для такого порівняння є коефіцієнт Жаккара [102]. Цей коефіцієнт (формула 2.10) дозволяє кількісно оцінити ступінь просторового перекриття між прогнозованою маскою та еталонною, де значення коефіцієнта варіюється від 0 до 1: 0 означає повну відсутність перекриття, а 1 – повне співпадіння.

Також є і інші метрики які можуть тут підходити, що робить цей компонент додатковим параметром що можна налаштовувати. Наприклад:

Індекс Дайса [103] тісно пов’язаний з коефіцієнтом Жаккара і часто використовується у задачах сегментації зображень:

$$Dice(A, B) = \frac{2\times|A\cap B|}{|A|+|B|}, \tag{2.17}$$

де A – згенерована сегментаційна маска, B – передбачена сегментаційна рамка.

Індекс Дайса також варіюється від 0 до 1, де 1 означає ідеальне співпадіння між двома множинами.

Таким чином, не залежно від вибору конкретної метрики, якщо маска, згенерована генератором, точно відповідає об'єкту на зображенні, і об'єкт настільки чітко визначений, що претренована модель здатна побудувати його точну маску, значення метрики оцінки буде високим. Це є свідченням високої якості генеративної моделі та її здатності створювати реалістичні й точні зображення.

На рисунку 2.39 (верхній графік) показано зміну коефіцієнта Жаккара протягом навчання однієї з моделей із використанням цього підходу. Чітко видно момент, коли модель почала генерувати результати, які хоч трохи нагадують реальні — це сталося приблизно на 60-й епосі (сегментаційна модель почала коректно визначати маску, і вона почала збігатися зі згенерованою). Подальші експерименти з моделлю показали, що найкращі епохи за метрикою Жаккара переважно збігаються з відібраними вручну. Для цього було обрано 10 перших епох за метрикою Жаккара та вручну відібрано 10 найкращих результатів серед 150 епох. В середньому збіг номерів епох між автоматично визначеними та вручну відібраними становить 70% (тобто 7 з 10 епох співпали), використовуючи різні значення для випадкових станів (random seed).

Для порівняння, на нижньому графіку рисунку 2.39 показано зміну функцій втрат генератора та критика залежно від номеру епохи. Однак, вибрати найкращу епоху на основі цих показників складно, оскільки вони не корелюють з реальною якістю генерації, і відсутній чіткий критерій, яке значення втрат є оптимальним з точки зору моделі.

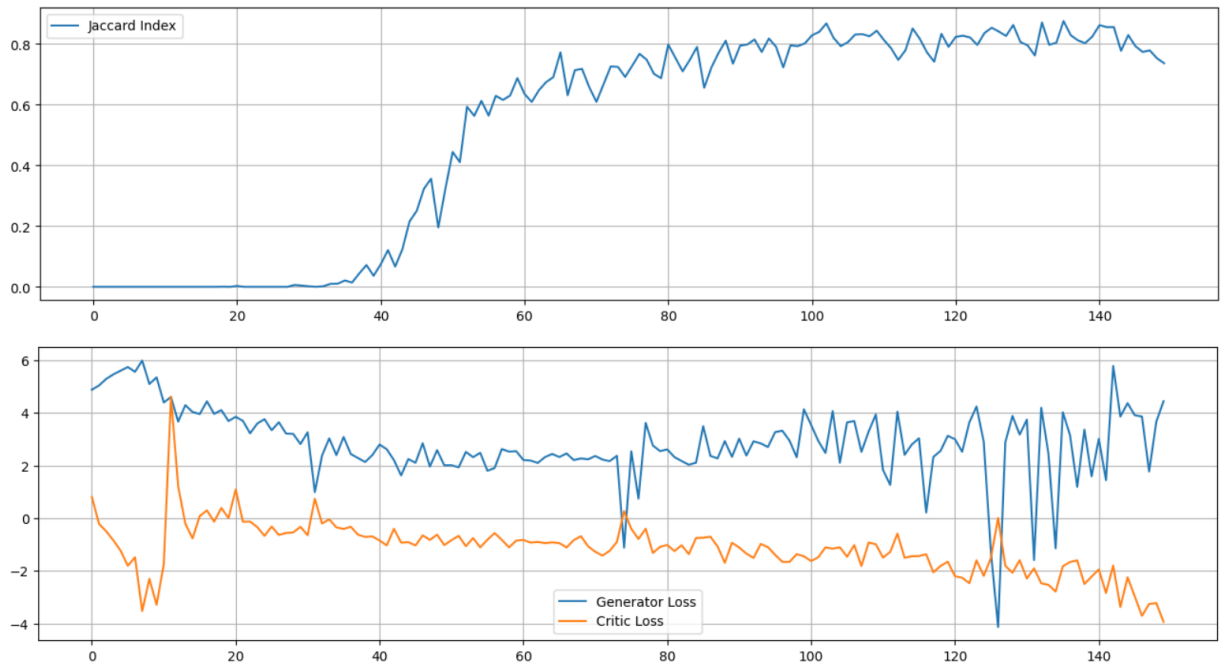


Рис. 2.39 Графіки зміни значень коефіцієнта Жаккара, функції втрат генератора та критика

2.6 Висновки до розділу

Розділ присвячено розробленню методів та теоретичних основ технології самовалідованих генеративних змагальних мереж.

Спроектовано основу для подальших експериментів, що представляє собою механізм тренування ГЗМ Вассерштейна. Як експериментальний набір даних вибрано власноруч зібраний набір даних квітів, що складається із 900 зображень.

У першому експерименті було підготовлено пайплайн тренування для зображень з роздільною здатністю 128x128 пікселів. Вирішено проблему артефактів, зокрема ефекту шахової дошки, який часто зустрічається в генеративних моделях. Це було досягнуто завдяки відмові від використання шарів транспонованої згортки, які відповідали за збільшення розміру зображення та його коректність. Натомість як основні блоки використовувалися шари звичайної згортки для покращення якості зображень і формування важливих ознак, а також шари білінійного підвищення дискретизації для збільшення розміру проміжних карт ознак.

Ця архітектура також апробована в задачі генерування цифр ID карток для збагачення набору даних паспортів для перевірки можливостей відкритих OCR систем.

Після цього вперше адаптовано задану архітектуру для задачі перетворення нульового класу у цільовий з одночасних створенням сегментаційних масок, для цього виконані наступні кроки:

- Заміна класичного генератора U-генератором, що може приймати на вхід одне зображення а видавати інше – трансформоване;
- Зміна підходу до генерування – на вхід генератору подається картинка нульового класу та згенерована матриця шуму в якості додаткового каналу;
- На виході генератор віддає трансформовану картинку та маску цільового об’єкта;

- Критик зазнає мінімальних змін приймаючі на вхід тепер додатковий канал – маску реальну або згенеровану.

Хоч даний підхід демонструє гарні результати, але він досить сильно трансформує вхідне зображення, що не дає змогу використовувати його для часткової аугментації. В якості додаткового рішення, було спроектовано алгоритм об’єднання вхідної картинки, з обмеженими по згенерованій масці ознаками створенного зображення.

Далі вперше запропоновано самовалідовану ГЗМ для задачі генерування сегментаційної маски та трансформації фону у цільовий клас. Для цього розроблено наступні компоненти самовалідації в складі архітектури ГЗМ:

- Additional Loss – це компонент, який додає додатковий штраф до основної оцінки критерію, залежно від того, наскільки значно було змінено зображення. Його впровадження спрямоване на підвищення стабільності та точності процесу генерації.

- Pretrained Segmentation Model – сегментаційна модель, навчена на наявних даних для цільового класу. Її завданням є прийом згенерованого зображення та побудова маски для відповідного цільового об'єкта.

- Mask Evaluation Metric – метрика, яка порівнює маску, згенеровану генератором, із маскою, створеною претренованою сегментаційною моделлю. Якщо маска генератора відповідає згенерованому об'єкту і цей об'єкт настільки точний, що претренована модель здатна визначити його маску, значення метрики буде високим, що вказує на високу якість моделі.

Результати розділу опубліковано в таких роботах автора: [1], [2], [3], [4].

РОЗДІЛ 3
РОЗРОБКА МЕТОДУ ОЦІНЮВАННЯ ЧАСТКОВО ЗГЕНЕРОВАНИХ ДАНИХ

3.1 Потреба у розробці метрик для оцінювання частково аугментованих даних

Часткове генерування даних має кілька значних переваг перед звичайний генеруванням даних. Однією з переваг є можливість ефективно трансформувати зображення високої роздільної здатності, не вимагаючи значних обчислювальних ресурсів. Це досягається завдяки тому, що зміни застосовуються лише до окремих частин зображення, а не до всього зображення цілком.

Розглянемо приклад зображення з роздільною здатністю 1920 на 1080 пікселів. Замість того, щоб трансформувати повністю все зображення, аугментація може бути застосована лише до невеликої його частини, наприклад, до області розміром 256 на 256 пікселів. Це дозволяє значно зменшити обчислювальне навантаження та прискорити процес підготовки даних.

Зменшення обчислювальних витрат. Трансформація великого зображення вимагає значних обчислювальних ресурсів, особливо якщо мова йде про складні аугментаційні техніки, такі як обертання, зміна кольорової гами або додавання шуму. Застосування цих трансформацій лише до невеликої частини зображення дозволяє скоротити час обробки та знизити навантаження на апаратне забезпечення.

Часткова аугментація дозволяє зберегти основну структуру та важливі характеристики вихідного зображення, одночасно збільшуючи його різноманітність. Це особливо корисно у випадках, коли важливо зберегти контекстну інформацію, а зміни вносяться лише до менш значущих частин зображення. Такий підхід може покращити продуктивність моделі, оскільки вона навчається розпізнавати важливі деталі на фоні варіативності.

Часткова аугментація дозволяє більш гнучко підходити до процесу підготовки даних. Наприклад, можна вибірково трансформувати лише ті області зображень, які потребують додаткової варіативності для покращення навчання моделі. Це може бути особливо корисним у випадках, коли деякі області зображення мають більше значення для класифікації або виявлення об'єктів, ніж інші.

Уявімо задачу розпізнавання обличь на зображеннях високої роздільної здатності. Замість того, щоб аугментувати все зображення, можна застосувати трансформації лише до обличь або окремих частин обличчя (очей, носу, роту, волосся), що значно скоротить час обробки і збереже важливу контекстну інформацію. Наприклад, обертання, додавання шуму або зміна яскравості може бути застосовано лише до невеликої області навколо очей, тоді як решта зображення залишиться незмінною. Це дозволить моделі краще навчатися на різноманітних прикладах, не вимагаючи при цьому надмірних обчислювальних ресурсів.

Разом із перевагами часткова аугментація даних додає викликів. Наприклад, при оцінюванні якості трансформацій картинки за допомогою методу, такого як FID, можуть виникати труднощі з виявленням значних відмінностей, якщо аугментований фрагмент є невеликим.

Це добре демонструє наведений нижче приклад. На рисунку 3.1 зображено чотири рентгенівські знімки. Верхнє ліве зображення є оригінальним, тоді як на решті зображень трансформовано область, виділену на першому зображенні червоною рамкою. Кожне з інших зображень є результатом різних моделей трансформацій, що описані у Розділі 2 цього дослідження.

Зображення у верхньому правому куті є результатом генеративної моделі на основі U-генератору (розділ 2.3), яка сильно спотворює вхідний фрагмент зображення, роблячи його вигляд ненатуральним і не даючи змогу правильно додати його в контекст знімку. Два зображення внизу отримані за допомогою більш якісних методів: ліве за допомогою агрегації вхідного зображення та маски

від першої моделі (розділ 2.3), а праве з використанням самовалідованої генеративної змагальної мережі (розділ 2.5).

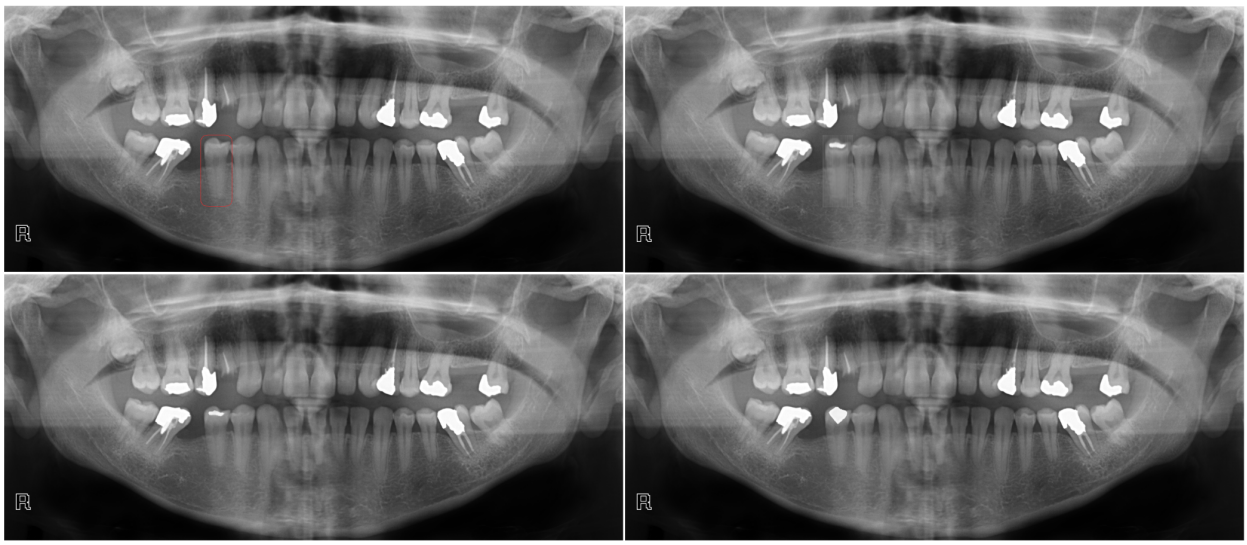


Рис. 3.1 Приклад часткового генерування з використанням різних підходів

Очікується, що метрика якості має відображати кількісно суттєві дефекти у поганої моделі. Для перевірки цього було взято 20 зображень та для кожного з них трансформовано 1-2 зуби (див. Додаток А). Додатково взято 10 зображень, без трансформації, щоб оцінити наскільки сильно відрізняються метрики та розподіли у порівнянні з узагалі іншими зображеннями.

На рисунку 3.2 показано попарне значення FID оцінки для кожного набору зображень:

- input images – нетрансформовані зображення,
- model_1 – трансформовані вхідні зображення з використанням першої моделі на основі U-GAN,
- model_2 – трансформовані вхідні зображення з використанням другої моделі з використанням після-обробки,
- model_3 – трансформовані вхідні зображення з використанням третьої моделі самовалідованої ГЗМ,
- additional images – додатковий набір нетрансформованих зображень.

Як показано на зображенні, немає чіткої різниці між добре аугментованими зображеннями, погано аугментованими та оригінальним зображенням. Але додаткові зображення кардинально на порядок відрізняються від всіх наборів даних.

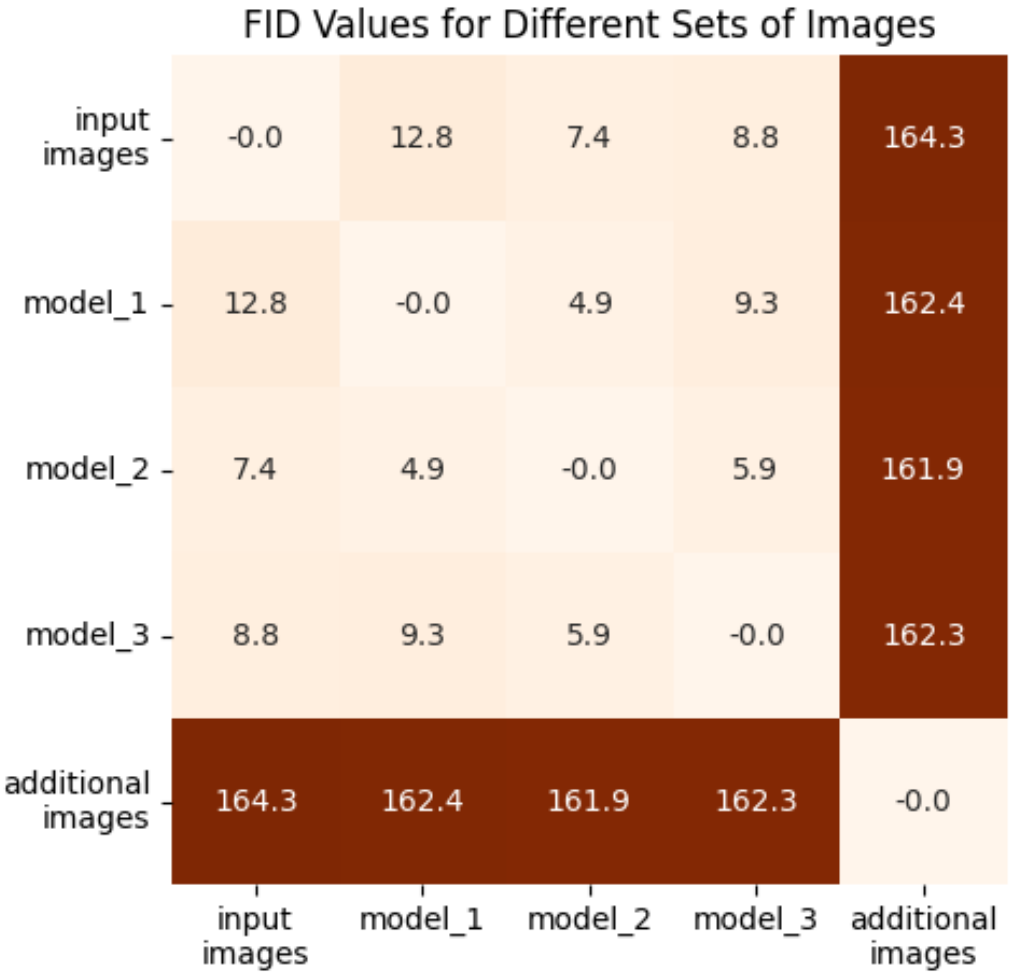


Рис. 3.2 Попарне значення FID для різних наборів даних

На рисунку 3.3 показано, який розподіл мають звичайні зображення та додаткові звичайні в двовимірному просторі отриманому за допомогою методу головних компонент. Зображення були оброблені моделлю InceptionV3 для отримання векторів ознак, що мають розмірність 2048. Після цього було застосовано метод головних компонент [104], що дозволив знизити розмірність векторів ознак з 2048 до двох основних компонент, для зручної візуалізації.

На лівій частині рисунку 3.3 показано розподіл реальних даних у двовимірному просторі (кожна біла точка – окреме зображення). На правій панелі

зображено розподіл додаткових зображень. Ці два розподіли мають досить різні параметри і відповідно форму.

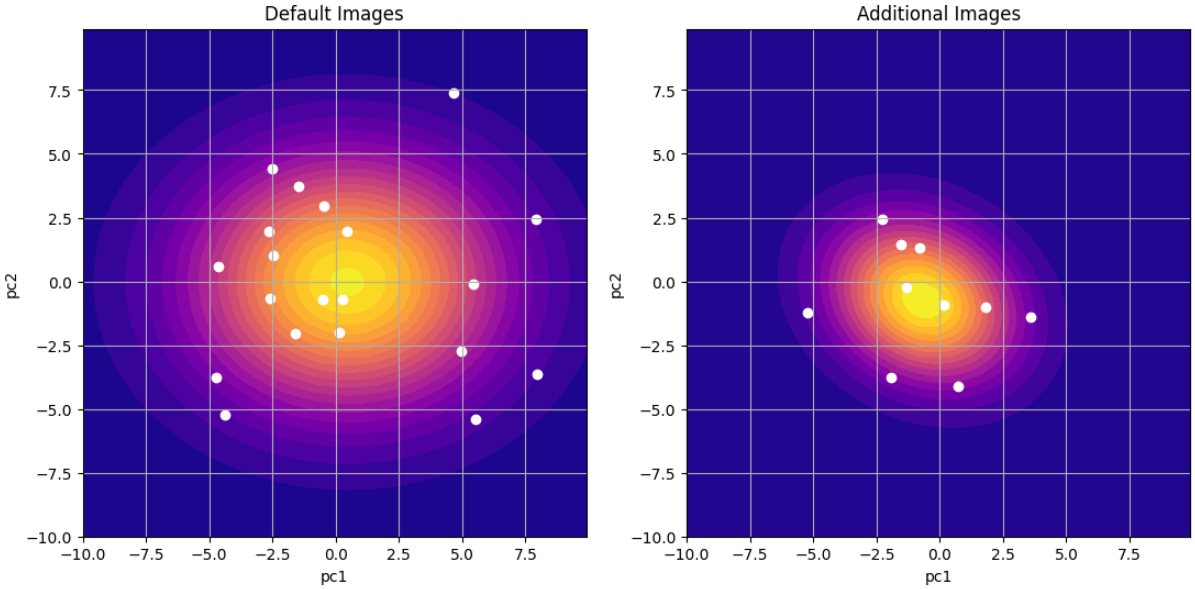


Рис. 3.3 Розподіли двох наборів реальних даних

На лівій панелі рисунку 3.4 показано розподіл реальних даних у двовимірному просторі. На правій панелі зображено розподіл трансформованих даних, отриманих за допомогою моделі на основні U-генератору без після-обробки. Як видно, обидва розподіли виглядають дуже схожими, що вказує на те, що суттєвої різниці між ними немає.

Ліва панель графіка демонструє, що реальні дані мають концентрований розподіл з чітко вираженим центром і поступовим зменшенням щільності до країв. Це свідчить про те, що реальні зображення мають схожі ознаки, які утворюють однорідний кластер у двовимірному просторі.

Права панель графіка показує розподіл трансформованих даних. Хоча трансформовані дані також мають концентрований розподіл з центром, як і реальні дані, вони демонструють трохи більшу розбіжність у щільності до країв. Це може свідчити про те, що трансформації, внесені звичайною моделлю ГЗМ, незначно впливають на структуру ознак зображень, проте ця різниця не є достатньо значущою для суттєвого зміщення розподілу.

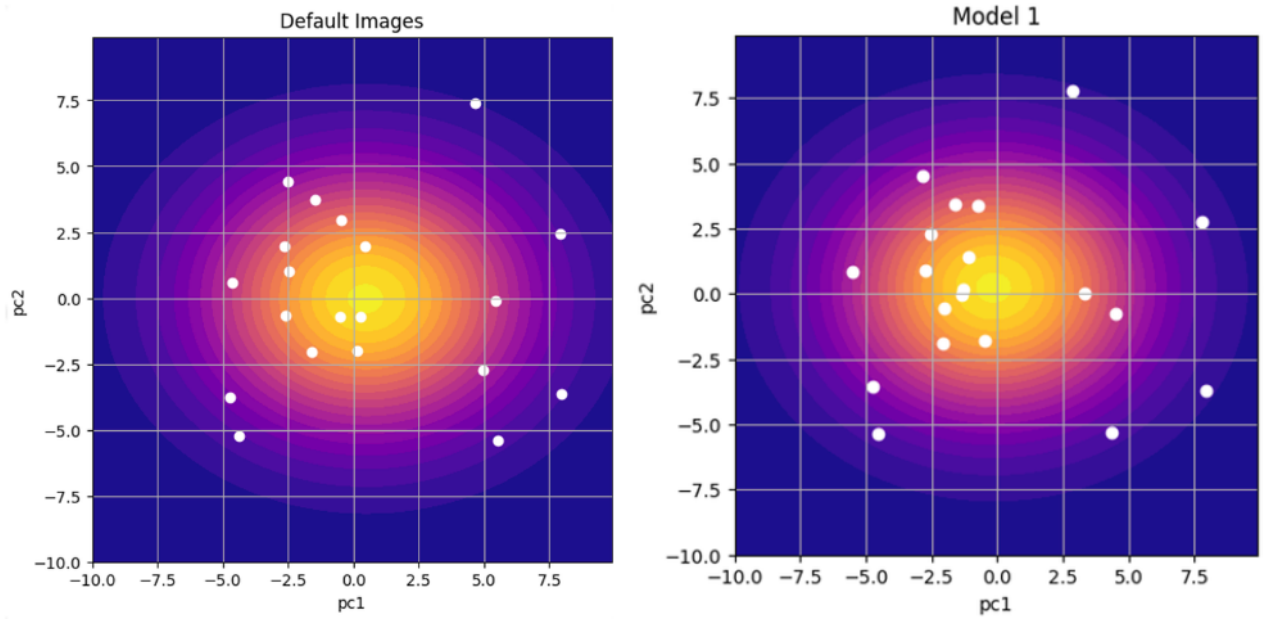


Рис. 3.4 Розподіли набору реальних даних, та трансформованим U-GAN

На рисунку 3.5 з правої частини показано розподіл реальних даних, з лівої трансформовані реальні дані за допомогою моделі U-генератору та використанням пост-обробки вихідного зображення та створеної сегментаційної маски пломби. Суттєвої різниці між формою розподілів або самих точок даних як і на рисунку з гіршою версією моделі немає.

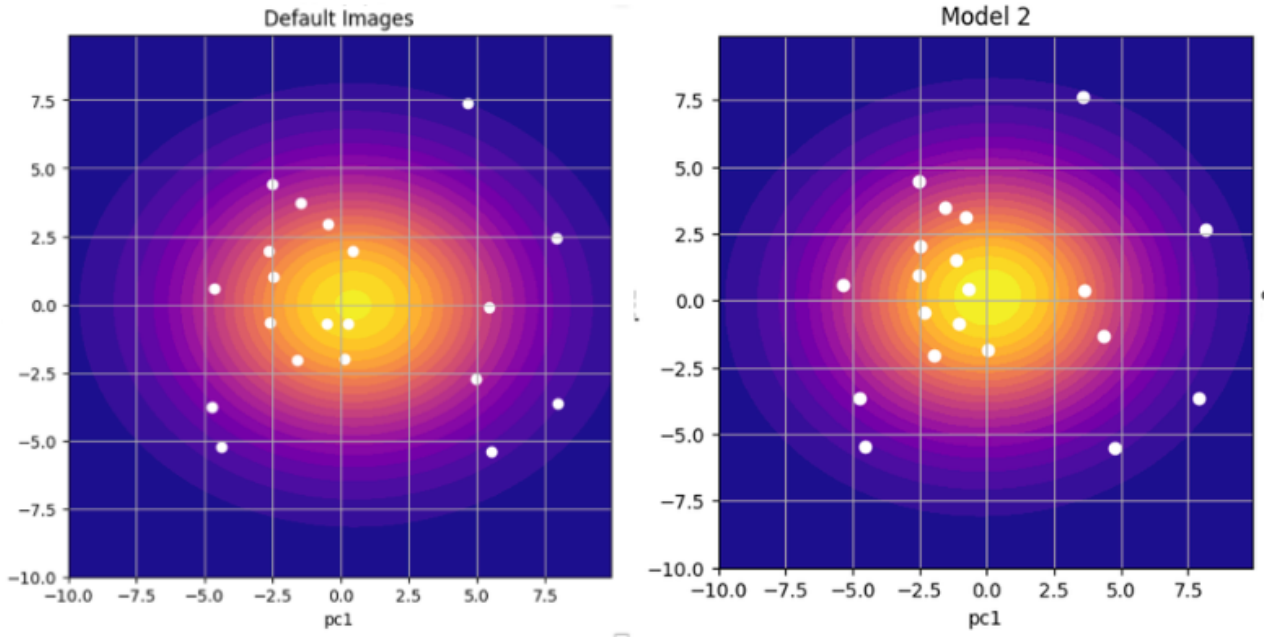


Рис. 3.5 Розподіли набору реальних даних, та трансформованим U-GAN з після-обробкою

Аналогічний експеримент було проведено з використанням самовалідаційної генеративної змагальної мережі для трансформації зображень. На рисунку 3.6 показано результати цього аналізу, де представлено розподіли реальних і трансформованих даних у двовимірному просторі після зниження розмірності за допомогою методу головних компонент.

Як і в попередньому експерименті, для зниження розмірності векторів ознак було застосовано метод головних компонент, що дозволило перевести вектори ознак у двовимірний простір для зручності візуалізації та порівняння розподілів реальних і трансформованих даних.

На лівій панелі рисунку 3.6 як і раніше показано розподіл реальних даних у двовимірному просторі. На правій панелі зображено розподіл трансформованих даних, отриманих за допомогою самовалідаційної ГЗМ. Обидва розподіли демонструють високу схожість, що вказує на збереження основної структури ознак після трансформації.

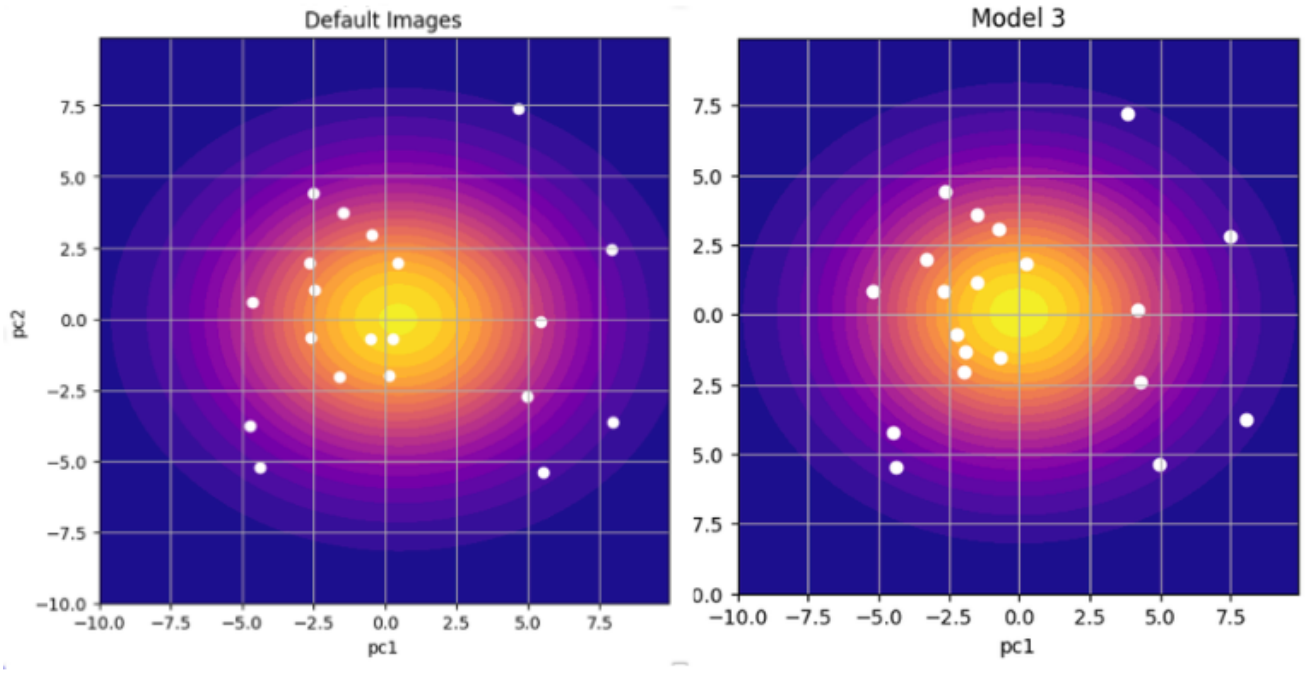


Рис. 3.6 Розподіли набору реальних даних, та трансформованим самовалідованою ГЗМ

Ручне порівняння результатів експерименту з використанням U-генератору, U-генератору з після-обробкою та самовалідаційної ГЗМ демонструє перевагу самовалідаційного підходу. Самовалідаційна ГЗМ краще зберігає структуру ознак вихідних даних, що видно з більшої схожості розподілів у двовимірному просторі. Але різниця в порівнянні є мінімальною, коли візуально ці підходи відрізняються суттєво.

Всі 3 розподіли мають дуже близьку природу до початкового розподілу. З наведеного вище випливає, що FID та інші методи екстрагування вектору ознак з цілого зображення для оцінювання частково згенерованих даних не підходять для оцінювання часткового згенерованих даних.

3.2 Проектування метрики для оцінювання частково аугментованих даних, створених генеративними моделями

Під час створення зображень за допомогою описаного у попередньому розділі підходу, генеруються частково аугментовані зображення. Це означає, що лише певні елементи зображення піддаються змінам. Наприклад, на комп'ютерній томографії знімку може бути змінений лише один зуб, або на супутниковому знімку – лише певна частина території. Такий підхід ускладнює завдання простого розподілу зображень на згенеровані і незгенеровані. Важливо точно визначити ті ділянки, які зазнали змін, оскільки це впливає на подальше аналізування і інтерпретацію результатів.

В роботі пропонується новий метод оцінювання частково згенерованих даних із залученням людини. На відміну від методу HYPE цей метод направлений на акцентування уваги саме на згенерованих частинах зображення і на те, наскільки добре людина здатна ідентифікувати ці частини.

Для вирішення цієї задачі розроблено підхід з участю людини. Сутність методу полягає в тому, що людина аналізує трансформоване зображення та ідентифікує змінені зони. Зміни можуть бути численними і різноманітними, тому завданням є знайти і визначити всі з них. Основні принципи наступні:

- Якщо людина правильно ідентифікує всі змінені ділянки, вона отримує максимальну оцінку, а модель, яка згенерувала зміни, – мінімальну.
- Якщо людина не визначає жодної зміненої ділянки, вона отримує мінімальну оцінку, а модель – максимальну.

Для об’єктивного оцінювання результатів використовуються кілька ключових метрик:

1. Влучність (Precision). Ця метрика показує, яка частка змін, знайдених людиною, дійсно є зміненими. Висока влучність свідчить про те, що експерт в основному правильно ідентифікує зміни з мінімальною кількістю хибних висновків:

$$Precision = \frac{TP}{TP+FP}, \tag{3.1}$$

де TP – кількість правильно знайдених зон, FP – кількість знайдених зон, що є помилковими.

2. Повнота (Recall). Ця метрика відображає, яка частка всіх змінених ділянок була знайдена людиною. Висока повнота свідчить про те, що експерт знаходить більшість або всі зміни, зроблені моделлю:

$$Recall = \frac{TP}{TP+FN}, \tag{3.2}$$

де TP – кількість правильно знайдених зон, FP – кількість незнайдених зон.

3. F1-міра (F1 score). Це гармонійне середнє між точністю та повнотою, яке забезпечує баланс між цими двома показниками. Висока F1-міра вказує на те, що і влучність, і повнота знаходяться на високому рівні:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \tag{3.3}$$

Для оцінки, наскільки точно користувач визначив змінені ділянки, використовується метрика IoU (індекса Жаккара). В контексті поточної задачі метрика визначає ступінь перетину області, вказаної користувачем, та реально згенерованої області. Чим більше співпадіння між цими областями, тим вищий показник IoU (див. формулу 2.10), що свідчить про більш точну ідентифікацію змін.

Алгоритм роботи розробленого методу [5], що показано на рисунку 3.7, включає наступні етапи:

1. Вибір зображень для оцінювального набору даних та вибір зон для трансформації: На першому етапі з вибраних зображень визначаються ділянки, які підлягатимуть трансформації. Ці ділянки позначені червоною рамкою на зображенні.

2. Трансформація вибраних ділянок генеративною моделлю. Кожне зображення піддається обробці генеративною моделлю, яка трансформує вибрані ділянки. У результаті отримуються трансформовані зображення та цільові координати боксів, де відбулась трансформація. Це дозволяє системі знати точні ділянки, що були змінені.

3. Оцінка трансформованих зображень користувачами. Трансформовані зображення надаються групі користувачів або експертів. Їм необхідно ідентифікувати ділянки, які вони вважають аугментованими (сині рамки на зображенні), тобто такими, що зазнали змін. Користувачі не знають точних зон трансформації, що дозволяє оцінити, наскільки непомітні або якісні ці зміни.

4. Розрахунок метрик для кожного користувача і зображення. На основі точності визначення ділянок, що були аугментовані на конкретному зображенні, для кожного користувача розраховуються індивідуальні метрики. Це включає такі показники, як влучність, повнота, F1.

5. Агрегація результатів серед всіх користувачів. Результати оцінки всіх експертів агрегуються для отримання метрики для кожного окремого зображення. Ця метрика відображає загальну оцінку якості трансформації наданої генеративною моделлю для одного зображення.

6. Агрегація результатів по всім картинкам. На цьому останньому етапі відбувається агрегація результатів по всьому набору зображень для отримання загальної оцінки. Це значення є ключовим показником ефективності роботи генеративної моделі. На основі цієї агрегованої оцінки формується підсумковий показник ефективності U-генератора.

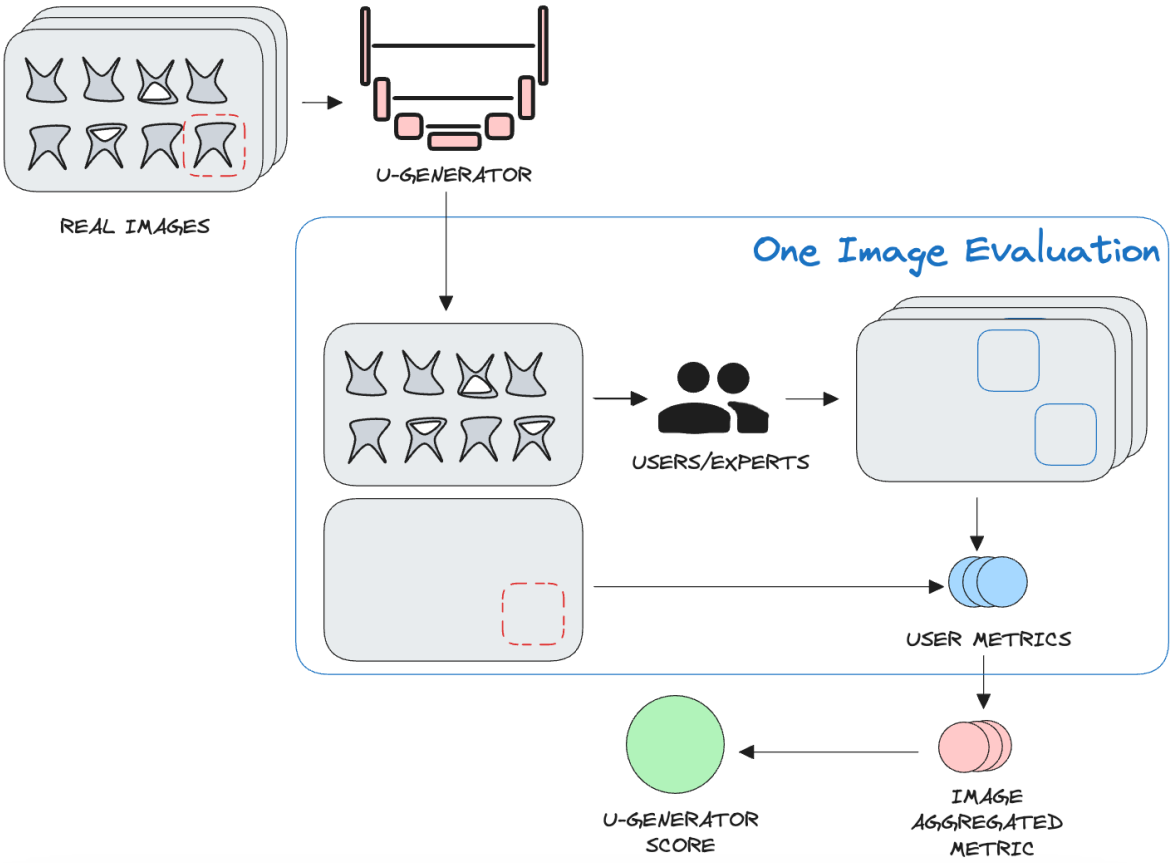


Рис. 3.7 Схема методу оцінки частково згенерованих даних

3.3 Перевірка метрики для оцінювання частково аугментованих даних

На рисунку 3.8 показаний приклад роботи онлайн-застосунку, що буде детальніше описаний у Розділі 4, для перевірки розробленого підходу. Користувачі мають можливість виділяти фрагменти на КТ знімку зубів, які вони вважають трансформованими або модифікованими. Ці виділені області вибираються прямокутниками на знімку. Після завершення роботи з поточним зображенням, користувач може перейти до наступного, натиснувши на кнопку «Next», розташовану внизу інтерфейсу.

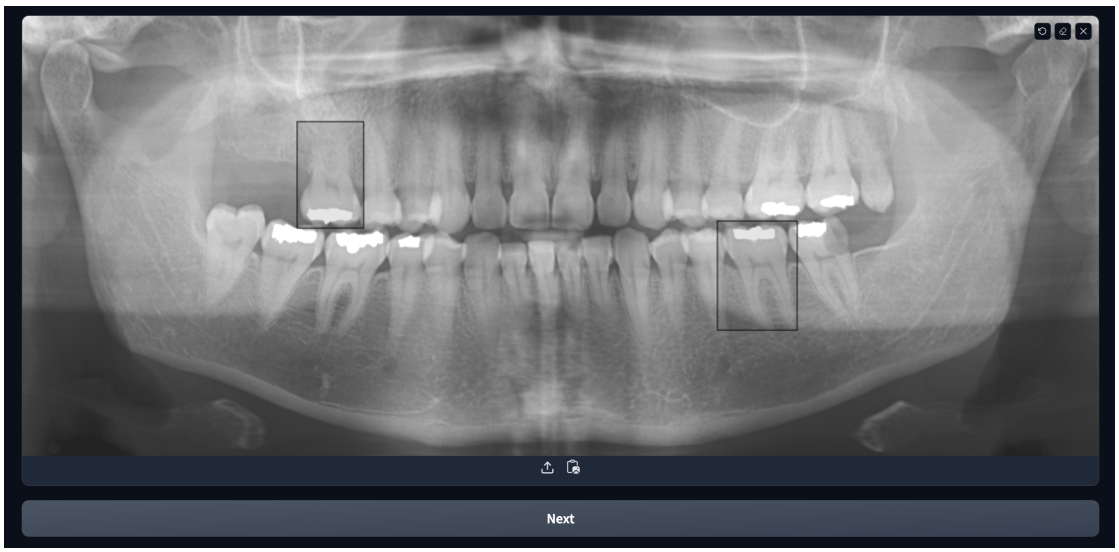


Рис. 3.8 Екранна форма застосунку розробленого методу

Для перевірки ефективності метрики оцінювання частково аугментованих даних було проведено оцінку на наборі даних КТ знімків зубів.

У експерименті взяло участь 30 користувачів, кожному з яких необхідно було опрацювати по 20 зображень трансформованих кожною з трьох зазначених вище моделей. Додатково додано 10 зображень, які не були трансформовані зовсім. На кожному зображенні можуть бути трансформовані від 0 до 2 зубів. Значення влучності, повноти та F1-міри для кожного окремого користувача і моделі наведені у таблиці 3.1. Для кожного окремого зображення метрики наведено у таблиці Додатку Б.

Таблиця 3.1 Результати методу оцінювання на рівні користувач-модель

Користувач	**F1**			**Влучність**			**Повнота**		
	Мод. 1	**Мод. 2**	**Мод. 3**	**Мод. 1**	**Мод. 2**	**Мод. 3**	**Мод. 1**	**Мод. 2**	**Мод. 3**
1	0.923	0.467	0.083	0.933	0.525	0.100	0.950	0.450	0.075
2	0.817	0.120	0.150	0.800	0.117	0.175	0.850	0.150	0.150
3	0.750	0.250	0.217	0.775	0.300	0.225	0.750	0.225	0.225
4	0.753	0.175	0.467	0.767	0.175	0.500	0.750	0.175	0.475
5	0.958	0.550	0.467	0.975	0.600	0.500	0.950	0.525	0.450

Продовження до табл. 3.1

Користувач	**F1**			**Влучність**			**Повнота**		
	Мод. 1	**Мод. 2**	**Мод. 3**	**Мод. 1**	**Мод. 2**	**Мод. 3**	**Мод. 1**	**Мод. 2**	**Мод. 3**
6	0.975	0.233	0.208	0.975	0.300	0.225	0.975	0.200	0.200
7	0.883	0.258	0.033	0.900	0.300	0.050	0.875	0.250	0.025
8	0.325	0.183	0.175	0.325	0.175	0.175	0.325	0.200	0.175
9	0.725	0.275	0.208	0.750	0.325	0.200	0.725	0.250	0.225
10	0.758	0.258	0.308	0.750	0.300	0.350	0.775	0.250	0.300
11	0.782	0.275	0.083	0.808	0.300	0.100	0.775	0.275	0.075
12	0.925	0.333	0.242	0.950	0.350	0.275	0.925	0.325	0.225
13	0.792	0.333	0.183	0.825	0.375	0.200	0.775	0.325	0.175
14	0.833	0.475	0.183	0.850	0.500	0.200	0.825	0.475	0.175
15	0.787	0.170	0.183	0.792	0.192	0.200	0.800	0.175	0.175
16	0.833	0.308	0.200	0.850	0.325	0.250	0.825	0.300	0.175
17	0.707	0.200	0.133	0.708	0.225	0.150	0.725	0.200	0.125
18	0.658	0.445	0.192	0.700	0.467	0.225	0.650	0.450	0.175
19	0.850	0.283	0.175	0.825	0.300	0.200	0.900	0.275	0.175
20	0.767	0.233	0.108	0.775	0.250	0.125	0.775	0.225	0.100
21	0.687	0.158	0.158	0.692	0.175	0.175	0.700	0.150	0.150
22	0.683	0.267	0.175	0.675	0.325	0.200	0.700	0.250	0.175
23	0.683	0.283	0.242	0.725	0.300	0.275	0.675	0.300	0.225
24	0.790	0.242	0.183	0.783	0.250	0.200	0.800	0.250	0.175
25	0.725	0.242	0.150	0.725	0.275	0.150	0.750	0.225	0.150
26	0.837	0.300	0.167	0.817	0.325	0.200	0.875	0.300	0.150
27	0.767	0.250	0.150	0.775	0.300	0.150	0.775	0.225	0.150
28	0.887	0.250	0.275	0.892	0.300	0.300	0.900	0.225	0.275
29	0.642	0.200	0.292	0.650	0.225	0.300	0.650	0.200	0.300
30	0.783	0.217	0.192	0.800	0.250	0.225	0.775	0.200	0.175

На рисунку 3.9 представлено гістограми, які відображають розподіли значень F1-метрики для трьох моделей, що тестувалися на 30 користувачах. Можна зазначити, що користувачі здобували найкращі результати на моделі 1, що є найслабшою. Розподіли метрики для моделі 2 та моделі 3 набагато кращі, розподіл для моделі 3 зміщений трохи лівіше, що говорить про перевагу над моделлю 2. Варто уточнити, що нижче значення F1 вказує на кращу якість моделі, тобто користувачу складніше відрізнити згенеровані частини від реальних.

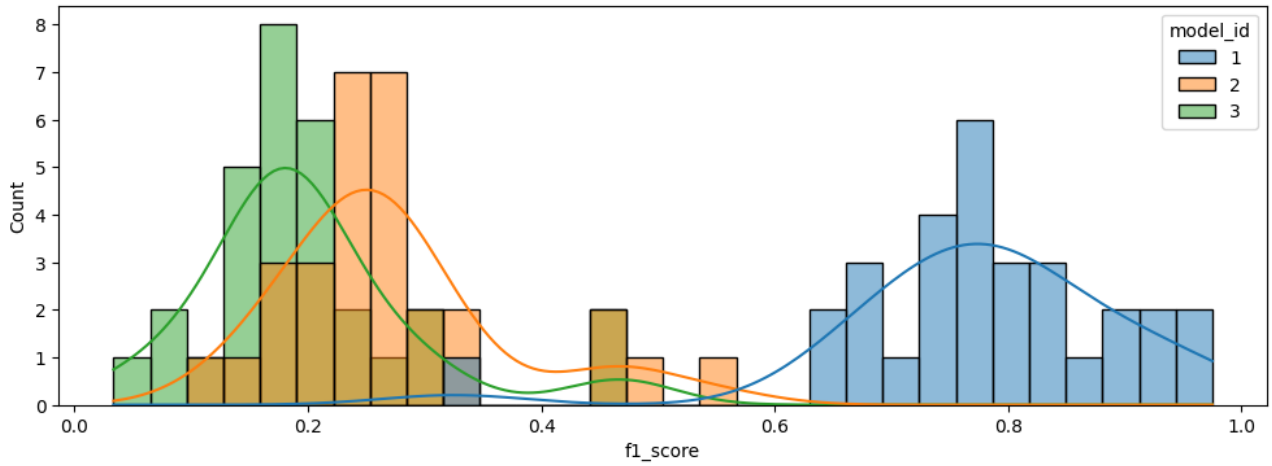


Рис. 3.9 Розподіли F1 оцінок користувачів

У таблиці 3.2 наведено середні значення метрик точності, повноти та F1-метрики для всіх користувачів по кожній з моделей. Результати свідчать про наступне:

1. Модель 1 показала найгірші середні значення за всіма метриками: влучність – 0.785556, повнота – 0.783333, F1 – 0.776167.

2. Модель 2 мала помітно кращі показники, середні значення яких склали: влучність – 0.304167, повнота – 0.267500, F1 – 0.274500.

3. Модель 3 виявилася найкращою серед усіх трьох моделей із середніми значеннями: влучність – 0.220000, повнота – 0.193333, F1 – 0.199444.

Таблиця 3.2 Середні результати по кожній з моделей

Модель	Влучність	Повнота	F1-міра
1	0.785556	0.783333	0.776167
2	0.304167	0.267500	0.274500
3	0.220000	0.193333	0.199444

На рисунках 3.10-3.11 показані всі вибори користувачів для першої та третьої моделі. Центри трансформованих боксів зубів показані зеленими точками.

На рисунку 3.10 зверху видно, що неякісна трансформація зуба верхньої щелепи сильно привернула увагу користувачів і він вибирався дуже багато разів. З нижньої сторони видно, що цей зуб хоч був знайдений рядом користувачів, але вони вибирали і не трансформовані зуби на цій картинці.

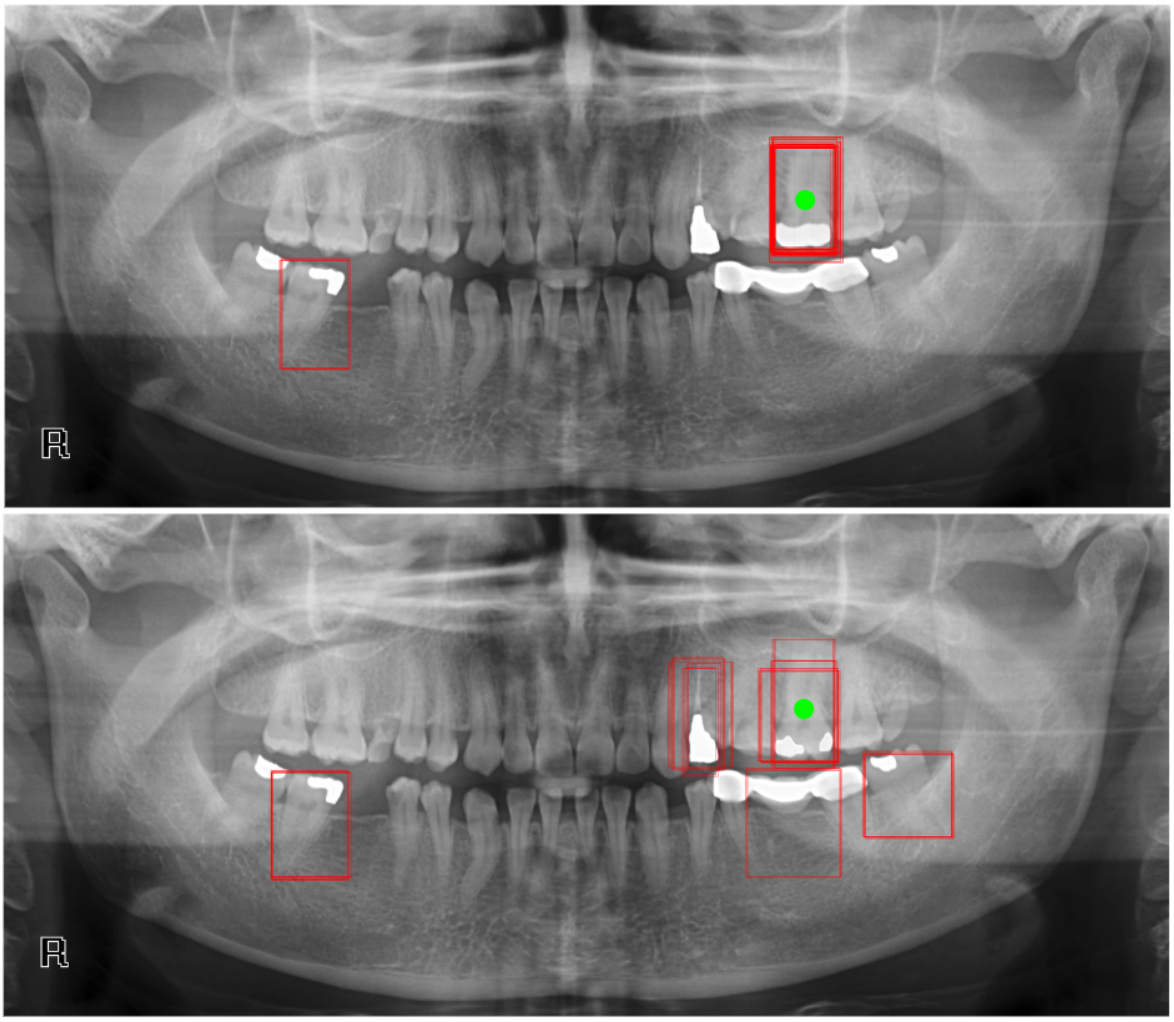


Рис. 3.10 Трансформація 27-го зубу моделлю 1 та 3

Верхня сторона рисунка 3.11 показує, що два підозрілих зуби у верхній щелепі мали сильні ознаки трансформації, за рахунок чого інші дивні пломби нижньої щелепи не вибирались часто. Модель 3 не створила сильних ознак трансформації, завдяки цьому один з трансформованих зубів взагалі не був вибраний, а також було вибрано багато разів дуже підозрілі пломби з нижньої щелепи.

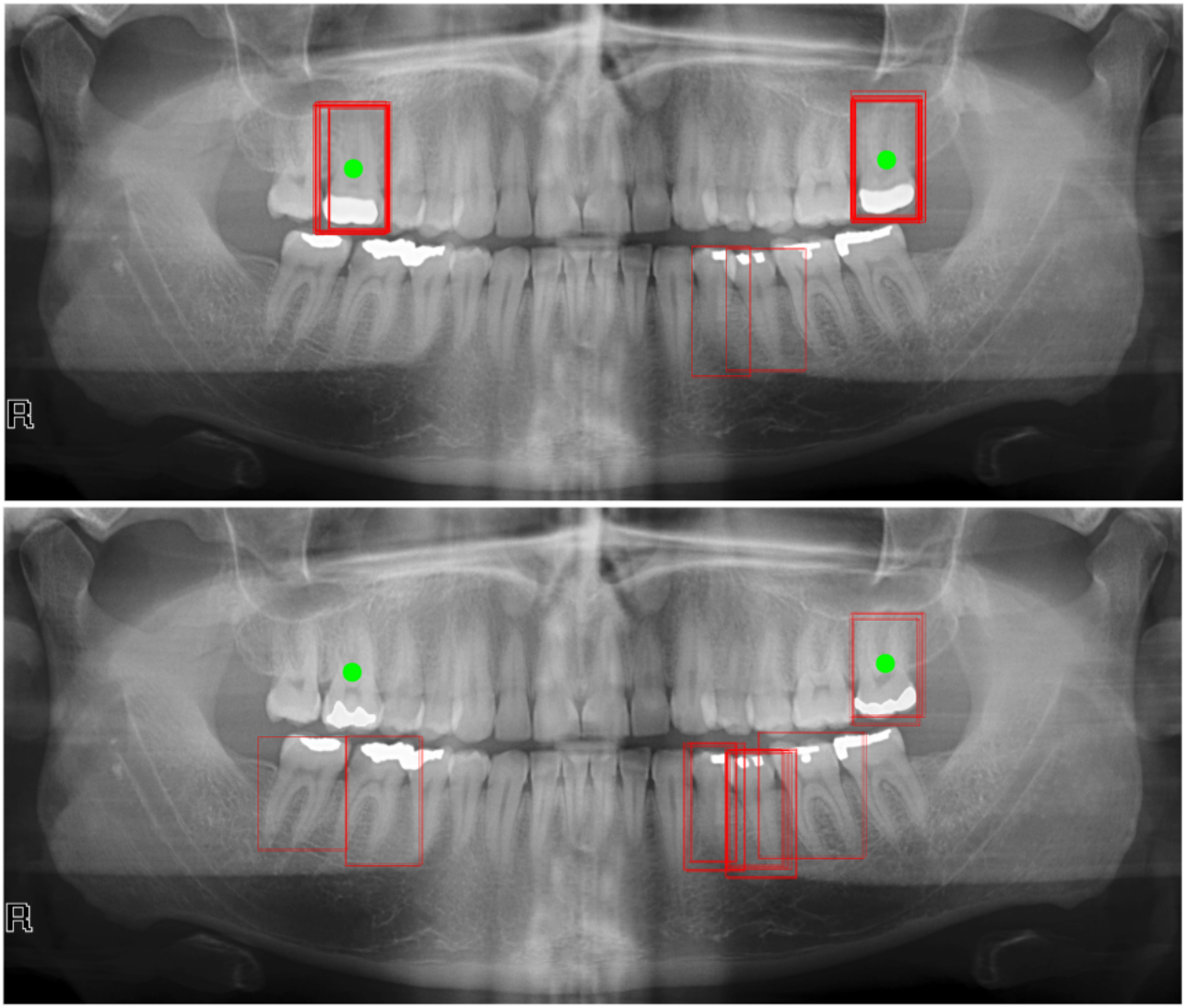


Рис. 3.11 Трансформація 16-го та 27-го зубів моделлю 1 та 3

Рисунок 3.12 демонструє дуже помітний артефакт у апріорі поганої моделі, за рахунок якого більшість користувачів сконцентрували увагу на ньому.

Самовалідована ГЗМ натомість створила гарний екземпляр, і користувачі розсіяли свою увагу по різним нетрансформованим екземплярам.

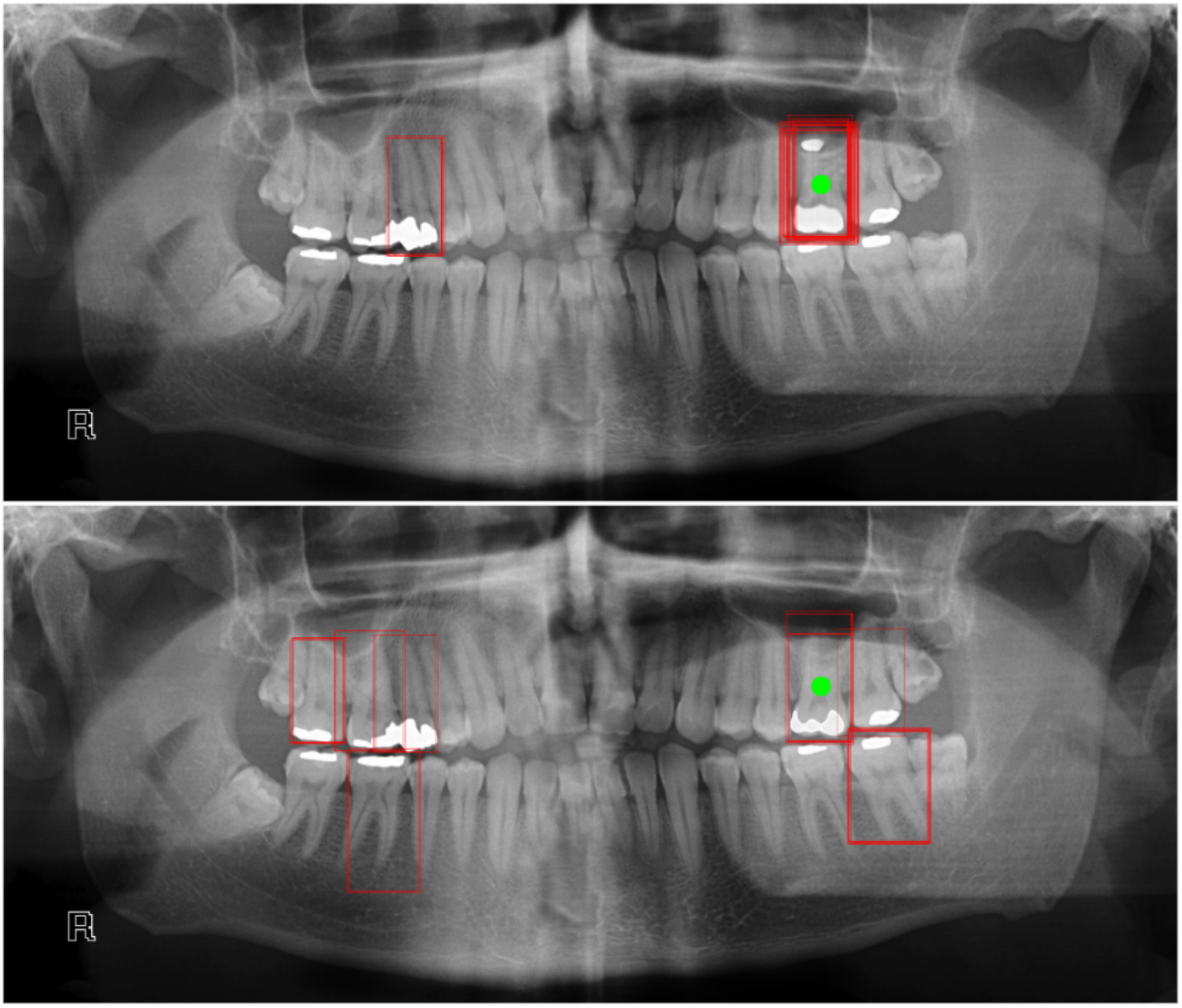


Рис. 3.12 Трансформація 26-го зубу моделлю 1 та 3

3.4 Висновки до розділу

У цьому розділі розглянуто на практиці проблему існуючих методів оцінювання, на прикладі однієї з найвідоміших метрик оцінювання – Fréchet Inception Distance, для використання їх у задачі часткової аугментації/трансформації даних. Показано, що маленькі зміни у зображеннях (навіть якщо вони досить погані) майже не впливають на розподіли даних, та не

ідентифікуються FID через схожість векторів, отриманих від моделі Inception, на якій заснований підхід FID.

Запропоновано алгоритм для оцінки частково згенерованих даних, що включає такі кроки:

- Вибір зображень для оцінювального набору даних та вибір зон для трансформації;
- Трансформація вибраних ділянок генеративною моделлю;
- Оцінка трансформованих зображень експертами (сині бокси);
- Розрахунок метрик для кожного користувача (precision, recall, f1);
- Агрегація результатів серед всіх користувачів;
- Агрегація результатів по всім картинкам.

В результаті технологія віддає як проксі-метрики успішності користувачів (precision, recall, f1 score), так і обернений до цих метрик показник успішності моделі.

Підхід перевірено на наборі даних панорамних знімків зубів для трьох різних моделей: ГЗМ на основі U-генератору, ГЗМ на основі U-генератору з після-обробкою, самовалідованої ГЗМ. До процесу оцінки було залучено 30 користувачів. За результатами цьому експерименту виявлено, що самовалідована ГЗМ має найкращі результати серед розглянутих підходів.

Результати розділу опубліковано в таких роботах автора: [4], [5], [6].

РОЗДІЛ 4
ПРОЕКТУВАННЯ, РЕАЛІЗАЦІЯ ТА ПРИКЛАДНЕ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ САМОВАЛІДОВАНИХ ГЕНЕРАТИВНИХ ЗМАГАЛЬНИХ МЕРЕЖ

4.1 Проектування програмного модулю самовалідованих генеративних змагальних мереж

На рисунку 4.1 зображена контекстна діаграма проектованої інформаційної технології самовалідованих генеративних змагальних мереж, яка створена за допомогою методології IDEF0 [105]. Основний блок діаграми описує процес створення самовалідованої генеративної змагальної мережі. На вхід система приймає набір даних, для якого створюється генеративна модель. На виході очікуються наступні результати:

- самовалідована ГЗМ,
- метрики оцінки генеративної моделі,
- результати аугментування на вхідному наборі даних,
- сегментаційна модель для рішення задачі сегментації повних зображень.

Механізмами виступають як розробник-науковець з даних, так і розмітник даних, як вхідних так і створених у процесі генерування та оцінювання. Керуванням виступають набори гіперпараметрів для навчання як ГЗМ, так і моделі сегментації.

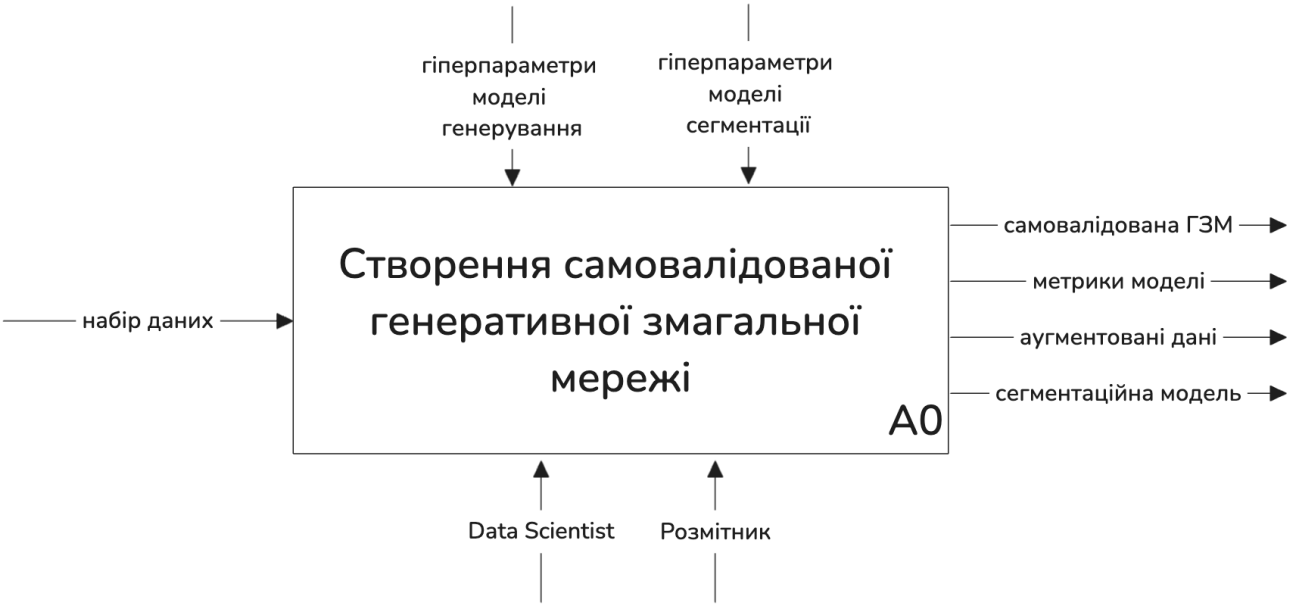


Рис. 4.1 Контекстна діаграма системи

Виділимо наступні основні модулі та дані цієї інформаційної технології (див. рисунок 4.2):

- Модуль тренування самовалідаційного компонента претренованої сегментаційної моделі, приймає на вхід набір даних з розміткою, та створює претреновану сегментаційну модель.
- Модуль тренування самовалідованої ГЗМ, приймає на вхід набір даних з розміткою та претреновану сегментаційну модель створюючи самовалідовану ГЗМ.
- Модуль часткової аугментації зображень, приймає на вхід модель та набір даних, створюючи його аугментовану версію.
- Модуль тренування сегментаційної моделі для вирішення задачі сегментації повних зображень. Приймає на вхід набір даних як реальних даних, так і аугментованих, повертає модель для сегментації даних цього домену.
- Модуль оцінки частково згенерованих даних. Приймає на вхід генеративну модель, повертає метрики оцінки моделі.
- Модуль візуалізації результатів. Використовується для отримання різноманітних графіків та метрик.

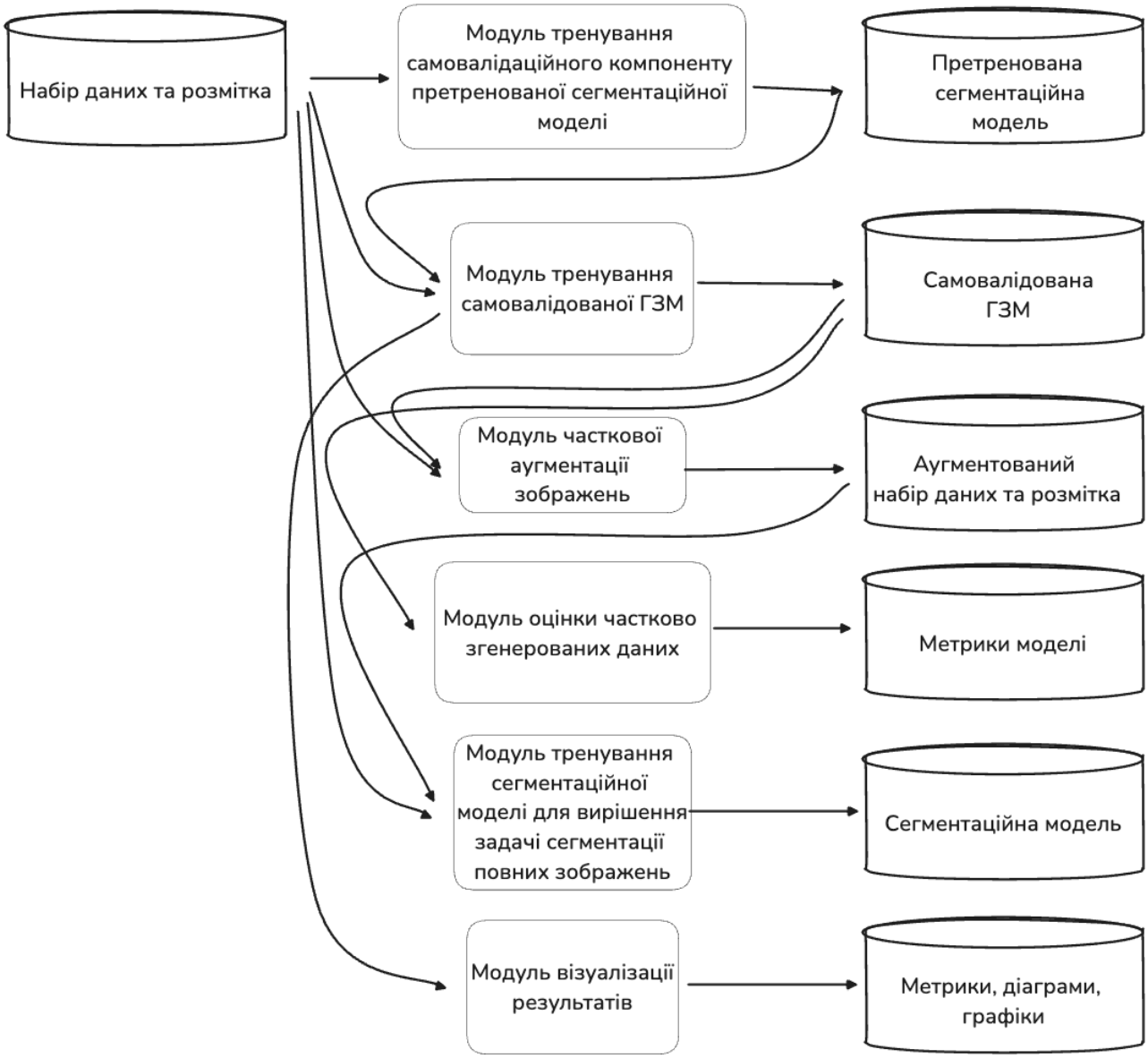


Рис. 4.2 Схема інформаційної технології

Далі детально опишемо алгоритм роботи кожного модулю.

Модуль тренування самовалідаційного компонент претренованої сегментаційної моделі.

Мета цього модулю створити претреновану сегментаційну модель, яка зможе робити сегментацію результату генерування ГЗМ. Для цього на вхід модулю поступає підготовлений розмічений набір даних – зображення та сегментаційна маска. Діаграма прецедентів цього модулю показана на рисунку 4.3. Серед основних дій спеціаліста з даних доступно налаштування та запуск процесу тренування моделі.

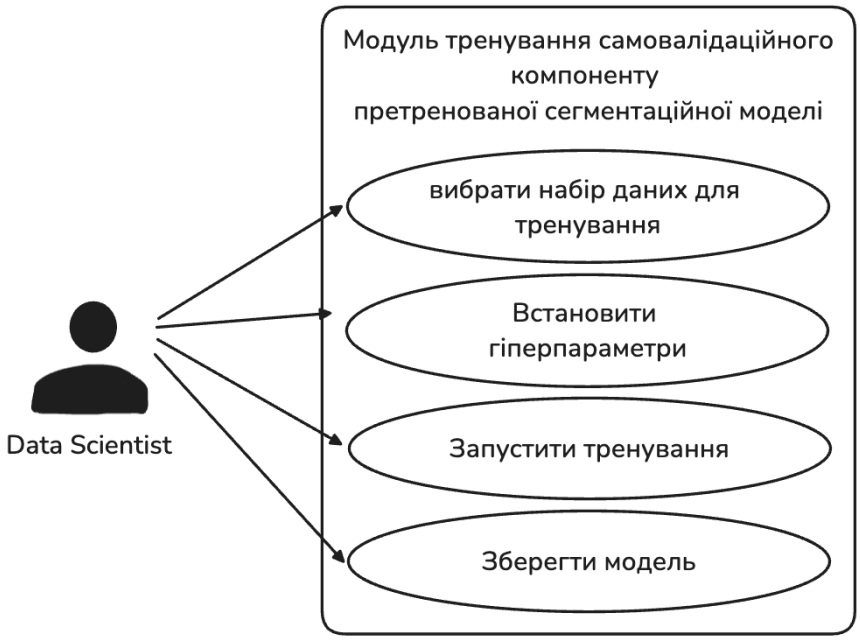


Рис. 4.3 Діаграма використання модуля тренування самовалідаційного компонента претренованої сегментаційної моделі

Модуль має базовий набір гіперпараметрів, які спеціаліст з даних має змогу змінювати, серед основних параметрів:

- розмір зображення: 128*128;
- U-Net модель на основі resnet34;
- швидкість навчання: 0.001;
- кількість епох: 50;
- функція втрат: функція Жаккара;
- метрика вибору моделі: коефіцієнт Жаккара;
- розмір пакету: 32;
- зупинка після 5 епох без покращення;
- оптимізатор: Adam.

Користувач може запустити тренування з різними наборами гіперпараметрами та після цього вибрати результати запуску (модель, параметри) для збереження, що мають кращі результати.

Модуль тренування самовалідованої ГЗМ.

Модуль тренування самовалідованої ГЗМ є основним модулем, який відповідає за створення самої моделі самовалідованої генеративної змагальної мережі. Діаграма прецедентів цього модуля показана на рисунку 4.4, і її основні дії включають етапи тренування моделей глибокого навчання.

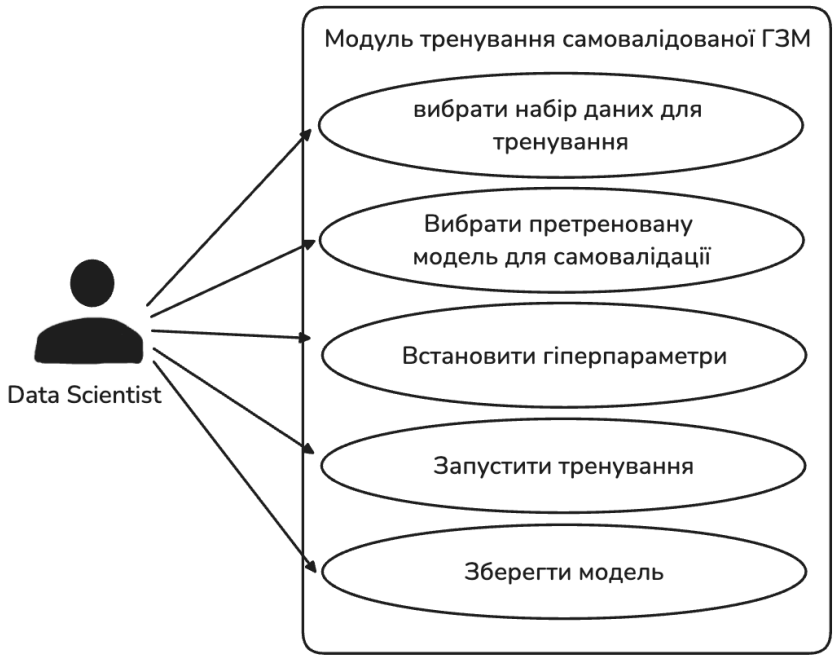


Рис. 4.4 Діаграма використання модуля тренування самовалідованої ГЗМ

Користувачу необхідно спочатку вибрати набір даних як цільового класу, так і нульового. Після цього вибрати одну з моделей сегментаційної самовалідації, створену в минулому модулі. Далі у користувача є можливість вибрати наступні гіперпараметри ГЗМ:

- розмір зображення: 128 на 128 пікселів;
- модель генератору: U-генератор;
- модель критику: згорткова нейронна мережа;
- швидкість навчання генератору: 0.001;
- швидкість навчання критику: 0.002;
- кількість епох: 150;
- випадковий стан: 42;
- кількість повторів циклу навчання генератору: 2;

- кількість повторів циклу навчання критика: 1;
- кількість каналів входу в генератор: 2 (1 чорно-біла картинка, 1 випадковий шум);
- кількість каналів входу в критик: 2 (1 чорно-біла картинка, 1 маска сегментації);
- розмір пакету: 16;
- функція втрат різниці між згенерованим і вхідним зображенням: rmse;
- початковий коефіцієнт функції втрат різниці: 10, з лінійним зменшенням до 1 протягом 150 епох.

Після завершення тренування у користувача є можливість зберегти модель для подальшого використання у модулі часткової аугментації зображень та проаналізувати тренувальні метрики і створені зображення у модулі візуалізації результатів.

Модуль візуалізації результатів генерування.

Цей модуль відповідає за повний процес візуалізації тренування та генерування ГЗМ (див. рис. 4.5). Він дозволяє візуалізувати наступне:

- створені під час тренування зображення;
- динаміку змін результатів генерування під час тренування;
- метрики навчання генератора, критика, самовалідаційних компонентів в залежності від номеру епохи;
- зміни динамічних гіперпараметрів модулю.

Окрім цього, модуль дозволяє створити нові агументації, на основі відкладених зображеннях нульового класу, використовуючи як звичайний режим генерування, так і прийом з обрізкою випадкового шуму.

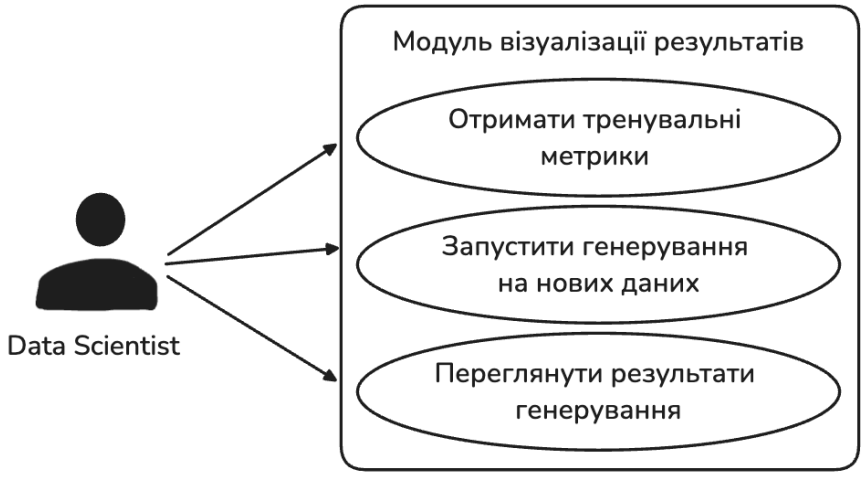


Рис. 4.5 Діаграма використання модуля візуалізації результатів

Модуль часткової аугментації зображень реалізує технологію часткового трансформування фонових зон на зображенні. Він дозволяє збагачувати вхідні зображення за рахунок додавання на них нових сегментаційних об’єктів, при цьому автоматично створюючи сегментаційні маски для них. Перелік можливих дій показано на діаграмі прецедентів на рисунку 4.6.

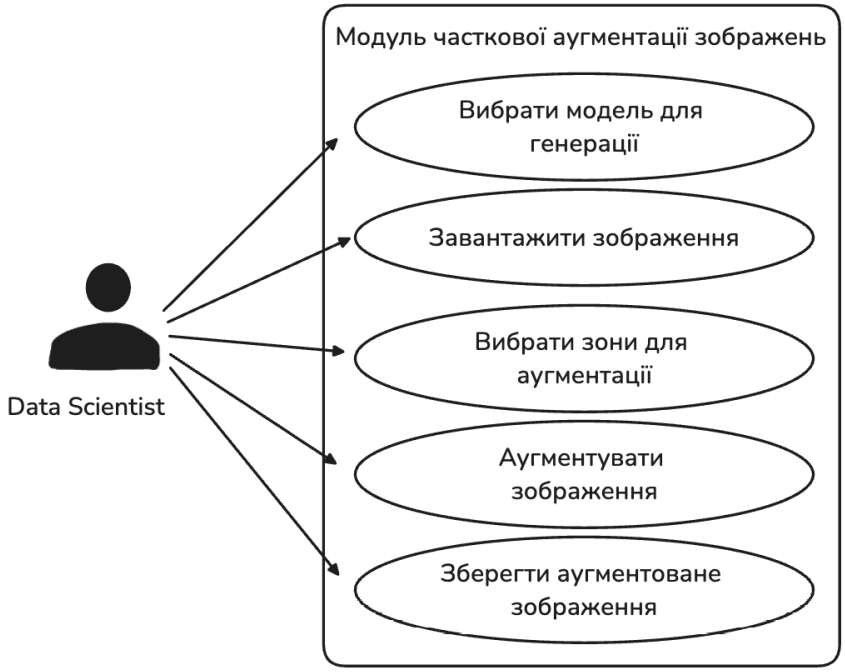


Рис. 4.6 Діаграма прецедентів модуля часткової аугментації зображень

Алгоритм роботи із модулем часткової аугментації зображень складається з наступних кроків:

- користувач обирає з переліку модель генерування;
- завантажує необхідне зображення для аугментації;
- вибирає потрібну кількість фрагментів для аугментації завдяки ручному вибору зони на зображенні;
- аугментувати зображення, якщо зображення не є потрібної якості, або потрібно створити декілька різних екземплярів для вибраних зон, або перевибрати інші зони для трансформації, то цей крок можна повторювати скільки необхідно раз;
- користувач може зберегти обрані зображення на свій пристрій для подальшого використання.

Модуль тренування сегментаційної моделі для задачі сегментації повних зображень.

Після створення набору даних з трансформаціями, його можна використати для тренування моделі, що здатна сегментувати повні зображення. Діаграма використання цього модулю показана на рисунку 4.7. Суть його дуже схожа на модуль тренування самовалідаційного компонента претренованої сегментаційної моделі. Використовуються ті самі моделі та схожі гіперпараметри, але основною відмінністю є використання повнорозмірних зображень, а не обрізаних фрагментів нульового та цільового класів.

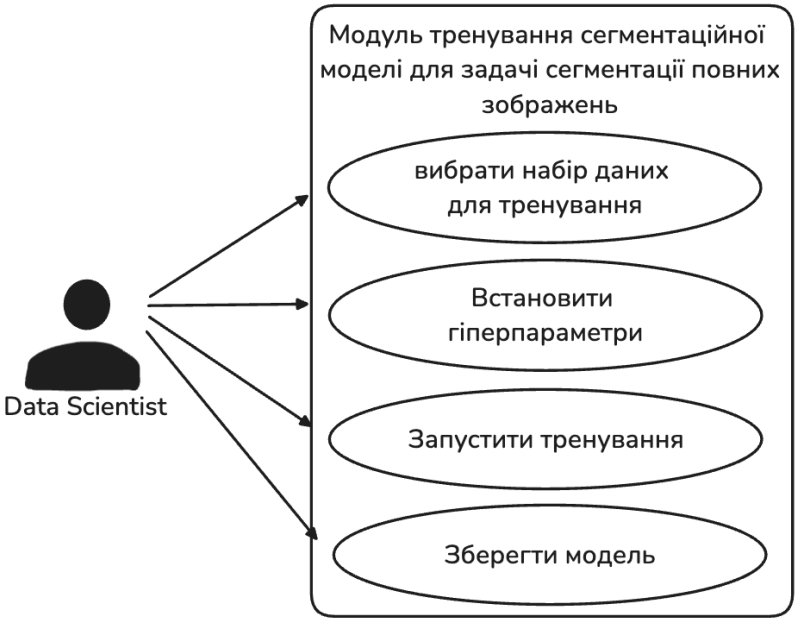


Рис. 4.7 Діаграма використання модуля тренування сегментаційної моделі для вирішення задачі сегментації повних зображень

Наведемо список гіперпараметрів та їх значень за замовчуванням для цього модулю:

- розмір зображення: 512*512;
- U-Net модель на основі resnet34;
- швидкість навчання: 0.01;
- кількість епох: 50;
- функція втрат: функція Жаккара;
- метрика вибору моделі: коефіцієнт Жаккара;
- розмір пакету: 64;
- зупинка після 10 епох без покращення;
- оптимізатор: Adam.

Модуль оцінки частково згенерованих даних.

Цей модуль реалізує метод для оцінки частково згенерованих даних, що описано у Розділі 3. В цьому модулі бере участь два актори: спеціаліст з даних та розмітник. Задачею першого є створення наборів даних для оцінки та отримання метрик для цього набору. Задачею розмітника є аналіз зображень на артефакти:

- Користувачу пропонується продивитись зображення, на якому в деяких місцях присутні згенеровані фрагменти. Для цього він має вибрати необхідний набір даних зі списку доступних наборів для оцінки.
- Після цього користувачу необхідно вирішити, які області він вважає згенерованими.
- Користувач виділяє в графічному інтерфейсі ці місця та завершує роботу з цим зображенням.
- Після проходження заданої кількості зображень користувачу виводить на екран його точність (метрики precision, recall, f1 score).

Спеціалісту з даних доступний функціонал завантаження нових наборів даних, зображень до них, а також правильних участків трансформації, щоб система могла оцінити перетини з боксами користувачів. Також можна отримати всі оцінки як для окремих користувачів, так і для конкретних зображень, або різні агрегації.

Діаграма використання цього модулю наведена на рисунку 4.8.

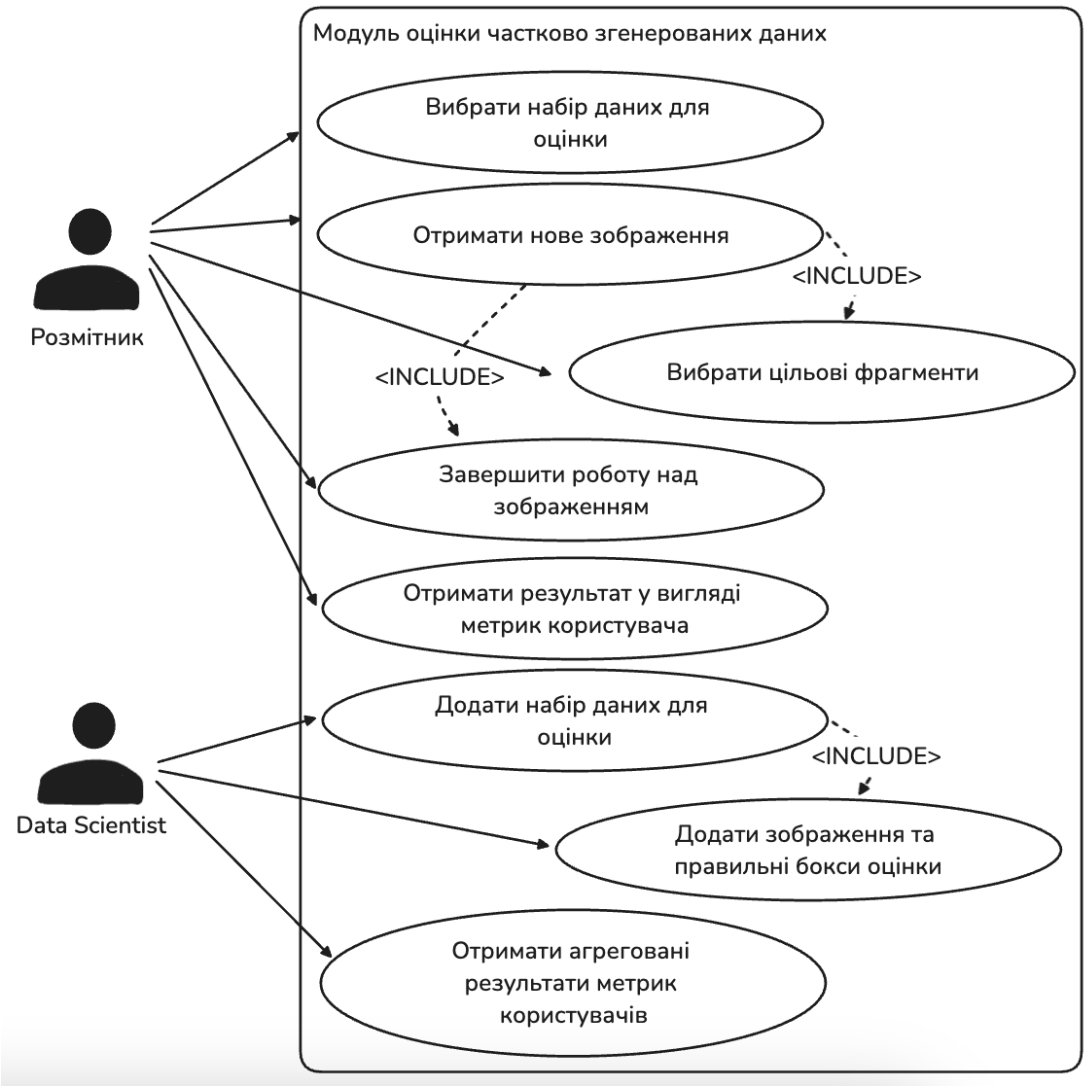


Рис. 4.8 Діаграма прецедентів модуля оцінки частково згенерованих даних

Враховуючі наведену вище інформацію, створимо UML-діаграму послідовності роботи модулів інформаційної технології, яка пропонується (див. рис. 4.9).

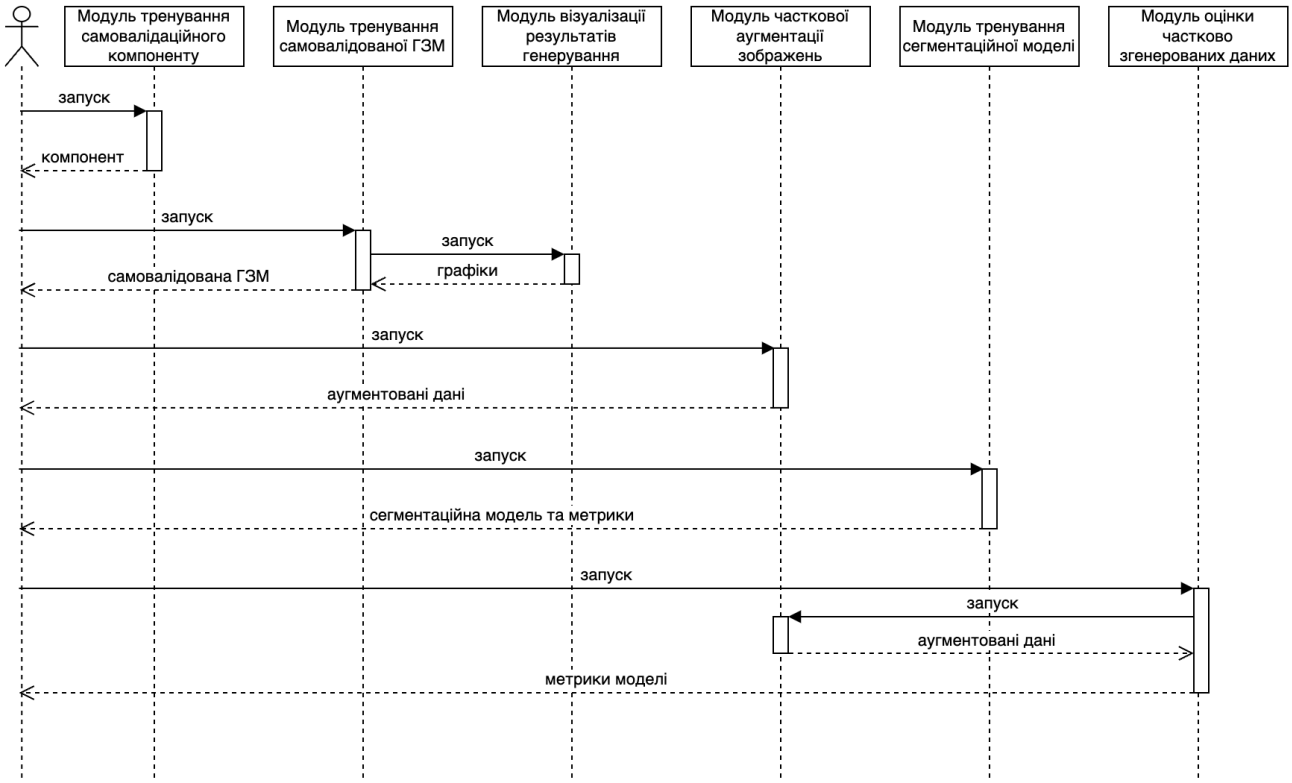


Рис. 4.9 UML-діаграма послідовності роботи модулів інформаційної технології

Ця діаграма демонструє наступну хронологію процесів:

- спочатку запускається модуль тренування самовалідаційного компонента, в ньому відбувається тренування компонента для використання в наступних модулях;
- далі відбувається запуск модуля тренування самовалідованої ГЗМ, в якому спочатку відбувається тренування, описане в розділах 2.4-2.5, а далі виклик модулю візуалізації результатів, що повертає графіки та артефакти навчання;
- отримавши на попередньому кроці самовалідовану ГЗМ, виконується модуль часткової аугментації зображень, що дозволяє створювати нові приклади даних;
- з використанням набору даних та нових прикладів запускається модуль тренування сегментаційної моделі, в результаті роботи якого отримується сегментаційна модель та її метрики;
- останнім кроком є запуск модулю оцінки частково згенерованих даних, у якому відбувається залучення модулю часткової аугментації зображень,

для створення прикладів для оцінки. Після цього відбувається сама оцінка, а метрики моделі повертаються користувачу.

На рисунку 4.10 показано діаграму діяльності створеної інформаційної технології зі сторони користувача. к можна побачити, перший модуль, представлений блоком «тренування самовалідованих компонентів», супроводжується додатковою дією з боку користувача, а саме вибором гіперпараметрів моделі. У випадку, якщо обрані гіперпараметри не дозволяють створити компонент необхідної якості, користувач має повернутись назад до кроку їх вибору. Таку саме структуру має блок «тренування Самовалідованої ГЗМ». У випадку недостатньої якості користувач може перевибрати налаштування і почати цей крок спочатку.

Після цього йде розділення процесу на дві незалежні гілки:

- тренування сегментаційної моделі;
- оцінка згенерованих даних.

Перед тим, як виконувати тренування сегментаційнох моделі, користувач може створити необхідну кількість згенерованих даних для покращення якості самої моделі.

У блоці оцінки користувачу потрібно узгодити кінцеву модель та обрати набір даних та бокси для трансформації, після цього відбувається аугментація цих боксів. Коли дані готові, то починається сам процес оцінки даних.

Якщо будь-яка з двох паралельних гілок не влаштовує користувача, то весь процес починається з моменту розголудження на тренування сегментаціїної моделі та оцінку згенерованих даних.

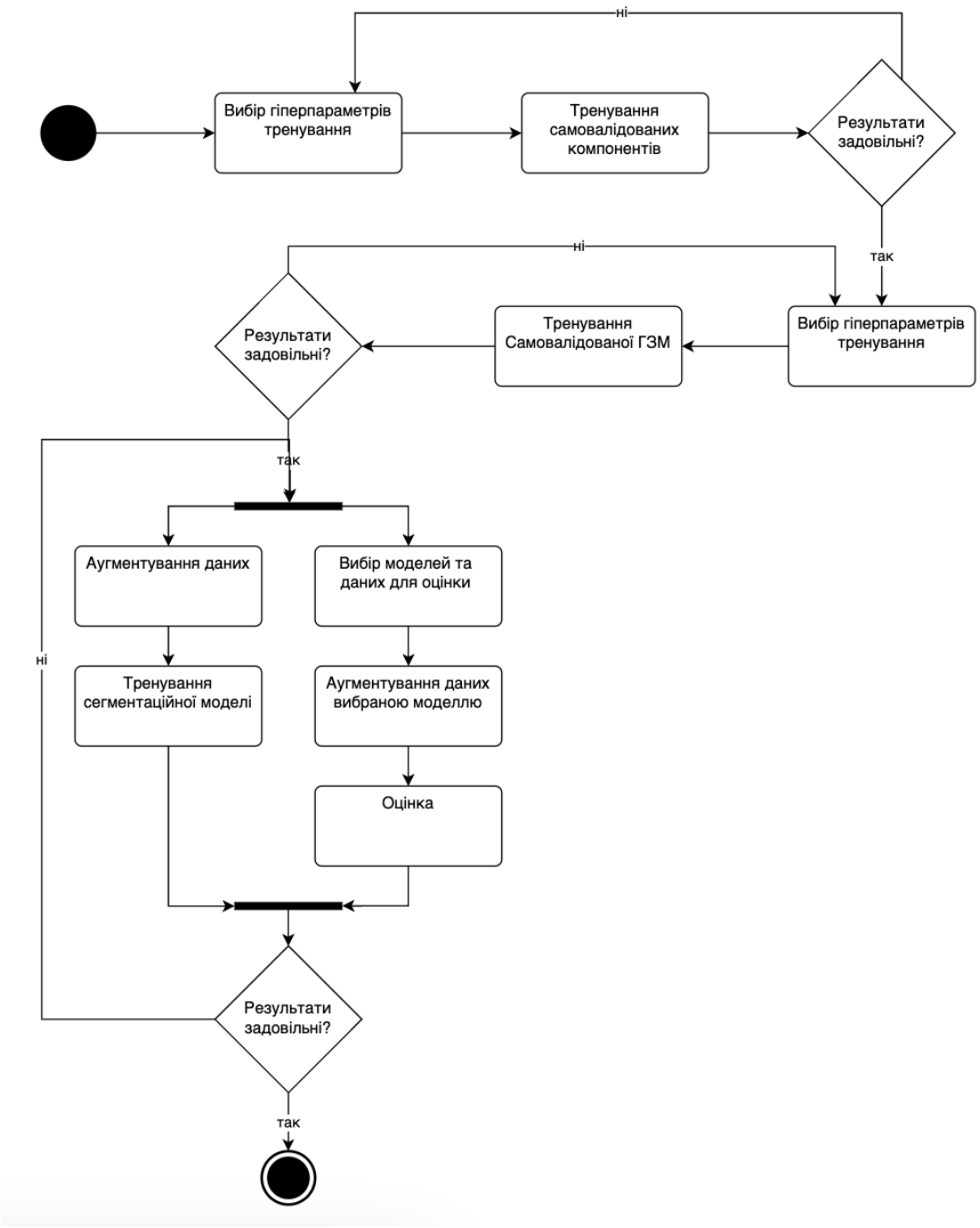


Рис. 4.10 UML-діаграма діяльності інформаційної технології

4.2 Реалізація комплексу програм для застосування інформаційної технології самовалідованих генеративних змагальних мереж

Для частини модулів, а саме:

- модуль тренування самовалідаційного компонента претренованої сегментаційної моделі;

- модуль тренування самовалідованої генеративно-змагальної мережі;
- модуль візуалізації результатів генерації;
- модуль тренування сегментаційної моделі для вирішення задачі сегментації повних зображень.

Реалізація відбувається у вигляді програм-ноутбуків (Jupyter notebooks [106]) на платформі Kaggle [107].

Ця платформа надає численні переваги для її використання:

- Потужності для тренування великих моделей: В роботі використовується відеокарта Nvidia Tesla P100 [108], що забезпечує достатню обчислювальну потужність для ефективного тренування складних моделей.
- Зберігання програм та їх версій: Kaggle дозволяє зручно зберігати різні версії програм та вибирати необхідну, що спрощує управління проектом та його версіями.
- Зручний доступ до наборів даних: Платформа надає можливість легко отримувати доступ до різноманітних наборів даних, а також створювати та використовувати власні набори даних як приватні, так і публічні.
- Зручні інструменти написання коду: Інтерфейс Kaggle забезпечує комфортне середовище для розробки та тестування коду і включає два основних типи програм: скріпти – це файли, які виконують увесь код послідовно та Jupyter Notebooks (блокноти), що складаються з послідовності комірок, де кожна клітинка відформатована у Markdown (для написання тексту) або мовою програмування на Python чи R.

Незважаючи на те, що реалізація цих модулів була проведена в середовищі Kaggle, вони легко можуть бути портовані в інші середовища, такі як:

- AWS SageMaker [109]: Хмарний сервіс від Amazon, який надає інструменти для тренування та розгортання моделей машинного навчання. Завдяки інтеграції з іншими сервісами AWS, такими як S3 для зберігання даних і Lambda для обробки запитів, SageMaker забезпечує масштабованість та зручність у роботі з великими обсягами даних. Крім того, він підтримує автоматичне налаштування параметрів моделі (hyperparameter tuning) та A/B тестування

моделей, що робить його зручним інструментом для створення та впровадження складних систем на базі машинного навчання.

- Google Colab [110]: Безкоштовний онлайн сервіс від Google, який дозволяє виконувати Python код у браузері з використанням GPU. Це ідеальний інструмент для швидкого прототипування та експериментів з обмеженими обчислювальними ресурсами. Окрім GPU, Colab пропонує інтеграцію з Google Drive для зберігання даних, а також можливість легко поширювати ноутбуки.

- Jupyter Notebook або JupyterLab: Локальні інструменти для інтерактивної роботи з даними та написання коду на Python. Jupyter забезпечує гнучке середовище для роботи з багатьма мовами програмування, не тільки Python. Ці інструменти широко використовуються науковцями та аналітиками для виконання обчислень у режимі реального часу, візуалізації даних та документування результатів. JupyterLab є розширеною версією Jupyter Notebook, з додатковими можливостями для керування файлами, підключення до віддалених обчислювальних ресурсів та інтеграції з іншими інструментами.

- Microsoft Azure Notebooks [111]: Хмарна платформа від Microsoft для створення та виконання Jupyter ноутбуків. Azure Notebooks дозволяє інтегруватися з сервісами Azure, такими як Azure ML, для автоматизації робочих процесів, тренування моделей у великих масштабах і розгортання рішень. Перевагою є глибока інтеграція з Microsoft Office 365 і можливість використання корпоративних рішень для аналітики та звітності.

- PyCharm [112]: Інтегроване середовище розробки (IDE) для Python від JetBrains, яке підтримує роботу з Jupyter ноутбуками. PyCharm пропонує розширені можливості для налагодження коду, аналізу продуктивності та інтеграції з системами контролю версій (наприклад, Git), що робить його гарним застосунком для роботи над складними проєктами. Завдяки інтеграції з Docker та іншими контейнеризаційними платформами, PyCharm спрощує процес розробки та тестування машинного навчання.

Ці середовища забезпечують гнучкість та зручність для розробки, тренування, тестування та розгортання моделей машинного та глибокого навчання, що робить їх придатними для різних потреб та умов роботи.

Модуль часткової аугментації зображень та модуль оцінки частково згенерованих даних було реалізовано використовуючи Visual Studio Code [113], і реалізовані вони у вигляді застосунків з веб-інтерфейсом для зручного користування іншими людьми. Visual Studio Code (VS Code) – це потужний і гнучкий редактор від компанії Microsoft, що став популярним завдяки легкості, розширюваності та підтримці багатьох мов програмування, включаючи Python, JavaScript, C++, Java та інші. В контексті роботи з машинним навчанням, даними та Jupyter ноутбуками, VS Code пропонує низку переваг, які варто розглянути детальніше. Деякі переваги Visual Studio Code:

Легкість і продуктивність. VS Code є інструментом, що швидко завантажується та споживає мало ресурсів у порівнянні з більш важкими IDE, такими як PyCharm. Це робить його ідеальним вибором для розробників, які працюють на машинах з обмеженими ресурсами або ж намагаються швидко створювати прототипи без зайвого навантаження на систему.

Підтримка розширень. Однією з ключових особливостей VS Code є встановлення розширень, які додають новий функціонал. Наприклад: робота з Jupyter Notebooks, жива демонстрація екрану іншим людям, робота з Docker.

Інтеграція з Git та системами контролю версій. VS Code надає зручну інтеграцію з Git, що дозволяє керувати змінами в проєктах, відстежувати історію змін, створювати та об'єднувати гілки. Інтерфейс VS Code дозволяє робити commit, push, pull та інші операції безпосередньо з редактора, що значно спрощує роботу над проєктами, особливо коли йдеться про командну розробку.

Для написання всього коду було обрано мову програмування Python. Вибір цієї мови аргументовано наступними характеристиками:

- Легкість у вивченні та використанні: Завдяки зрозумілому синтаксису, Python є ідеальним вибором для початківців, але при цьому залишається потужним інструментом для досвідчених розробників.

- Інтерпретована мова: Python є інтерпретованою мовою, де код виконується рядок за рядком, без потреби компіляції.
- Динамічна типізація: Типи змінних визначаються коли код виконується, що дозволяє писати більш гнучкий і швидкий код.
- Багатоплатформність: Python можна використовувати у різних операційних системах, таких як Windows, macOS, Linux та інші.
- Велика стандартна бібліотека: Python має багату стандартну бібліотеку, яка охоплює різні аспекти програмування, від роботи з файлами і мережами до розробки графічних інтерфейсів користувача і обробки даних.
- Можливості для наукових обчислень і машинного навчання: З використанням бібліотек, таких як NumPy, SciPy, TensorFlow, Keras і PyTorch, Python є популярним вибором для наукових досліджень, аналізу даних, машинного навчання та штучного інтелекту.
- Підтримка ООП: Python підтримує ООП, що дозволяє створювати складні програми з використанням класів і об'єктів.
- Велика спільнота та підтримка: Python має одну з найбільших спільнот дослідників у світі, що забезпечує доступ до великої кількості ресурсів, бібліотек і фреймворків.

Для модулів тренування моделей були використанні наступні основні бібліотеки:

- NumPy [114] – це бібліотека для роботи з великими масивами та матрицями чисел, яка включає в себе велику кількість математичних функцій для виконання різноманітних обчислень. Вона є основою для багатьох інших наукових бібліотек в Python, надаючи потужний інструментарій для чисельних обчислень.
- Pandas [115] – це бібліотека для обробки та аналізу даних, яка надає високо рівневі структури даних, такі як DataFrame, для зручної маніпуляції з табличними даними. Вона широко використовується для завантаження, очищення, фільтрації, агрегації та візуалізації даних.
- Matplotlib [116] – це бібліотека для створення якісних візуалізацій, що можна налаштувати у будь який спосіб. Вона дозволяє створювати різноманітні

графіки та діаграми, та забезпечує гнучкість у налаштуванні їх зовнішнього вигляду.

- Seaborn [117] – це бібліотека для візуалізації даних, побудована на основі Matplotlib, яка спрощує створення складних статистичних графіків. Вона забезпечує високо рівневий інтерфейс для малювання привабливих та інформативних візуалізацій.

- PyTorch [118] – це бібліотека для проєктування моделей глибокого навчання, яка забезпечує гнучке та динамічне обчислення градієнтів. Вона широко використовується для розробки та навчання нейронних мереж, завдяки простому та інтуїтивному інтерфейсу.

- Scikit-learn [119] – це бібліотека моделей та метрик машинного навчання, яка надає простий та ефективний інструментарій для побудови та оцінки моделей. Вона включає в себе алгоритми для класифікації, регресії, кластеризації та зниження розмірності.

- OpenCV [120] – це бібліотека для обробки зображень та комп'ютерного зору, яка містить численні алгоритми для обробки, аналізу та розпізнавання зображень та відео. Вона широко використовується у розробці додатків для обробки зображень.

- Segmentation Models PyTorch [121] – це бібліотека для сегментації зображень, побудована на PyTorch, яка містить реалізації популярних архітектур нейронних мереж для завдань сегментації. Вона забезпечує зручний інтерфейс для тренування та оцінки моделей сегментації.

Всі модулі мають прозору ООП структуру і можуть легко бути адаптовані до нових вимог чи наборів даних. Розглянемо основні сутності розрозблених модулів.

Модуль тренування самовалідаційного компонента претренованої сегментаційної моделі та модуль тренування сегментаційної моделі для вирішення задачі сегментації повних зображень складаються з наступних класів:

DataRetriever наслідує базовий клас torch.utils.data.Dataset. Цей клас відповідає за завантаження та передобробку зображень та масок та видачу їх за запиту. Реалізує основні методи, що показані на рисунку 4.11:

- __init__ – ініціалізує початкові компоненти та викликає метод первинного завантаження даних;
- __len__ – повертає кількість зразків у наборі даних);
- __getitem__ – повертає зображення та маску за заданим індексом;
- load – завантажує дані у оперативну пам’ять.

```python
class DataRetriever(torch.utils.data.Dataset):
    def __init__(self, transforms, ids):
        self.transforms = transforms
        self.ids = ids
        self.load()

    def __len__(self) -> int:
        return len(self.images)

    def __getitem__(self, index):
        image = self.images[index]
        mask = self.masks[index]
        tr = self.transforms(image=image, mask=mask)
        image = tr["image"]
        mask = tr["mask"]

        return {"X": image, "y": mask.unsqueeze(0)}

    def load(self):
        self.images = []
        self.masks = []
        for image_name in self.ids:
            img = cv2.imread(image_name)
            img = img.astype(float)

            mask = cv2.imread(image_name.replace("images", "masks"))
            mask = cv2.cvtColor(mask, cv2.COLOR_BGR2RGB).astype(float)[:, :, 0] / 255

            self.images.append(img)
            self.masks.append(mask)
```

Рис. 4.11 Реалізація класу, що завантажує дані

Клас SegmentationModel, що бере функціональність від torch.nn.Module, потрібен для реалізації структури та функціоналу основної сегментаційної моделі, складається з таких методів:

- __init__– ініціалізація моделі, приймає у якості параметрів кількість класів (num_classes), архітектуру енкодеру (encoder_name) та девайс (device) на якому буде знаходитись модель. Викликає допоміжний метод build_model;
- build_model – використовує бібліотеку Segmentation Models PyTorch для завантаження претренованої моделі (рядок 9 на рисунку 4.12);
- forward – виклик прогнозу моделі.

```
1  class SegmentationModel(nn.Module):
2      def __init__(self, num_classes, encoder_name="resnet34", device='cuda'):
3          self.num_classes = num_classes
4          self.encoder_name = encoder_name
5          self.device = device
6          self.model = self.build_model()
7
8      def build_model(self):
9          model = smp.Unet(encoder_name=self.encoder_name, classes=self.num_classes)
10         model.to(self.device)
11         return model
12
13     def forward(self, batch):
14         return self.model(batch)
```

Рис. 4.12 Реалізація класу моделі сегментації

Клас Train, що відповідає за процес тренування моделі та збирає інші компоненти разом, складається з наступних функцій:

- __init__ – визначення моделі, метрик, оптимізатору.
- fit – запуск повного тренувального циклу на задану кількість епох.
- train – запуск однієї епохи тренування для всіх даних. На рисунку 4.13 показано реалізацію цього методу.
- validate – запуск однієї епохи валідації, в функції відбувається підрахунок і повернення функції втрат та метрики коефіцієнта Жакарра.

- save – збереження моделі після завершення всього тренування.

```python
def train(loader) -> dict:
    self.model.train()
    total_loss = 0.0
    correct_predictions = 0
    total_samples = 0
    jaccard_scores = []

    for loader_data in tqdm(loader, desc="Training", disable=not verbose):
        batch_images = loader_data["X"].to(self.device)
        batch_labels = loader_data["y"].to(self.device)

        self.optimizer.zero_grad()

        outputs = self.model(batch_images)
        loss = self.criterion(outputs, batch_labels.squeeze(0))

        loss.backward()
        self.optimizer.step()

        total_loss += loss.item()
        _, predicted = torch.max(outputs, 1)
        correct_predictions += (predicted == batch_labels.squeeze(0)).sum().item()
        total_samples += batch_labels.size(0)

        jaccard_batch = self.metric_fn(outputs.sigmoid(), batch_labels)
        jaccard_scores.append(jaccard_batch.item())

    epoch_loss = total_loss / len(loader)
    average_jaccard = sum(jaccard_scores) / len(jaccard_scores)

    current_lr = self.optimizer.param_groups[0]['lr']

    return {
        "epoch_loss": epoch_loss,
        "lr": current_lr,
        "average_jaccard": average_jaccard,
    }
```

Рис. 4.13 Реалізація методу train класу тренування

Модуль тренування самовалідованої ГЗМ складається з наступних класів:

DataRetriever наслідує torch.utils.data.Dataset. По своїй суті дуже схожий з класом, що отримував дані у сегментаційних моделей. Так само реалізує методи

__init__, __len__ (повертає кількість зразків у наборі даних), __getitem__ (повертає зразок даних та його мітку за індексом), load (завантажує дані у оперативну пам’ять).

Generator наслідує torch.nn.Module. Цей клас описує найголовніший компонент цього дослідження – генератор. Реалізація класу показана на рисунку 4.14.

```
class Generator(nn.Module):
    def __init__(self, n_channels, n_classes, bilinear=True):
        super(Generator, self).__init__()
        self.n_channels = n_channels
        self.n_classes = n_classes
        self.bilinear = bilinear

        self.inc = (DoubleConv(n_channels, 64))
        self.down1 = (Down(64, 128))
        self.down2 = (Down(128, 256))
        self.down3 = (Down(256, 512))
        factor = 2 if bilinear else 1
        self.down4 = (Down(512, 1024 // factor))
        self.up1 = (Up(1024, 512 // factor, bilinear))
        self.up2 = (Up(512, 256 // factor, bilinear))
        self.up3 = (Up(256, 128 // factor, bilinear))
        self.up4 = (Up(128, 64, bilinear))
        self.outc = (OutConv(64, n_classes))

    def forward(self, x):
        x1 = self.inc(x)
        x2 = self.down1(x1)
        x3 = self.down2(x2)
        x4 = self.down3(x3)
        x5 = self.down4(x4)
        x = self.up1(x5, x4)
        x = self.up2(x, x3)
        x = self.up3(x, x2)
        x = self.up4(x, x1)
        logits = self.outc(x)
        return logits

```

Рис. 4.14 Програмна реалізація класу генератору

Наведемо опис його методів:

- Метод __init__ визначає архітектуру генератора. Він використовує допоміжні класи DoubleConv, Down, Up, OutConv, що реалізують в собі компоненти U-Net архітектури нейромережі завдяки модулю PyTorch. Параметрами цього модулю є кількість вхідних каналів (n_channels), кількість каналів на виході (n_classes) та флаг (bilinear), що вказує на те, використовувати шар білінійного підвищення дискретизації чи шар транспонованої згортки. Кількість вхідних та вихідних каналів має дорівнювати кількості каналів вхідного зображення (один – якщо зображення у відтінках сірого, три – якщо кольорове) та додатковий канал під матрицю шуму на вході та згенеровану сегментаційну маску на виході.

- Метод forward приймає на вхід x (тензор зображення нульового класу та шум) та проводить його крізь всю модель, щоб отримати на виході трансформоване зображення та цільову маску.

Другим за важливістю класом є Critic (див. рис. 4.15), що також наслідує torch.nn.Module. Цей клас є класичним прикладом реалізації згорткової нейронної мережі:

- Конструктор класу __init__ приймає два параметри: im_chan – кількість каналів у вхідному зображенні, що утворюється генератором., hidden_dim – коефіцієнт кількості нейронів прихованого шару (64 за замовчуванням). Конструктор використовує модуль nn.Sequential, що складається з кількох блоків, створених за допомогою методу make_crit_block. Кожен блок зменшує розмір зображення за допомогою операцій згортки.

- Метод make_crit_block – допоміжний метод, що створює окремий блок для модуля критика. Блок включає такі елементи: 2D-згортку (Conv2d), пакетну нормалізацію (BatchNorm2d), активаційну функцію Leaky ReLU для додавання нелінійності. Кожен блок має можливість бути кінцевим шаром, де відсутня нормалізація та активація.

- Метод forward потрібен для обчислення вихідних значень критика. Він отримує вхідне зображення, обробляє його через мережу критика і повертає результати після приведення до потрібного розміру.

```python
class Critic(nn.Module):
    def __init__(self, *, im_chan=3, hidden_dim=64):
        super().__init__()
        self.crit = nn.Sequential(
            self.make_crit_block(im_chan, hidden_dim, kernel_size=4, stride=2),
            self.make_crit_block(hidden_dim, hidden_dim, kernel_size=4, stride=2),
            self.make_crit_block(hidden_dim, hidden_dim * 2, kernel_size=4, stride=2),
            self.make_crit_block(hidden_dim * 2, hidden_dim * 4, kernel_size=4, stride=2),
            self.make_crit_block(hidden_dim * 4, hidden_dim * 8, kernel_size=4, stride=2),
            self.make_crit_block(hidden_dim * 8, 1, kernel_size=4, stride=1, final_layer=True),
        )

    def make_crit_block(
        self,
        input_channels,
        output_channels,
        kernel_size=4,
        stride=2,
        final_layer=False
    ):
        if not final_layer:
            return nn.Sequential(
                nn.Conv2d(input_channels, output_channels, kernel_size, stride, padding=1),
                nn.BatchNorm2d(output_channels),
                nn.LeakyReLU(0.2, inplace=True),
            )
        return nn.Sequential(
            nn.Conv2d(input_channels, output_channels, kernel_size, stride),
        )

    def forward(self, x):
        crit_pred = self.crit(x)
        return crit_pred.view(len(crit_pred), -1)
```

Рис. 4.15 Програмна реалізація класу критика

Клас EvalModel – це клас моделі з модулю тренування самовалідаційного компонента претренованої сегментаційної моделі для використання цього компонента для тренування ГЗМ.

Клас Train має схожі методи з класом Train із модулів сегментації. Його основні методи це __init__ – ініціалізація всіх компонентів генератора і критика (оптимізатор, додаткові моделі, метрики) для тренування, fit – метод для запуск

тренувального циклу, save – збереження всіх необхідних змінних для генератора і критика для випадку продовження навчання та використання моделі генератора у інших модулях.

Окремо потрібно виділити методи train_gen та train_critic, що запускають одну ітерації (пакет даних) тренування для генератору та критику відповідно. Ці методи йдуть кожну ітерацію кожної епохи один за одним поступово покращуючи ці змагальні компоненти за рахунок один одного.

Модуль візуалізації результатів генерування складається з окремих функцій для візуалізації:

- plot_models_loss_history – показує зміну в штрафах генератора та критику на протязі навчання, в залежності від номерів епох та ітерацій.
- plot_jaccard_index_history – показує зміну оціночної самовалідованої метрики під час навчання (див. рис. 4.16).
- plot_generator_output – показує результат трансформації вхідного зображення генератором.
- plot_generation_history – показує зміну результатів генерування на протязі епох з деяким шагом на одному й тому самому значенні випадкового стану (див. рис. 2.18).
- plot_segmentation_output – показує результат сегментації згенерованого зображення претренованою сегментаційною моделлю.

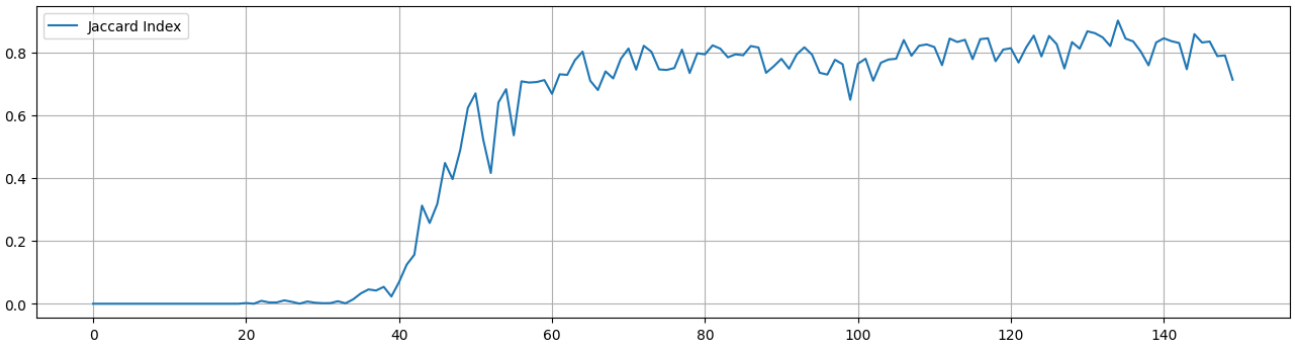


Рис. 4.16 Зміна метрики Жаккара під час навчання

Модуль часткової аугментації зображень реалізовано у вигляді веб-застосунку [10]. Веб-інтерфейс реалізовано з використанням Python бібліотеки Gradio [122].

Gradio – це бібліотека, яка дозволяє швидко створювати інтерактивні веб-інтерфейси для аналітики, машинного навчання та інших моделей. Вона дозволяє користувачам легко налаштовувати інтерфейси для введення та виведення даних, таких як текст, зображення, аудіо та інше, і миттєво інтегрувати їх з будь-якими Python функціями або моделями. Gradio спрощує процес демонстрації та тестування моделей, роблячи їх доступними через веб-браузер.

Код розробленого застосунку доступний для загального користування в публічному репозиторії на GitHub [123]. Архітектура програми є модульною, що спрощує процес додавання нових моделей для трансформації зображень у різних предметних доменах. Кожна генеративна модель представлена у вигляді класу на Python, який реалізує інтерфейс із двох основних методів:

__init__ – конструктор, що ініціалізує модель;

__call__(image: np.ndarray) -> np.ndarray – метод виклику моделі, який отримує на вхід вибраний фрагмент зображення та повертає відповідний трансформований фрагмент такого ж розміру. Це дозволяє безшовно інтегрувати трансформований фрагмент назад у вихідне зображення, уникаючи артефактів масштабування.

Діаграма класів веб-застосунку показана на рисунку 4.17. На діаграмі показано використання поліморфізму в базовому класі Model та наслідниках ShipModel та ToothModel завдяки визначенню спільних методів __init__, __call__. Головне вікно програми (клас MainWindow) може використовувати будь-яку модель без питань до внутрішньої логіки цих моделей).

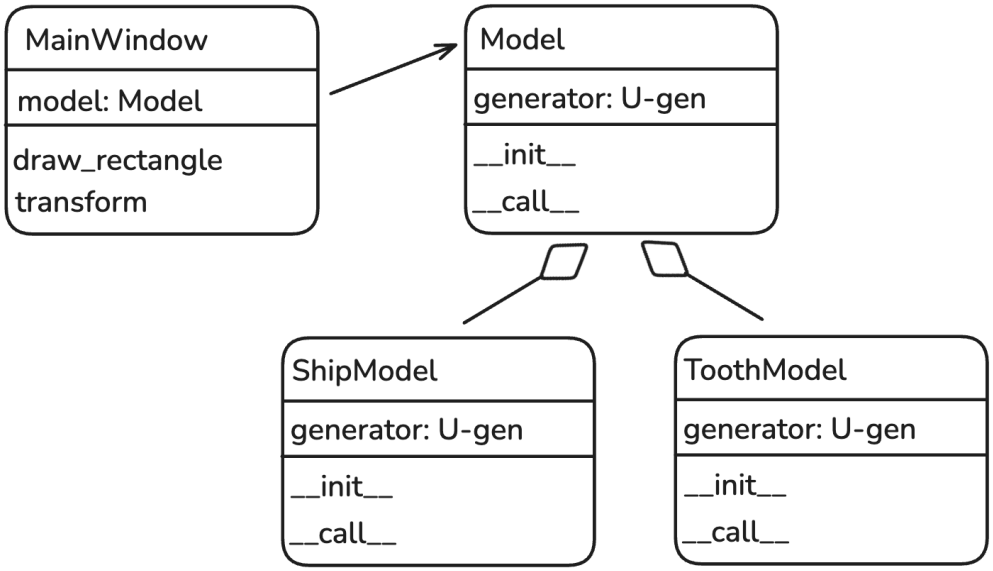


Рис. 4.17 Діаграма класів модулю часткової аугментації зображень

Приклад роботи застосунку часткової аугментації показано на рисунку 4.18. Спочатку потрібно вверху вибрати необхідну модель, в цьому прикладі це “ToothModel”. Далі користувач завантажує зображення у лівій частині, та наносить на зображенні зони, які він хоче аугментувати. Після цього натискається кнопка “Generate”, та через декілька секунд (в залежності від потужностей машини, на якій розгорнуто застосунок) у правій частині з’являється зображення, яке можна завантажити на пристрій.

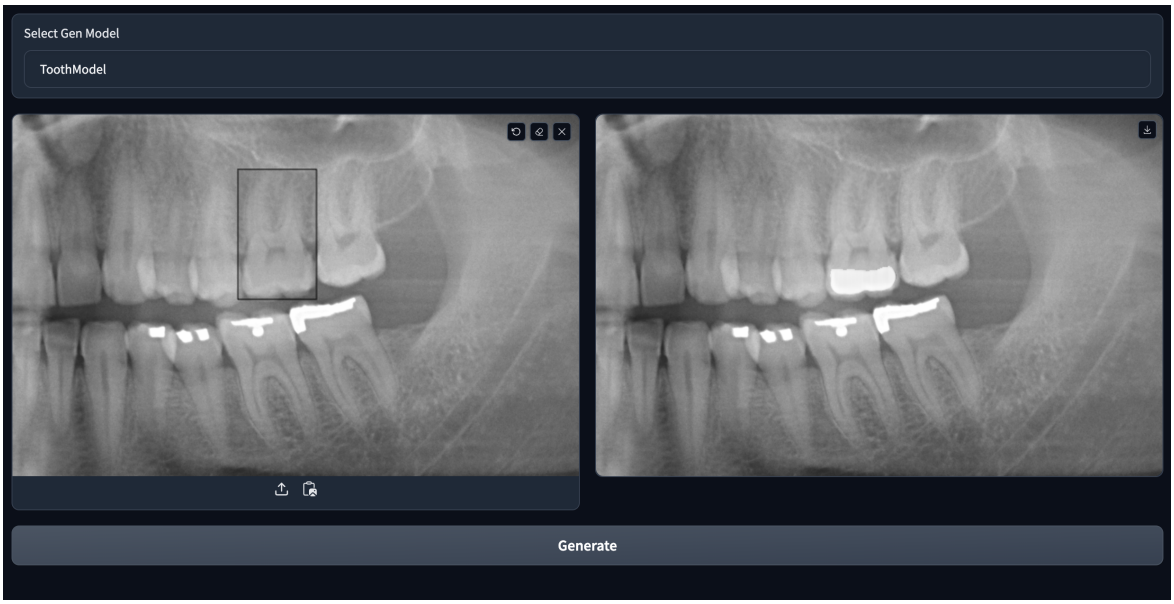


Рис. 4.18 Екранна форма застосунку часткової аугментації

Модуль оцінки частково згенерованих даних також реалізований з використанням бібліотеки Gradio.

При роботі з цим застосунком спочатку потрібно авторизуватись (логін та пароль на етапі написання дослідження видається для кожного конкретного користувача вручну) та вибрати необхідний набір даних для оцінки.

Далі в основному вікні програми (див. рис. 4.19) показується зображення користувачу по одному за ітерацію. Користувачу необхідно вибрати зони, які він вважає аугментованими. Після цього натискається кнопка для переходу до іншої картинки. В кінці роботи над конкретним набором у користувача є можливість подивитись на свої метрики.

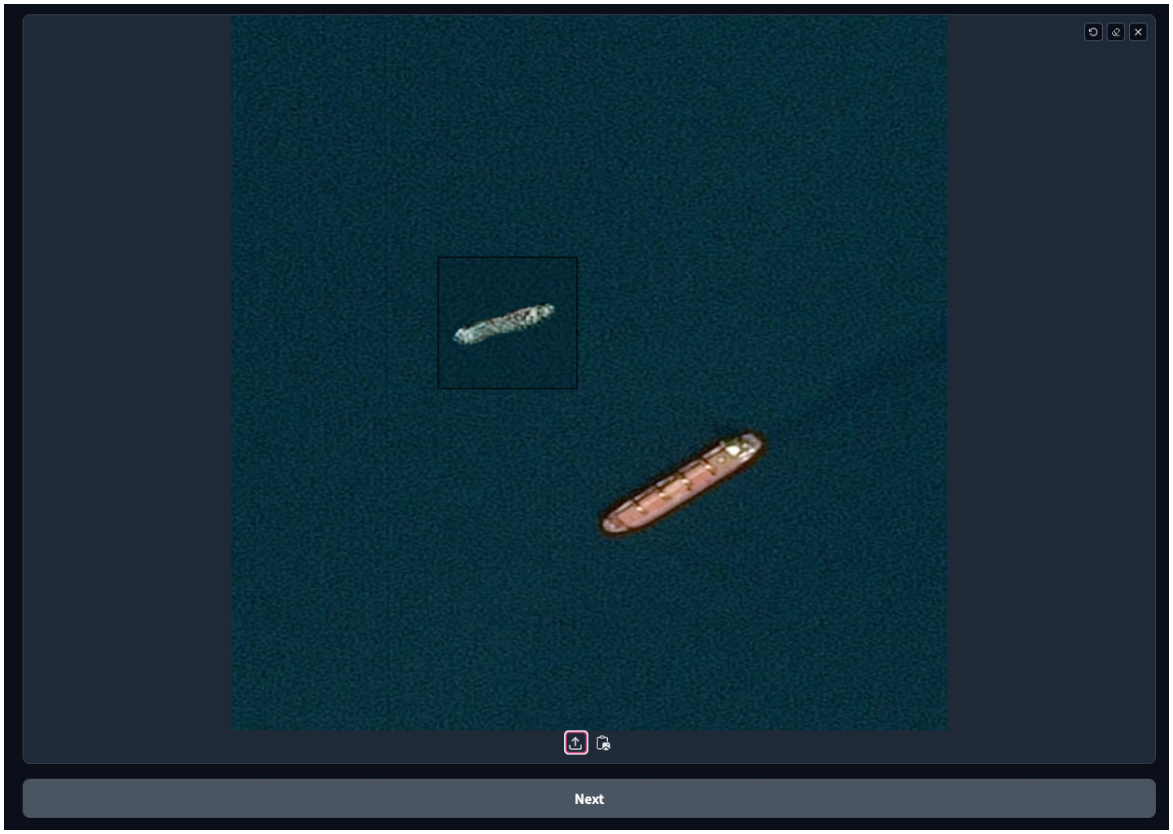


Рис. 4.19 Екранна форма застосунку оцінки згенерованих даних

4.3 Застосування створеної інформаційної технології для аугментації даних в процесі навчання моделей сегментації

Для оцінки ефективності розроблених методів було проведено кілька експериментів на різних наборах даних.

4.3.1 Трансформація зубів на стоматологічних панорамних рентгенівських знімках

Перша частина цього експерименту базується на результатах, представлених у розділі 2.3, де вдалося покращити точність сегментації, а також вдосконалити фінальну модель для сегментації пломб на зображеннях окремих зубів розміром 128x128 пікселів. Це було досягнуто за допомогою моделі U-генератора та післяобробки результатів генерування зображень і масок.

Метою цього експерименту є побудова аналогічної моделі сегментації, але з використанням додаткових компонентів самовалідації, описаних у розділах 2.4–2.5.

Цей підхід дозволяє автоматично визначати найкращу епоху, оптимізуючи час, витрачений на вибір моделі, і усувати необхідність постобробки згенерованого зображення, оскільки створюються зображення цільового класу, готові до використання та інтеграції.

Для проведення експерименту була натренована модель сегментації на 128 зображеннях, а тестування виконано на 20 зображеннях. Результати тестування продемонстрували високу точність моделі, яка склала 94,2% за коефіцієнтом Жаккара. Ця модель використовувалася як компонент самовалідації для перевірки якості згенерованих даних під час навчання самовалідованої ГЗМ.

Протягом 150 епох була натренована модель ГЗМ, причому найкраща епоха вибиралася на основі коефіцієнта Жаккара. Для підвищення якості навчання додатково застосовувалася функція втрат на основі квадратичної помилки. Початковий коефіцієнт для цієї функції був встановлений на рівні 10 із лінійним

зменшенням до 1 протягом 150 епох. Найкращою епохою, за обраною метрикою, була визначена 135-та епоха.

На рисунку 2.39 наведено графік експерименту, що демонструє динаміку зміни метрик генерації в процесі навчання моделі. Приклад отриманих зображень показано на рисунку 4.20. У першому ряду зображено вхідні зображення, які подаються до генератора разом із шумом, що дає можливість створювати нові варіації при кожному запуску. Другий ряд ілюструє результати генерації з доданою пломбою та відповідною маскою. Варто відзначити, що, крім області пломби, зміни в інших частинах зображення є мінімальними.

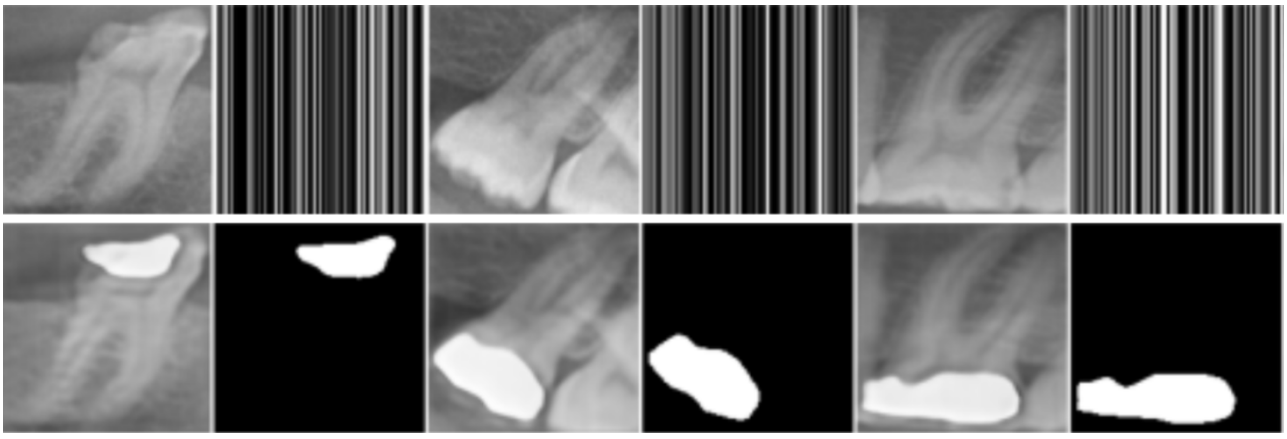


Рис. 4.20 Приклади генерації пломб

Завдяки роботі генератора базовий тренувальний набір для валідаційної сегментаційної моделі було автоматично доповнено 64 зображеннями. Модель, навчена на цьому розширеному наборі, досягла середнього значення метрики 96.3% (за результатами експериментів з 50 різними випадковими станами, див. таблицю 4.1), що на 2% перевищує показник моделі, навченої без додаткових даних (див. рис. 4.21). Це підтверджує ефективність використання генеративної моделі для покращення продуктивності сегментаційної моделі шляхом автоматичного розширення тренувального набору. Подібне покращення (96.2%) спостерігалося й у розділі 2.3, проте використання проміжних етапів допомогло автоматизувати процес тренування та вибору моделі.

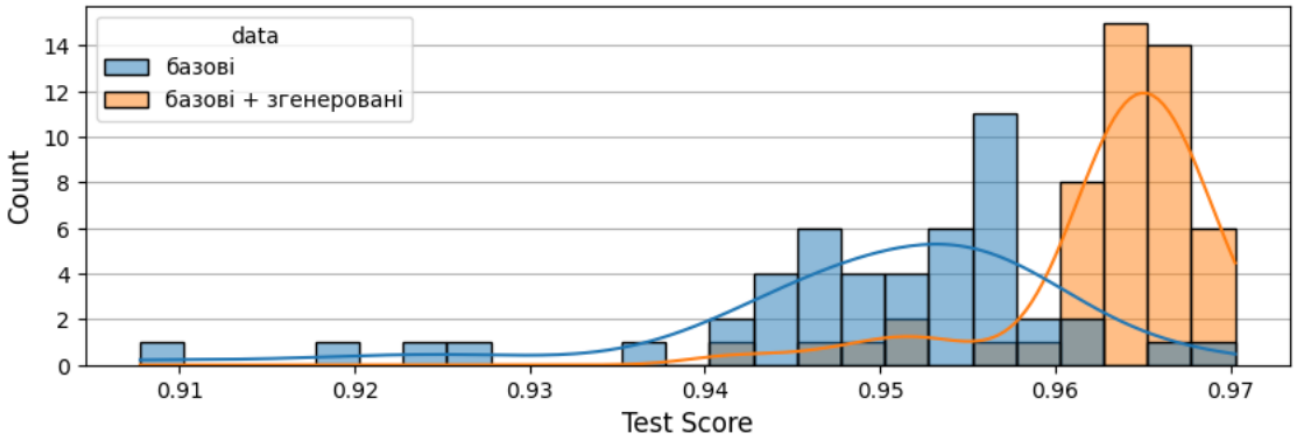


Рис. 4.21 Розподіл оцінок сегментації на тестових даних

Таблиця 4.1 Результати сегментаційної моделі для різних випадкових станів для базових даних та доданих даних створених самовалідованої ГЗМ

Seed	**Базові**			**Базові + 64 Згенеровані**		
	Тест	**Валідація**	**Тренування**	**Тест**	**Валідація**	**Тренування**
0	0.941	0.955	0.937	0.952	0.952	0.923
1	0.944	0.958	0.949	0.968	0.963	0.970
2	0.957	0.957	0.949	0.968	0.956	0.966
3	0.947	0.951	0.956	0.963	0.949	0.964
4	0.966	0.961	0.961	0.962	0.961	0.958
5	0.956	0.948	0.951	0.942	0.941	0.893
6	0.956	0.961	0.965	0.965	0.960	0.965
7	0.955	0.949	0.943	0.966	0.956	0.969
8	0.947	0.955	0.931	0.970	0.961	0.963
9	0.957	0.948	0.957	0.967	0.956	0.969
10	0.955	0.953	0.948	0.962	0.951	0.961
11	0.961	0.953	0.947	0.947	0.934	0.921
12	0.950	0.942	0.954	0.967	0.941	0.965
13	0.958	0.951	0.959	0.966	0.953	0.958

Продовження до табл. 4.1

Seed	Базові			Базові + 64 Згенеровані		
	Тест	Валідація	Тренування	Тест	Валідація	Тренування
14	0.944	0.964	0.954	0.959	0.967	0.957
15	0.953	0.951	0.954	0.962	0.962	0.963
16	0.951	0.955	0.962	0.965	0.953	0.966
17	0.952	0.956	0.956	0.969	0.965	0.965
18	0.955	0.946	0.936	0.969	0.961	0.972
19	0.956	0.961	0.957	0.964	0.963	0.958
20	0.918	0.930	0.918	0.969	0.961	0.958
21	0.945	0.957	0.944	0.965	0.964	0.962
22	0.957	0.963	0.950	0.967	0.963	0.958
23	0.949	0.954	0.927	0.952	0.928	0.943
24	0.954	0.961	0.890	0.962	0.960	0.957
25	0.908	0.912	0.943	0.967	0.958	0.965
26	0.960	0.966	0.958	0.965	0.967	0.961
27	0.938	0.945	0.940	0.968	0.969	0.967
28	0.951	0.960	0.959	0.967	0.963	0.964
29	0.794	0.793	0.842	0.967	0.952	0.963
30	0.952	0.956	0.881	0.964	0.959	0.958
31	0.957	0.959	0.961	0.964	0.962	0.954
32	0.835	0.840	0.914	0.967	0.958	0.964
33	0.942	0.947	0.939	0.963	0.954	0.952
34	0.946	0.946	0.948	0.964	0.958	0.965
35	0.928	0.923	0.928	0.962	0.938	0.934
36	0.956	0.957	0.957	0.965	0.957	0.966

Продовження до табл. 4.1

Seed	Базові			Базові + 64 Згенеровані		
	Тест	Валідація	Тренування	Тест	Валідація	Тренування
37	0.957	0.943	0.959	0.966	0.940	0.963
38	0.948	0.952	0.951	0.965	0.961	0.949
39	0.959	0.948	0.957	0.963	0.950	0.963
40	0.924	0.934	0.917	0.963	0.958	0.959
41	0.945	0.957	0.944	0.964	0.956	0.961
42	0.950	0.941	0.868	0.965	0.953	0.950
43	0.945	0.951	0.941	0.955	0.955	0.912
44	0.968	0.958	0.969	0.962	0.954	0.961
45	0.947	0.943	0.945	0.964	0.950	0.963
46	0.955	0.958	0.955	0.950	0.949	0.900
47	0.955	0.959	0.963	0.969	0.963	0.969
48	0.947	0.948	0.941	0.966	0.958	0.968
49	0.953	0.956	0.956	0.962	0.954	0.955

Друга частина експерименту зосереджена на сегментації пломб на панорамному КТ-знімку зубів. Для цього було використано базовий набір даних із 92 зображень, який було аугментовано за допомогою створеної генеративної моделі, що дозволило отримати варіації кожного з існуючих зображень. Аугментація здійснювалась частково, застосовуючи інструмент на основі генеративних моделей. Приклади таких варіацій наведено на рисунку 4.22. Важливо зазначити, що навіть для людського ока складно відрізнити пломби, які було додано штучно, що свідчить про ефективність запропонованого підходу.

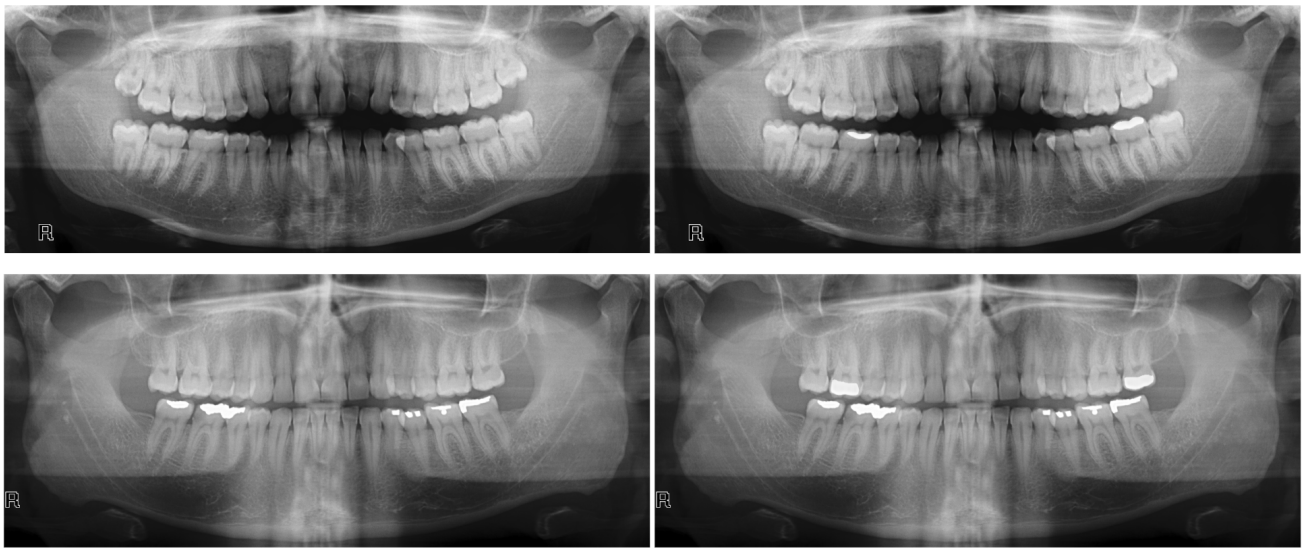


Рис. 4.22 Приклади трансформування реальних даних КТ-знімків розробленою моделлю.

Після створення додаткових даних було взято базову модель зі статті про оцінку якості передобробки панорамних стоматологічних рентгенівських знімків [124]. Її якість на звичайних даних становила 90,5% за коефіцієнтом Жаккара. Під час навчання використовувалися ті самі параметри, проте при виборі зображень для ітерації з ймовірністю 0,5 обиралося або реальне зображення, або трансформоване за описаним вище підходом. Коефіцієнт Жаккара для натренованої моделі становив 91,4%, що підтверджує ефективність запропонованого підходу.

4.3.2 Трансформація суден на супутникових знімках

Дані для цього експерименту взяті з конкурсу на платформі Kaggle, де учасники мали завдання досягти найвищого коефіцієнта Жаккара для сегментації кораблів на супутникових знімках [125]. Використання супутникових і дронових знімків стає дедалі важливішим у різних галузях, зокрема в сільському господарстві, оскільки ці дані дозволяють здійснювати точний моніторинг та аналіз ситуацій, таких як врожайність або поширення хвороб рослин, у реальному часі.

У цьому експерименті використовували ті ж моделі та гіперпараметри, що й у попередньому прикладі з КТ-знімками зубів. Основною відмінністю є те, що генератор працював із кольоровими зображеннями, що збільшує варіативність та складність даних. Претренована модель для оцінки мала точність 81%. Найкращою епохою для генератора визнано 82-гу епоху, а результати показано на рисунку 4.23.

Рис. 4.23 Результати генерації кораблів ГЗМ на кращій епосі

Як видно з рисунку 4.23, через складність даних генератор справляється із задачею менш якісно. Часто згенеровані кораблі є розмитими, не мають чітких контурів і виглядають спотвореними або «розбитими», як другий корабель у другому ряду. Маски також мають викривлення: деякі містять «дірки» (перший корабель у третьому ряду), або зайві частини (перший корабель у четвертому ряду). Проте, незважаючи на неточності, додаткові згенеровані дані допомогли покращити коефіцієнт Жаккара оцінювальної моделі з 81% до 84%, за умови додавання 64 згенерованих прикладів до тренувального набору.

На рисунку 4.24 показано повноцінні супутникові знімки розміром 768x768 пікселів із догенерованими кораблями. Було перевірено ефективність запропонованого підходу для цієї задачі. В якості базової моделі використовували U-net34 із публічної роботи конкурсу [126]. Взяли частину датасету з 128 зображень, до яких у деяких місцях додали 1-2 цільові кораблі, як показано на рисунку 4.25.

Рис. 4.24 Приклад аугментування супутникових знімків розробленою ГЗМ: знімок з Google Maps [127]

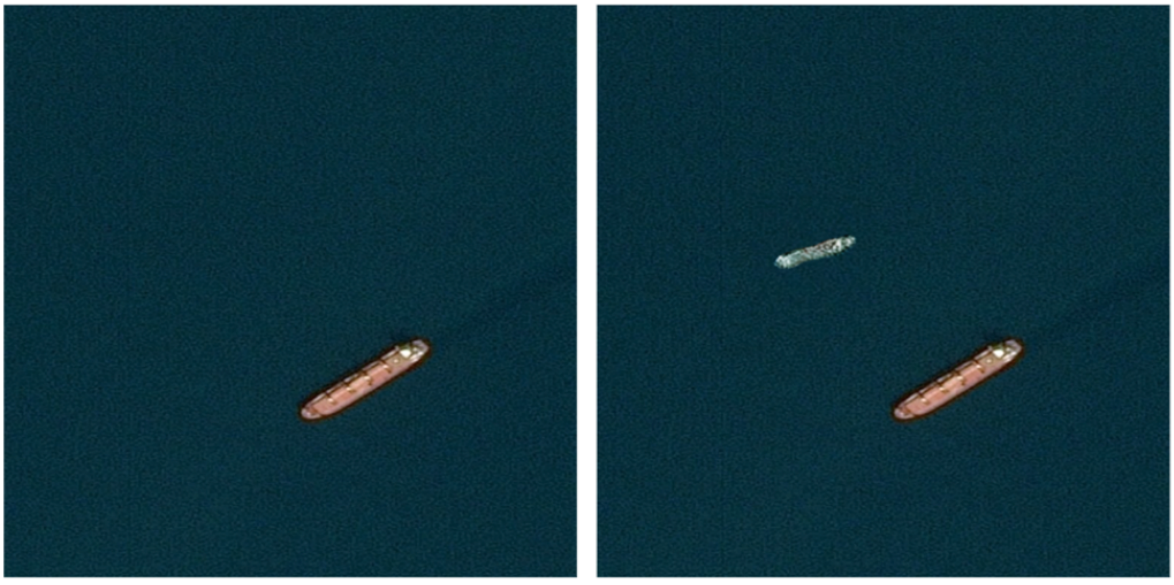


Рис. 4.25 Приклад аугментування супутникових знімків з набору даних Kaggle

Тренування для моделі сегментації відбувалося аналогічно до описаного у попередньому розділі. При відсутності додаткових даних точність моделі склала 63.4%, а з доданими даними зросла до 65.2%.

Загальні результати експериментів на двох наборах даних наведені у таблиці 4.2. Можна побачити що при тренуванні моделі сегментації з доданими даними від самовалідованої ГЗМ покращують результати у всіх експериментах.

Таблиця 4.2 Порівняння експериментів тренування сегментаційної моделі з додаванням згенерованих даних та без

Модель	**Jaccard Index (без згенерованих даних)**	**Jaccard Index (з згенерованими даними)**	**Різниця, %**
Модель сегментації цільових об’єктів зубів	94.2	96.3	2.1
Модель сегментації об’єктів кораблів	81.3	84	3.7
Модель сегментації повних зображень зубів	90.5	91.4	0.9
Модель сегментації повних зображень кораблів з супутника	63.4	65.2	1.8

4.4 Порівняння функціональних можливостей створеної інформаційної технології з аналогами

Серед аналогів можна виділити наступні фреймворки, які реалізовують схожий або частковий функціонал розробленої технології.

Бібліотека Albumentations [128] є популярним засобом для здійснення різних аугментацій зображень, які застосовуються для покращення моделей

комп'ютерного зору через збільшення різноманітності тренувальних даних. Вона надає широкий набір можливих трансформацій, таких як геометричні (наприклад, повороти, масштабування, зміщення), кольорові (зміни насиченості, контрастності, яскравості) та текстурні (наприклад, шуми, розмиття). Ці аугментації допомагають зменшити перенавчання та зробити моделі більш стійкими до різних варіантів вхідних даних.

Однією з ключових переваг Albumentations є її підтримка різних типів аугментації для різних задач комп'ютерного зору, таких як детектування об'єктів, сегментація та робота з ключовими точками. Це робить її універсальним інструментом, придатним для застосування в різноманітних сценаріях.

Однак, хоч Albumentations і пропонує великий вибір трансформацій, є певні обмеження. Зокрема, бібліотека не підтримує часткову аугментацію, що може бути корисним для завдань, де потрібно застосовувати трансформації лише до окремих частин зображення або об’єктів на ньому. Це може обмежувати гнучкість у сценаріях, де потрібно більш детально контролювати процес трансформації зображення.

На рисунку 4.26 представлений приклад аугментації повороту, виконаної за допомогою офіційного інтерфейсу Albumentations, що ілюструє, як виглядає застосування однієї з базових трансформацій бібліотеки.

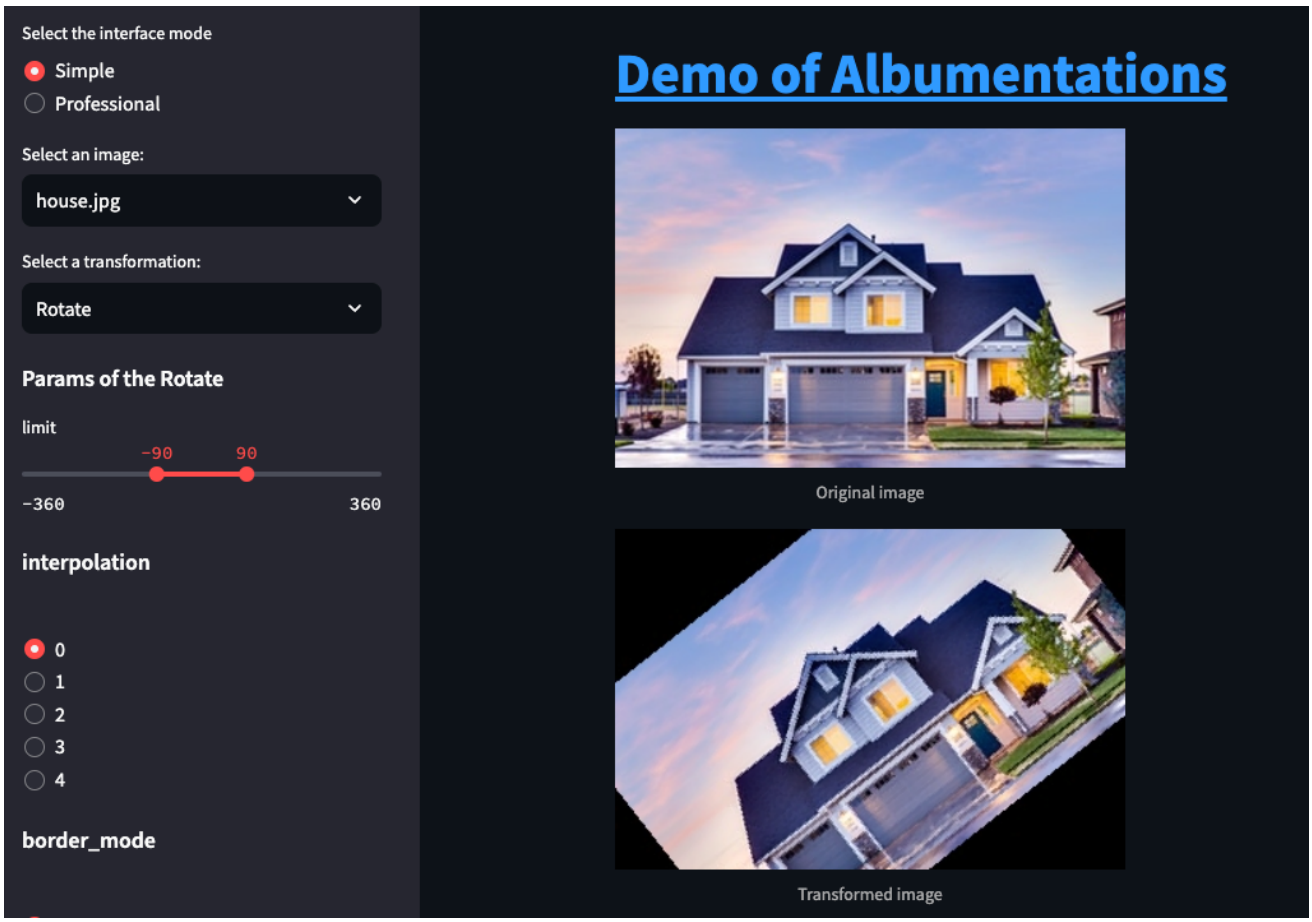


Рис. 4.26 Веб-сайт фреймворка albumentations [128]

Automatic1111 [129] – це популярний графічний інтерфейс для роботи з моделями Stable Diffusion, створений у вигляді базі веб-додатка для генерації зображень на основі текстових підказок та зображень (див. рис. 4.27). Він надає зручні функції для взаємодії з моделями генеративного штучного інтелекту, зокрема:

- Генерація зображень: Користувачі можуть вводити текстові підказки, які потім перетворюються в зображення за допомогою моделей Stable Diffusion.
- Різноманітні налаштування: Присутні параметри для налаштування якості, кількості кроків (ітерацій), ширини та висоти зображення, вибору необхідною кількості зображень та встановлення випадкового стану для повторюваності результатів.

- Інтерактивне редагування зображень: Він підтримує функції редагування на зразок inpainting (відновлення частин зображень) і outpainting (розширення меж зображення).

- Підтримка розширень: Завдяки модульній системі, Automatic1111 можна розширювати новими плагінами для покращення функціональності, одним із популярних плагінів є ControlNet, для додавання до SD додаткових моделей та інформації.

- Контроль над моделями та вагами: Дає можливість легко завантажувати й керувати різними версіями моделей Stable Diffusion, а також індивідуально налаштовувати їхні ваги.

- Зручний у використанні для інференсу на локальних машинах або серверах: Можна запускати як на персональному комп'ютері, так і на хмарних сервісах, що дає користувачам більшу гнучкість.

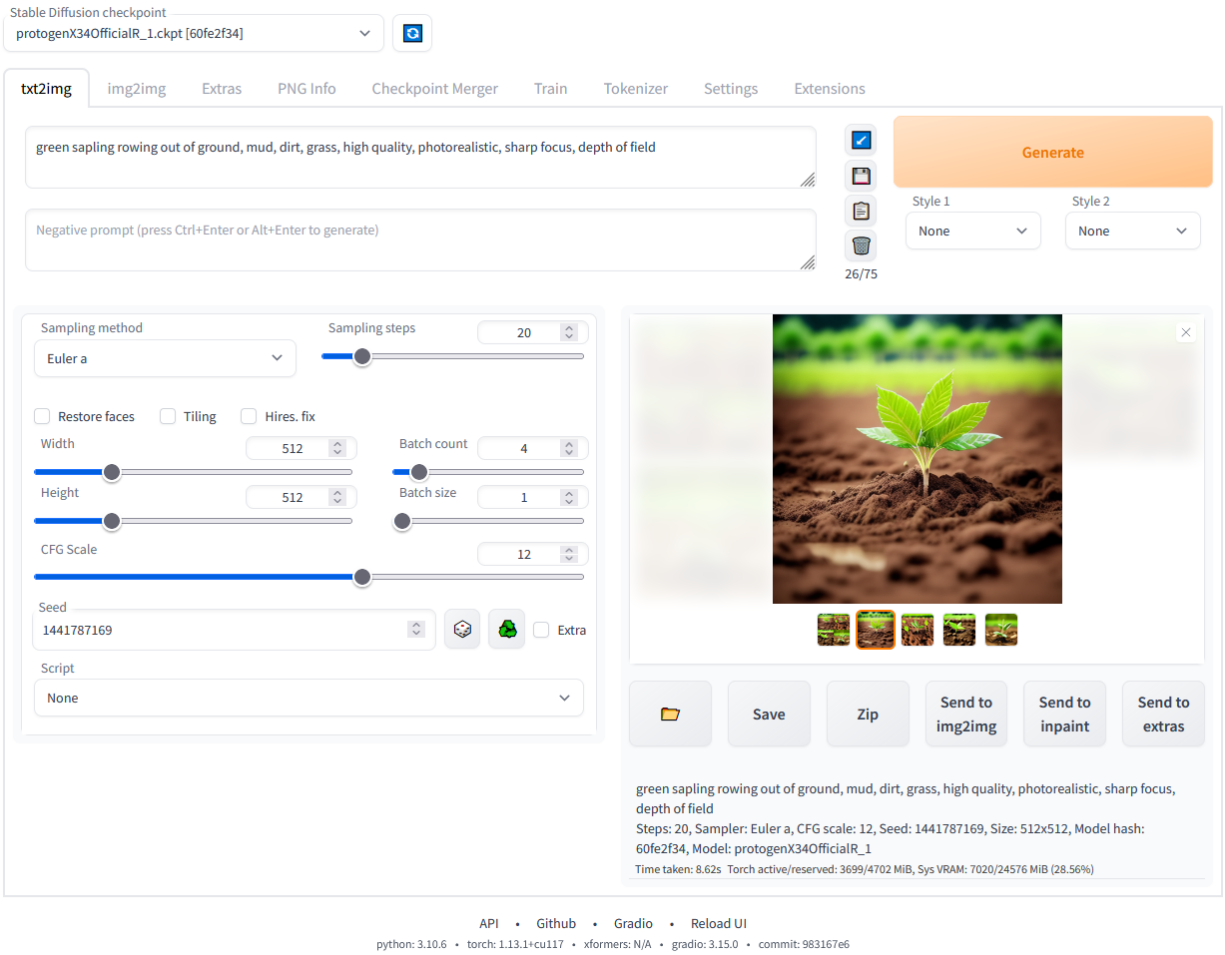


Рис. 4.27 Інтерфейс Stable Diffusion Web UI [129]

Як зазначено вище automatic1111 має підтримку інпейтингу [130] – техніки часткової аугментації зображень, яка дозволяє змінювати окремі частини зображення або додавати нові елементи, зберігаючи решту без змін. Це надзвичайно корисний інструмент, коли необхідно редагувати лише конкретні ділянки зображення, не втрачаючи загального контексту або стилю.

Як показано на прикладі (див. рис. 4.28), інпейтинг дозволяє змінювати фрагменти зображень, додаючи нові об'єкти (наприклад, капелюшок на собаці) або виправляти наявні елементи. Це особливо корисно для сценаріїв, де потрібна точна і локальна модифікація.

Однак, як і в запропонованій інформаційній системі, для використання цього інструменту необхідна модель тонко налаштована під конкретну область, оскільки доступні моделі не можуть зовсім згенерувати зубну пломбу або правильний корабель на супутниковому знімку (див. рис. 4.29).

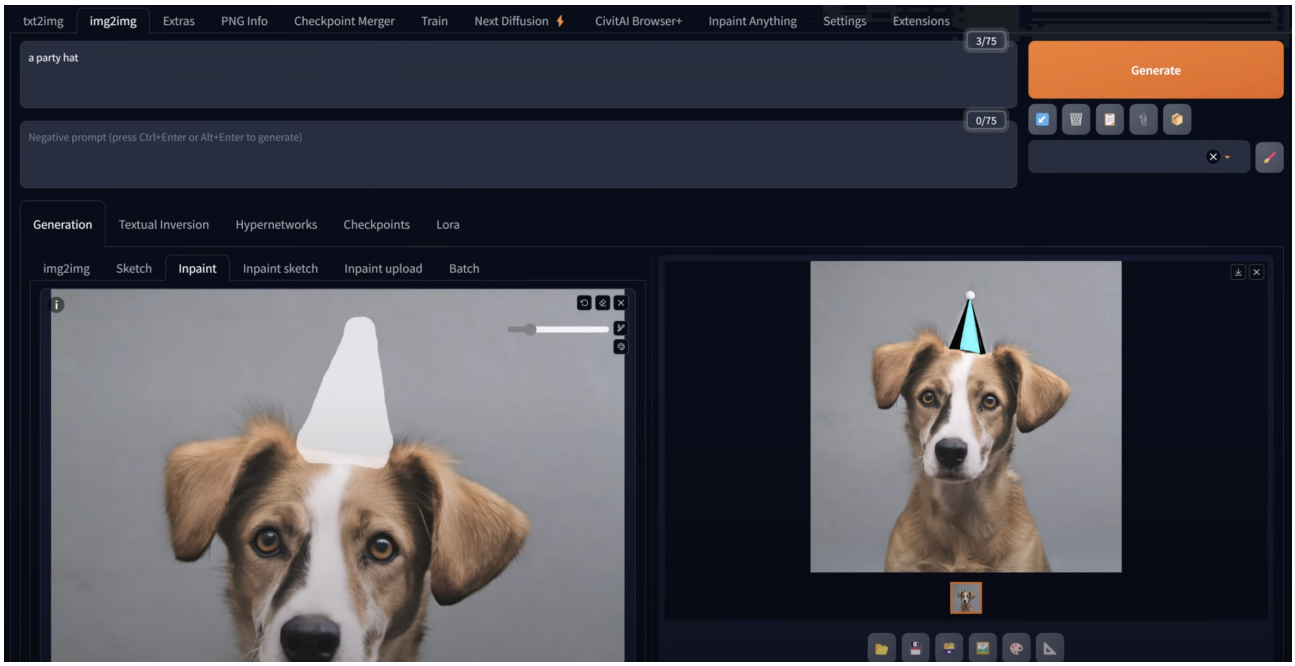


Рис. 4.28 – Приклад інпейтингу в automatic1111 [131]

Рис. 4.29 Результати створення корабля на супутниковому знімку з використанням SD

Іншим недоліком, який притаманний всього інпейтингу є відсутність одразу згенерованої маски для створеного об’єкта. Приблизна маска може бути отримана з тієї, що обрав користувач, але цілком можлива ситуація, що об’єкт буде не повністю збігатися з цією маскою (капелюх на зображенні 4.27 має менший розмір аніж намальована маска), тому потрібна додаткова постобробка.

4.5 Впровадження розробленої технології у навчальний процес

Результати дослідження впроваджені у навчальний процес Вінницького національного технічного університету під час викладання дисципліни «Інформаційні технології моніторингу та аналізу стану складних систем» для студентів, які навчаються за освітньою програмою «Інформаційні системи та технології» рівня «аспірант» спеціальності 126 Інформаційні системи та технології, а також – дисципліни «Інтернет речей та інтелектуальний аналіз даних» для аспірантів, які навчаються за освітньою програмою «Інформаційні системи та технології» рівня «доктора філософії» спеціальності 126 для навчання знанням і навичкам застосування методів і моделей машинного навчання та

глибокого навчання для генерування зображень. Ці результати можуть використовуватися як додатковий матеріал для навчання та поглибленого розуміння концепцій, пов’язаних із цією областю.

Зокрема, при викладанні цих дисциплін використовуються такі результати досліджень, отримані у дисертаційній роботі здобувача, що підтверджується актом впровадження ВНТУ від 19 вересня 2024 року (див. Додаток Д):

- підготовка набору даних для навчання генеративних моделей: вибір потрібного набору даних, первинний аналіз даних, розмітка, візуалізація розподілів, огляд сучасних основних Python-бібліотек для вирішення цих задач (cv2, albumentations, matplotlib, seaborn, pandas);
- тренування та використання моделей генерування, таких як autoencoder, variational autoencoder, Deep Convolutional Generative Adversarial Networks, Wasserstein GAN та StyleGAN2, з використанням сучасних інструментів та фреймворків на мові програмування Python (Keras, PyTorch);
- дослідження методів оцінки результатів генерування як за допомогою ручних методів (HYPE), так і автоматичних (Inception Score, Fréchet Inception Distance).

Використання зазначених результатів дозволило підвищити якість навчального процесу із згаданих дисциплін.

4.6 Висновки до розділу

У розділі 4 показано проектування, реалізація, використання, впровадження та порівняно з аналогами інформаційної технології самовалідованої генеративної змагальної мережі.

На етапі проектування:

- створено IDEF0 діаграма повноцінної системи;
- розроблена схема потрібних модулів;
- функціонал модулів показано з використанням UML-діаграми прецедентів;

- наведено діаграми послідовності та активності для описання взаємодії між модулями;
- спроектовано діаграму класів для опису програмного коду модулів.

Вибрано необхідні технології реалізації, такі як мова програмування Python та її відкриті бібліотеки pytorch, numpy, pandas, opencv, albumentations, gradio. Визначено потрібні сервіси для збереження та запуску коду:

- веб-платформа експертів науки про дані Kaggle для збереження, контролю версій та зручного доступу до наборів даних розроблених модулів для навчання.
- GitHub для збереження, контролю версій та публікації веб-сервісів.
- Visual Studio Code для написання модулів, що реалізують веб-застосунки.

Виконано реалізації інформаційної технології. Успішно застосовано для аугментації даних для процесу навчання моделей сегментації для наборів даних панорамних знімків зубів, а саме пломб та супутникових знімків кораблів. Порівняно якість сегментаційних моделей для частин зображень з лише цільовим класом та повних зображень, звідки ці частини були взяті, без використання розробленої технології та з нею. У всіх експериментах на обох наборах даних зафіксовано покращення метрик.

Проведено порівняння функціональних можливостей та якості з такими аналогами як: бібліотека мови програмування Python albumentations; веб-застосунок Stable Diffusion Web UI (automatic1111). Бібліотека albumentations хоч і надає потужний арсенал функцій для звичайного аугментування даних, але не здатна працювати з частковими аугментаціями. Automatic1111 має широкі можливості роботи з різноманітними задачами завдяки використанню Stable Diffusion моделей, але, як і в розробленій системі, для специфічних задач необхідно тонко налаштовувати власні моделі генерування, а часткова аугментація може бути проведена за рахунок задачі інпейтингу, але з обмеженням у вигляді коррекції масок сегментації.

Охарактеризовано впровадження у навчальний процес Вінницького національного технічного університету під час викладання дисциплін «Інформаційні технології моніторингу та аналізу стану складних систем» та «Інтернет речей та інтелектуальний аналіз даних» для аспірантів, які навчаються за освітньою програмою «Інформаційні системи та технології».

Результати розділу опубліковано в таких роботах автора: [4], [5], [10].

ВИСНОВКИ

Усі завдання, поставлені в ході дослідження, виконано. Мету дисертації досягнуто, а саме вирішено актуальне науково-прикладне завдання покращення моделей та методів генерування штучних даних на основі генеративних змагальних нейронних мереж, а також оцінки якості результатів генерування. Для цього розроблено інформаційну технологію з використанням глибокого навчання та сучасних засобів та інструментів проектування і реалізації програмних продуктів. Розроблена інформаційна технологія на основі самовалідованої генеративної змагальної мережі є взаємопов'язаним комплексом формалізованих постановок задач, моделей, алгоритмів і програм для тренування, оцінювання та використання генеративної та сегментаційної моделей під конкретний домен даних.

1. Проведено огляд відомих підходів до генерації даних за допомогою моделей глибокого навчання, таких як автокодувальники (звичайні та варіаційні), генеративні змагальні мережі (DCGAN, Wasserstein GAN, BigGAN, ProGAN, StyleGAN, StyleGAN2) і дифузійні моделі (Stable Diffusion 1-3, Stable Diffusion XL). Показано їхні переваги та недоліки. Вибір зроблено на користь генеративних змагальних мереж завдяки оптимальному балансу між перевагами та недоліками порівняно з іншими моделями. Недоліки автокодувальників включають втрату інформації через надмірне стиснення шару коду, що призводить до низької якості відновлених даних та обмежених можливостей генерації. Дифузійні моделі характеризуються високою обчислювальною складністю, що робить процес тренування та генерації ресурсомістким і тривалим, оскільки генерація даних може займати багато часу через поступовий характер процесу. Основними недоліками генеративних змагальних мереж є складність тренування, яка вимагає ретельного налаштування гіперпараметрів, і нестабільність процесу, зокрема ризик перенавчання, коли генератор і дискримінатор перестають покращуватися, що може призвести до колапсу мод генерованих даних. У даній роботі пропонується рішення цих проблем за допомогою методу самовалідації.

2. Виконано аналіз основних проблем оцінювання результатів генерації: на основі людської оцінки (методи HYPEtime та HYPE∞) та автоматичних методів оцінки (методи Inception Score, Fréchet Inception Distance, влучність, точність, покриття, щільність, GAN-train та GAN-test, непараметричний тест для виявлення копіювання даних, Perceptual Path Length, метрика лінійної роздільності). Проаналізовано переваги, недоліки та обмеження кожного з методів, зроблено акцент на недостатній увазі методів щодо оцінки частково згенерованих даних. Показано неефективність метрики FID на наборі частково аугментованих КТ-знімків зубів: для відомо поганих трансформацій вона дає такі ж значення, як і для якісних трансформацій.

3. Розроблено нову технологію для підвищення ефективності генеративних змагальних мереж. Запропоновано метод застосування відомої архітектури U-генератора в складі генеративної змагальної мережі для задачі трансформування цільового класу на основі нульового класу фону з одночасним генеруванням сегментаційної маски для створеного цільового об'єкта. На основі цього вперше запропоновано модель самовалідованої генеративної змагальної мережі для задачі трансформування цільового класу сегментації. Цей підхід включає введення до архітектури навчання генеративної змагальної мережі додаткових самовалідаційних компонентів, зокрема: компонент штрафу за значну зміну структури зображення, яке надходить на вхід генератора; компонент автоматичних метрик оцінювання, серед яких можна виділити модель на основі претренованої сегментаційної моделі, що сегментує згенероване зображення, та метрику, яка порівнює цю маску з маскою, автоматично створеною генератором. Це дозволяє в процесі навчання моделі використовувати інтерпретовані метрики оцінювання, що демонструють, наскільки згенеровані дані є зрозумілими для сегментаційної моделі та наскільки добре маска, згенерована генератором, відповідає вимогам.

4. Запропоновано підхід оцінювання частково аугментованих даних, заснований на людській оцінці. Він включає взаємодію з користувачами, яким необхідно визначити на повному зображенні зони, що були перетворені. В

залежності від точності визначення зон (величини перетину зон вказаних користувачем з реально зміненими зонами) розраховуються метрики користувача (влучність, повнота, F_1 міра). Ці метрики є оберненими до метрик моделі, результати якої оцінюються: чим краще людина ідентифікує аугментовані зони, тим менш правдоподібно їх створює модель. Проведено порівняльне оцінювання трьох розроблених моделей: 1) генеративна змагальна мережа на основі U-генератора; 2) та ж модель, але з постобробкою вихідного зображення і сегментаційної маски; 3) самовалідована ГЗМ. Оцінювання проводили 30 осіб для набору панорамних знімків зубів. Результати показали перевагу самовалідованої ГЗМ, що також підтверджується іншими метриками.

5. Розроблено структуру запропонованої інформаційної технології на основі самовалідованої генеративної змагальної мережі з окремим компонентом оцінювання та охарактеризовано 6 розроблених модулів. Побудовано контекстну діаграму IDEF0 для системи. Створено UML-діаграми використання для кожної з підсистем та UML-діаграми класів для опису програмної архітектури модулів. Також наведено діаграми послідовності та активності для опису взаємодії між модулями.

6. Розроблено програмно-інформаційне забезпечення для автоматизації запропонованих методів і компонентів інформаційної технології. На основні складові отримано свідоцтво про реєстрацію авторського права на твір (комп'ютерну програму) «Трансформація цільового класу для задачі сегментації з використанням U-GAN». Деякі модулі, такі як модуль тренування самовалідаційного компонента претренованої сегментаційної моделі, модуль тренування самовалідованої генеративно-змагальної мережі, модуль візуалізації результатів генерації, а також модуль тренування сегментаційної моделі для задачі сегментації повних зображень, були реалізовані у вигляді програм-ноутбуків на платформі Kaggle, але можуть бути легко перенесені на інші популярні платформи (AWS SageMaker, Google Colab, Jupyter Notebook або JupyterLab, Microsoft Azure Notebooks, PyCharm, Visual Studio Code). Модуль часткової аугментації зображень і модуль оцінки частково згенерованих даних було реалізовано у Visual Studio Code

у вигляді застосунків з веб-інтерфейсом для зручного використання іншими користувачами. Проведено випробування програмно-інформаційного забезпечення на реальних даних. Його успішно застосовано для аугментації даних у процесі навчання моделей сегментації на наборах даних панорамних знімків зубів (зокрема пломб) та супутникових знімках кораблів. Порівняно якість сегментаційних моделей для частин зображень із лише цільовим класом та для повних зображень, звідки ці частини були взяті, без використання розробленої технології та з її застосуванням. В усіх експериментах на обох наборах даних зафіксовано покращення метрик. Проведено порівняння функціональних можливостей та якості з такими аналогами, як бібліотека для мови програмування Python albumentations і веб-застосунок Stable Diffusion Web UI.

7. Результати дисертаційного дослідження рекомендовані для впровадження в практичну діяльність при розробці інтелектуальних систем для вирішення задач комп'ютерного зору в ТОВ «ОРНЕТІКА». Також проведено впровадження у навчальний процес Вінницького національного технічного університету під час викладання дисциплін «Інформаційні технології моніторингу та аналізу стану складних систем» і «Інтернет речей та інтелектуальний аналіз даних» для аспірантів, які навчаються за освітньою програмою «Інформаційні системи та технології».

Основні результати дисертаційного дослідження викладені в 10 наукових працях. Опубліковано 4 статті у фахових журналах України за спеціальністю 126; 1 статтю, що включена до наукометричної бази даних Scopus; 5 тез доповідей на науково-практичних конференціях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Я. О. Ісаєнков і О. Б. Мокін, "Аналіз генеративних моделей глибокого навчання та особливостей їх реалізації на прикладі WGAN," *Вісник Вінницького політехнічного інституту*, том 160, вип. 1, с. 82–94, Берез. 2022. doi: 10.31649/1997-9266-2022-160-1-82-94.

[2] O. Bychkov, L. Zubyk, D. Gololobov, Y. Isaienkov, G. Grynkevych, and A. Ivanytska, "The System for Recognizing Useful Information of the Client's ID-Card Based on Machine Learning Technologies," *CEUR Workshop Proceedings,* vol. 2845, pp. 159–169, Dec. 2023.

[3] Я. О. Ісаєнков і О. Б. Мокін, "Трансформація цільового класу для задачі сегментації з використанням U-GAN," *Вісник Вінницького політехнічного інституту*, том 172, вип. 1, с. 81–87, Лют. 2024. doi:10.31649/1997-9266-2024-172-1-81-87.

[4] Я. О. Ісаєнков і О. Б. Мокін, "Самовалідований U-GAN для трансформації цільового класу в задачах сегментації," *Вісник Вінницького політехнічного інституту*, том 174, вип. 3, с. 102-111, Черв. 2024. doi: 10.31649/1997-9266-2024-174-3-102-111.

[5] Я. О. Ісаєнков і О. Б. Мокін, "Метод оцінки частково згенерованих даних," *Наука, технології, інновації*, том 31, вип. 3, с. 114-126, Вер. 2024. doi: 10.35668/2520-6524-2024-3-13.

[6] Я. Ісаєнков і О. Мокін, "Проблема оцінювання генеративних змагальних мереж," *у Матеріалах L Науково-технічної конференції підрозділів Вінницького національного технічного університету*, 2021, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2021/paper/view/12468 .

[7] Я. Ісаєнков і О. Мокін, "Артефакт межі у генеративних змагальних мережах," у *Матеріалах LI Науково-технічної конференції підрозділів Вінницького національного технічного університету*, 2022, Вінницький

національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/16039 .

[8] Я. Ісаєнков, "Використання методів глибокого навчання для оцінки якості медичних записок пацієнтів," *у Матеріалах конференції Проблеми вищої математичної освіти: виклики*, 2022, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/pmovc/pmovc22/paper/view/15558 .

[9] Я. Ісаєнков і О. "Мокін О, "Вплив гіперпараметрів дифузійних моделей на якість генерування," у *Матеріалах LII Науково-технічної конференції підрозділів Вінницького національного технічного університету*, 2023, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/17174 .

[10] Я. Ісаєнков, О. Мокін, О. Хорошева, і А. Герасимович, "Розробка застосунку для часткової аугментації зображень на основі генеративних моделей," у *Матеріалах LIII Науково-технічної конференції підрозділів Вінницького національного технічного університету,* 2024, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20614 .

[11] Ісаєнков Я.О. Авторське свідоцтво на комп'ютерну програму «Трансформація цільового класу для задачі сегментації з використанням U-GAN», №130591 від 14.10.2024 р.

[12] ChatGPT. Електронний ресурс. Режим доступу: https://chat.openai.com/ (01.09.2024).

[13] Midjorney. Електронний ресурс. Режим доступу: https://www.midjourney.com/home (01.09.2024).

[14] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and Improving the Image Quality of StyleGAN," *in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8107-8116. https://doi.org/10.1109/CVPR42600.2020.00813 .

[15] S. Sundaram and N. Hulkund, “GAN-based Data Augmentation for Chest X-ray Classification,” *arXiv:2107.02970 [cs, eess]*, Jul. 2021, Available: https://arxiv.org/abs/2107.02970 .

[16] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, “CNN-Generated Images Are Surprisingly Easy to Spot… for Now,” *in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8692-8701. https://doi.org/10.1109/CVPR42600.2020.00872 .

[17] This vessel does not exist. [Online]. Available: https://thisvesseldoesnotexist.com/ . (01.09.2024).

[18] S. Chaudhuri, D. Ritchie, J. Wu, K. Xu, and H. Zhang, "Learning Generative Models of 3D Structures," *Computer Graphics Forum*, vol. 39, no. 2, pp. 643-666, 2020. doi: https://doi.org/10.1111/cgf.14020.

[19] A. Sage, E. Agustsson, R. Timofte, and L. Van Gool, "Logo Synthesis and Manipulation With Clustered Generative Adversarial Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[20] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, “Cascaded Diffusion Models for High Fidelity Image Generation,” *arXiv:2106.15282 [cs]*, Dec. 2021, Available: https://arxiv.org/abs/2106.15282.

[21] Augmented Reality for Jewelry. [Online]. Available: https://tryon.jewelry/main . (01.09.2024).

[22] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, “Generative Adversarial Networks,” *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, Jul. 2021, doi: https://doi.org/10.1145/3459992.

[23] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, “Deepfakes and beyond: A Survey of face manipulation and fake detection,” *Information Fusion*, vol. 64, pp. 131–148, Dec. 2020, doi: https://doi.org/10.1016/j.inffus.2020.06.014.

[24] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48, M. F. Balcan and K. Q.

Weinberger, Eds., New York, NY, USA: PMLR, Jun. 2016, pp. 1060-1069. [Online]. Available: https://proceedings.mlr.press/v48/reed16.html.

[25] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,”*2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, doi: https://doi.org/10.1109/cvpr.2017.632.

[26] В. Гаврілов, і В. Данилов, “ГЕНЕРУВАННЯ МУЗИЧНИХ КОМПОЗИЦІЙ МЕТОДАМИ МАШИННОГО НАВЧАННЯ,” *ГРААЛЬ НАУКИ*, pp. 277–280, Nov. 2021, doi: 10.36074/grail-of-science.19.11.2021.052.

[27] M. Bińkowski *et al.*, “High Fidelity Speech Synthesis with Adversarial Networks,” *arXiv:1909.11646 [cs, eess]*, Sep. 2019. Available: https://arxiv.org/abs/1909.11646v2.

[28] M. Pasini, “MelGAN-VC: Voice Conversion and Audio Style Transfer on arbitrarily long samples using Spectrograms,” *arXiv.org*, 2019. Available: https://arxiv.org/abs/1910.03713.

[29] Voice Cloning for Content Creators. [Online]. Available: https://www.respeecher.com/. (01.09.2024).

[30] M. Wang, "Video Description with GAN," *2020 IEEE 3rd International Conference on Computer and Communication Engineering Technology (CCET)*, Beijing, China, 2020, pp. 10-13, doi: 10.1109/CCET50901.2020.9213129.

[31] D. Croce, G. Castellucci, and R. Basili, "GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Jul. 2020, pp. 2114-2119. doi: https://10.18653/v1/2020.acl-main.191.

[32] L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, and H. Li, “Generative Adversarial Network for Abstractive Text Summarization,” in arXiv e-prints, 2017. [Online]. Available: https://arxiv.org/pdf/1711.09357.pdf .

[33] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, “Efficient GAN-Based Anomaly Detection,” in *arXiv e-prints*, 2018. [Online]. Available: https://arxiv.org/pdf/1802.06222.pdf .

[34] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, “ClusterGAN: Latent Space Clustering in Generative Adversarial Networks,” in *arXiv e-prints*, 2018. [Online]. Available: https://arxiv.org/pdf/1809.03627.pdf .

[35] M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AIChE Journal*, vol. 37, no. 2, pp. 233–243, Feb. 1991, doi: https://doi.org/10.1002/aic.690370209 .

[36] D. Kingma, and M. Welling, “Auto-Encoding Variational Bayes,” in *arXiv e-prints*, 2013. [Online]. Available: https://arxiv.org/pdf/1312.6114.pdf .

[37] I. Csiszar, “I-Divergence Geometry of Probability Distributions and Minimization Problems,” *The Annals of Probability*, vol. 3, no. 1, Feb. 1975, doi: https://doi.org/10.1214/aop/1176996454 .

[38] I. Goodfellow *et al.*, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: https://doi.org/10.1145/3422622.

[39] M. Maschler, E. Solan, and S. Zamir, *Game Theory*. Cambridge University Press, 2013.

[40] J. A. Rice, *Mathematical Statistics and Data Analysis*. Duxbury Resource Center, 1995.

[41] M. Feurer and F. Hutter, “Hyperparameter Optimization,” *Automated Machine Learning*, pp. 3–33, 2019, doi: https://doi.org/10.1007/978-3-030-05318-5_1.

[42] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” in *arXiv e-prints*, 2015. [Online]. Available: https://arxiv.org/pdf/1511.06434.pdf .

[43] M. Baraban, S. Baraban, and V. Garmash, “DEVELOPMENT OF A PROGRESSIVE WEB APPLICATION WITH A CONVOLUTIONAL NEURAL NETWORK FOR IMAGE RECOGNITION,” *Information technology and computer engineering*, vol. 50, no. 1, pp. 7–14, 2021, doi: https://doi.org/10.31649/1999-9941-2021-50-1-7-14.

[44] R. Kvetny, R. Masliy, and O. Kyrylenko, “Detection and classification of traffic objects using the environment digits,” *Optoelectronic information-power*

technologies, vol. 39, no. 1, pp. 14–20, 2020, doi: https://doi.org/10.31649/1681-7893-2020-39-1-14-20.

[45] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang, “Color image segmentation: advances and prospects,” *Pattern Recognition*, vol. 34, no. 12, pp. 2259–2281, Dec. 2001, doi: https://doi.org/10.1016/S0031-3203(00)00149-7.

[46] V. Lytvyn, N. Oleksiv and M. Nazarkevych, "Tanks detection and recognition based on custom dataset model," *2024 IEEE 4th International Conference on Smart Information Systems and Technologies (SIST)*, Astana, Kazakhstan, 2024, pp. 583-586, doi: 10.1109/SIST61555.2024.10629455.

[47] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: https://doi.org/10.1038/nature14539.

[48] M. Arjovsky, S. Chintala; and L. Bottou, “Wasserstein Generative Adversarial Networks,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 214-223.

[49] L. Rüschendorf, “The Wasserstein distance and approximation theorems,” *Probability Theory and Related Fields*, vol. 70, no. 1, pp. 117–129, Mar. 1985, doi: https://doi.org/10.1007/bf00532240.

[50] S. Hochreiter, “The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 06, no. 02, pp. 107–116, Apr. 1998, doi: https://doi.org/10.1142/s0218488598000094.

[51] C. Garbin, X. Zhu, and O. Marques, “Dropout vs. batch normalization: an empirical study of their impact to deep learning,” *Multimedia Tools and Applications*, vol. 79, no. 19–20, pp. 12777–12815, Jan. 2020, doi: https://doi.org/10.1007/s11042-019-08453-9.

[52] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved training of wasserstein GANs,” in Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS’17), 2017, pp. 5769-5779. doi: https://dl.acm.org/doi/10.5555/3295222.3295327.

[53] A. Brock, J. Donahue, and K. Simonyan, “Large Scale GAN Training for High Fidelity Natural Image Synthesis,” in arXiv e-prints, 2018. [Online]. Available: https://arxiv.org/pdf/1809.11096.pdf .

[54] S. Suthaharan, “Support Vector Machine,” *Machine Learning Models and Algorithms for Big Data Classification*, vol. 36, pp. 207–235, 2016, doi: https://doi.org/10.1007/978-1-4899-7641-3_9.

[55] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive Growing of GANs for Improved Quality, Stability, and Variation,” *arXiv.org*, 2017. https://arxiv.org/abs/1710.10196.

[56] CelebA Dataset. [Online]. Available: https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html/CelebA/CelebA.html (01.09.2024).

[57] T. Karras, S. Laine, and T. Aila, “A Style-Based Generator Architecture for Generative Adversarial Networks,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, doi: https://doi.org/10.1109/cvpr.2019.00453.

[58] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and Improving the Image Quality of StyleGAN,” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8107-8116. http://doi.org/10.1109/CVPR42600.2020.00813 .

[59] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, “Diffusion Models in Vision: A Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 1–20, 2023, doi: https://doi.org/10.1109/TPAMI.2023.3261988.

[60] L. Fei-Fei, J. Deng, and K. Li, “ImageNet: Constructing a large-scale image database,” *Journal of Vision*, vol. 9, no. 8, pp. 1037–1037, Mar. 2010, doi: https://doi.org/10.1167/9.8.1037.

[61] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *Lecture Notes in Computer Science*, vol. 9351, pp. 234–241, 2015, doi: https://doi.org/10.1007/978-3-319-24574-4_28.

[62] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models,” *2022 IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2022, doi: https://doi.org/10.1109/cvpr52688.2022.01042.

[63] Stable Diffusion 2.0 Release. [Online]. Available: https://stability.ai/news/stable-diffusion-v2-release (01.09.2024).

[64] A. Radford *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," *www.semanticscholar.org*, 2021. Avaliable: https://www.semanticscholar.org/paper/Learning-Transferable-Visual-Models-From-Natural-Radford-Kim/6f870f7f02a8c59c3e23f407f3ef00dd1dcf8fc4

[65] P. Esser *et al.*, "Scaling Rectified Flow Transformers for High-Resolution Image Synthesis," *arXiv (Cornell University)*, Mar. 2024, doi: https://doi.org/10.48550/arxiv.2403.03206.

[66] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020, Available: https://jmlr.org/papers/volume21/20-074/20-074.pdf.

[67] W. Peebles and S. Xei, "Scalable Diffusion Models with Transformers," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2022, pp. 4172-4182. doi: https://doi.org/10.48550/arXiv.2212.09748.

[68] D. Podell *et al.*, "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis," *arXiv.org*, Jul. 04, 2023. doi: https://doi.org/10.48550/arXiv.2307.01952.

[69] C. Meng *et al.*, "SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations," *arXiv:2108.01073 [cs]*, Jan. 2022. doi: https://doi.org/10.48550/arXiv.2108.01073.

[70] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," *arXiv.org*, Jun. 10, 2016. doi: https://doi.org/10.48550/arXiv.1606.03498.

[71] C. Szegedy *et al.*, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015, doi: https://doi.org/10.1109/cvpr.2015.7298594.

[72] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium,” Jun. 2017, doi: https://doi.org/10.48550/arxiv.1706.08500.

[73] S. Mehdi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly, “Assessing Generative Models via Precision and Recall,” *arXiv (Cornell University)*, Jan. 2018, doi: https://doi.org/10.48550/arxiv.1806.00035.

[74] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo, “Reliable Fidelity and Diversity Metrics for Generative Models,” *arXiv (Cornell University)*, Jan. 2020, doi: https://doi.org/10.48550/arxiv.2002.09797.

[75] K. Shmelkov, C. Schmid, and K. Alahari, “How Good Is My GAN?,” *Springer eBooks*, pp. 218–234, Jan. 2018, doi: https://doi.org/10.1007/978-3-030-01216-8_14.

[76] C. Meehan, K. Chaudhuri, and S. Dasgupta, “A Non-Parametric Test to Detect Data-Copying in Generative Models,” *arXiv.org*, Apr. 12. doi: https://doi.org/10.48550/arXiv.2004.05675.

[77] S. Zhou, M. L. Gordon, R. Krishna, A. Narcomey, L. Fei-Fei, and M. S. Bernstein, “HYPE: A Benchmark for Human eYe Perceptual Evaluation of Generative Models,” *arXiv.org*, Oct. 31, 2019. doi: https://doi.org/10.48550/arXiv.1904.01121.

[78] T. N. Cornsweet, “The Staircase-Method in Psychophysics,” *The American Journal of Psychology*, vol. 75, no. 3, pp. 485–491, 1962, doi: https://doi.org/10.2307/1419876.

[79] A. Odena, V. Dumoulin, and C. Olah, “Deconvolution and Checkerboard Artifacts,” *Distill*, vol. 1, no. 10, Oct. 2016, doi: https://doi.org/10.23915/distill.00003.

[80] This Person Does Not Exist. [Online]. Available: https://thispersondoesnotexist.com/. (01.09.2024).

[81] O.V. Komenchuk, V.B. Mokin, Ye.M. Kryzhanovsky, and V.O. Budiak, “Intelligent Technology of Buildings Plan Construction, Based on Aerial Photography of their Roofs,” *Visnyk of Vinnytsia Politechnical Institute*, vol. 172, no. 1, pp. 101–109, Jan. 2024, doi: https://doi.org/10.31649/1997-9266-2024-172-1-101-109.

[82] В. Данилов, і О. Зарицький, “ВИКОРИСТАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ СЕРЦЕВИХ ЗАХВОРЮВАНЬ ЗА ДАНИМИ ЕКГ ТА ЗАСТОСУВАННЯ СУЧАСНИХ ГЕНЕРАТИВНО ЗМАГАЛЬНИХ МЕРЕЖ ДЛЯ ЗБІЛЬШЕННЯ НАВЧАЛЬНОЇ ВИБІРКИ,” *InterConf*, no. 14(113), pp. 472–478, Jun. 2022, doi: 10.51582/interconf.19-20.06.2022.050.

[83] Flowers Dataset. [Online]. Available: https://www.kaggle.com/datasets/ihelon/flowers. (01.09.2024).

[84] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, “Deconvolutional networks,” *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2010, doi: https://doi.org/10.1109/cvpr.2010.5539957.

[85] Apply a 2D Transposed Convolution Operation in PyTorch. [Online]. Available: https://www.geeksforgeeks.org/apply-a-2d-transposed-convolution-operation-in-pytorch/. (01.09.2024).

[86] X. Glorot, A. Bordes, and Y. Bengio. "Deep sparse rectifier neural networks." *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 2011.

[87] M. Abramowitz, *HANDBOOK of mathematical functions : With Formulas, Graphs, and Mathematical Tables*. National Bureau Of Standards, Washington, 1972.

[88] A. H. Reynolds, “Convolutional Neural Networks (CNNs),” 2019. [Online]. Avaliable: https://anhreynolds.com/blogs/cnn.html. (01.09.2024).

[89] J. Xu, Z. Li, B. Du, M. Zhang, and J. Liu, “Reluplex made more practical: Leaky ReLU,” *2020 IEEE Symposium on Computers and Communications (ISCC)*, Jul. 2020, doi: https://doi.org/10.1109/iscc50000.2020.9219587.

[90] W. H. Press and E. Al, *Numerical recipes in C : the art of scientific computing*. New Delhi: Cambridge University Press, 2009.

[91] M. Marchesi, “Megapixel Size Image Creation using Generative Adversarial Networks,” *arXiv.org*, May 31, 2017. doi: https://doi.org/10.48550/arXiv.1706.00082.

[92] T. T. H. Nguyen, A. Jatowt, M. Coustaty, and A. Doucet, “Survey of Post-OCR Processing Approaches,” *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–37, Jul. 2021, doi: https://doi.org/10.1145/3453476.

[93] PaddleOCR. [Online]. Avaliable: https://github.com/PaddlePaddle/PaddleOCR?tab=readme-ov-file. (01.09.2024).

[94] W. Yu, T. Chang, X. Guo, X. Wang, B. Liu, and Y. He, “UGAN: Unified Generative Adversarial Networks for Multidirectional Text Style Transfer,” *IEEE Access*, vol. 8, pp. 55170–55180, 2020, doi: https://doi.org/10.1109/access.2020.2980898.

[95] Abdi and S. Kasaei, “Panoramic Dental X-rays With Segmented Mandibles,” 2020. [Online]. Available: https://data.mendeley.com/datasets/hxt48yk462/2.

[96] “International Organization for Standardization. International Standard 3950-1984(E). Dentistry--designation system for teeth and areas of the oral cavity.,” *PubMed*, vol. 30, no. 2, pp. 134–5, Apr. 1985.

[97] O. Köksoy, “Multiresponse robust design: Mean square error (MSE) criterion,” *Applied Mathematics and Computation*, vol. 175, no. 2, pp. 1716–1729, Apr. 2006, doi: https://doi.org/10.1016/j.amc.2005.09.016.

[98] L. Mentaschi, G. Besio, F. Cassola, and A. Mazzino, “Problems in RMSE-based wave model validations,” *Ocean Modelling*, vol. 72, pp. 53–58, Dec. 2013, doi: https://doi.org/10.1016/j.ocemod.2013.08.003.

[99] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)?,” *Geoscientific Model Development Discussions*, vol. 7, no. 1, pp. 1525–1534, Feb. 2014, doi: https://doi.org/10.5194/gmdd-7-1525-2014.

[100] J. Tayman and D. A. Swanson, “On the validity of MAPE as a measure of population forecast accuracy,” *Population Research and Policy Review*, vol. 18, no. 4, pp. 299–322, 1999, doi: https://doi.org/10.1023/a:1006166418051.

[101] A. A. Kalinin, V. Iglovikov, A. Rakhlin, and A. A. Shvets, “Medical Image Segmentation Using Deep Neural Networks with Pre-trained Encoders,” *Advances in*

intelligent systems and computing, pp. 39–52, Jan. 2020, doi: https://doi.org/10.1007/978-981-15-1816-4_3.

[102] R. Real and J. M. Vargas, "The Probabilistic Basis of Jaccard's Index of Similarity," *Systematic Biology*, vol. 45, no. 3, pp. 380–385, 1996, doi: https://doi.org/10.2307/2413572.

[103] L. R. Dice, "Measures of the Amount of Ecologic Association Between Species," *Ecology*, vol. 26, no. 3, pp. 297–302, Jul. 1945, doi: https://doi.org/10.2307/1932409.

[104] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (PCA)," *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, Mar. 1993, doi: https://doi.org/10.1016/0098-3004(93)90090-r.

[105] J. Leonard., *Systems engineering fundamentals : supplementary text*. Defense Systems Management College: The Press, 2001.

[106] Jupyter. [Online]. Avaliable: https://jupyter.org/. (01.09.2024).

[107] Kaggle: Your Machine Learning and Data Science Community. [Online]. Avaliable: https://www.kaggle.com/. (01.09.2024).

[108] Data Sheet: Tesla P100. [Online]. Avaliable: https://images.nvidia.com/content/tesla/pdf/nvidia-tesla-p100-PCIe-datasheet.pdf. (01.09.2024).

[109] Amazon SageMaker. [Online]. Avaliable: https://aws.amazon.com/sagemaker/. (01.09.2024).

[110] Google Colaboratory. [Online]. Avaliable: https://colab.google/. (01.09.2024).

[111] Learn more about all the notebooks experiences from Microsoft and GitHub. [Online]. Avaliable: https://visualstudio.microsoft.com/vs/features/notebooks-at-microsoft/. (01.09.2024).

[112] PyCharm: The Python IDE for data science and web development. [Online]. Avaliable: https://www.jetbrains.com/pycharm/. (01.09.2024).

[113] Visual Studio Code: Code. Editing. Redefined. [Online]. Avaliable: https://code.visualstudio.com/. (01.09.2024).

[114] Numpy. [Online]. Avaliable: https://numpy.org/. (01.09.2024).

[115] Pandas. [Online]. Avaliable: https://pandas.pydata.org/. (01.09.2024).

[116] Matplotlib: Visualization with Python. [Online]. Avaliable: https://matplotlib.org/. (01.09.2024).

[117] seaborn: statistical data visualization. [Online]. Avaliable: https://seaborn.pydata.org/. (01.09.2024).

[118] PyTorch. [Online]. Avaliable: https://pytorch.org/. (01.09.2024).

[119] scikit-learn: Machine Learning in Python. [Online]. Avaliable: https://scikit-learn.org/stable/index.html. (01.09.2024).

[120] OpenCV. [Online]. Avaliable: https://opencv.org/. (01.09.2024).

[121] Segmentation Models. [Online]. Avaliable: https://segmentation-models-pytorch.readthedocs.io/en/latest/. (01.09.2024).

[122] Gradio: Build & share delightful machine learning apps. [Online]. Avaliable: https://www.gradio.app/. (01.09.2024).

[123] Target Class Transformation For Segmentation Task. [Online]. Avaliable: https://github.com/yisaienkov/target_class_transformation_for_segmentation_task_tool. (01.09.2024).

[124] O.V. Komenchuk and O.B. Mokin, “Analysis of Methods for Preprocessing of Panoramic Dental X-Rays for Image Segmentation Tasks,” *Visnyk of Vinnytsia Politechnical Institute*, vol. 170, no. 5, pp. 41–49, Jan. 2023, doi: https://doi.org/10.31649/1997-9266-2023-170-5-41-49.

[125] W. Reade, and J. Faudi, “Airbus Ship Detection Challenge,” *Kaggle: Your Machine Learning and Data Science Community*. [Online]. Available: https://www.kaggle.com/competitions/airbus-ship-detection. (01.09.2024).

[126] M. Shugaev, “Unet34 submission TTA (0.699 new public LB),” *Kaggle: Your Machine Learning and Data Science Community*. [Online]. Available: https://www.kaggle.com/code/iafoss/unet34-submission-tta-0-699-new-public-lb. (01.09.2024).

[127] Google Maps. [Online]. Avaliable: https://www.google.com/maps. (01.09.2024).

[128] Albumantations. Do more with less data. [Online]. Avaliable: https://albumentations.ai/. (01.09.2024).

[129] Stable Diffusion web UI. [Online]. Avaliable: https://github.com/AUTOMATIC1111/stable-diffusion-webui. (01.09.2024).

[130] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, 2000, doi: https://doi.org/10.1145/344779.344972.

[131] Wolf Dynamics World - WDW, "Advanced inpainting with Automatic1111," *YouTube*, Jan. 14, 2024. Avaliable: https://www.youtube.com/watch?v=HBsCDgVIWlU. (01.09.2024).

ДОДАТКИ

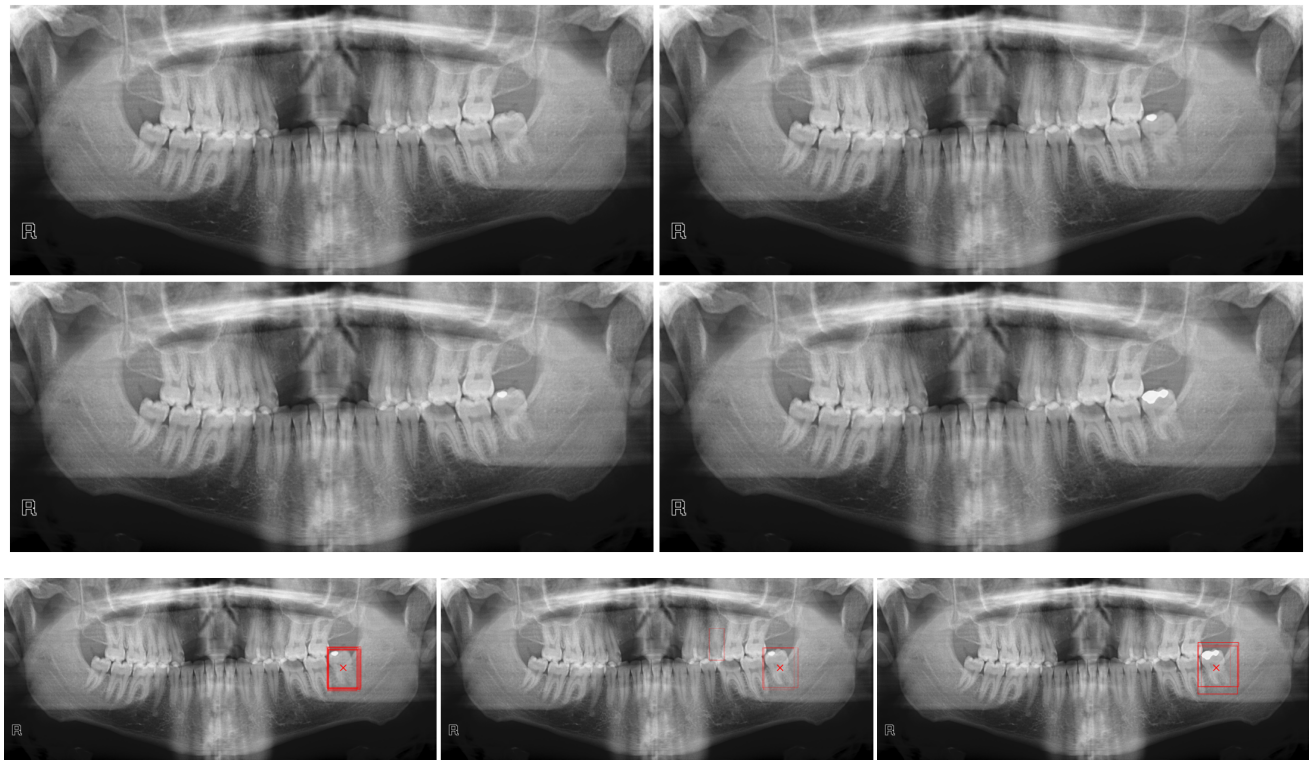
ДОДАТОК А
ТРАНСФОРМОВАНІ НАБОРИ ДАНИХ ПАНОРАМНИХ ЗНІМКІВ ЗУБІВ ТА ОЦІНЮВАННЯ ЇХ КОРИСТУВАЧАМИ

Нижче наведено 20 панорамних рентгенівських знімків (зображення згори ліворуч кожного рисунку, де представлено сітку 2x2 панорамних знімків).

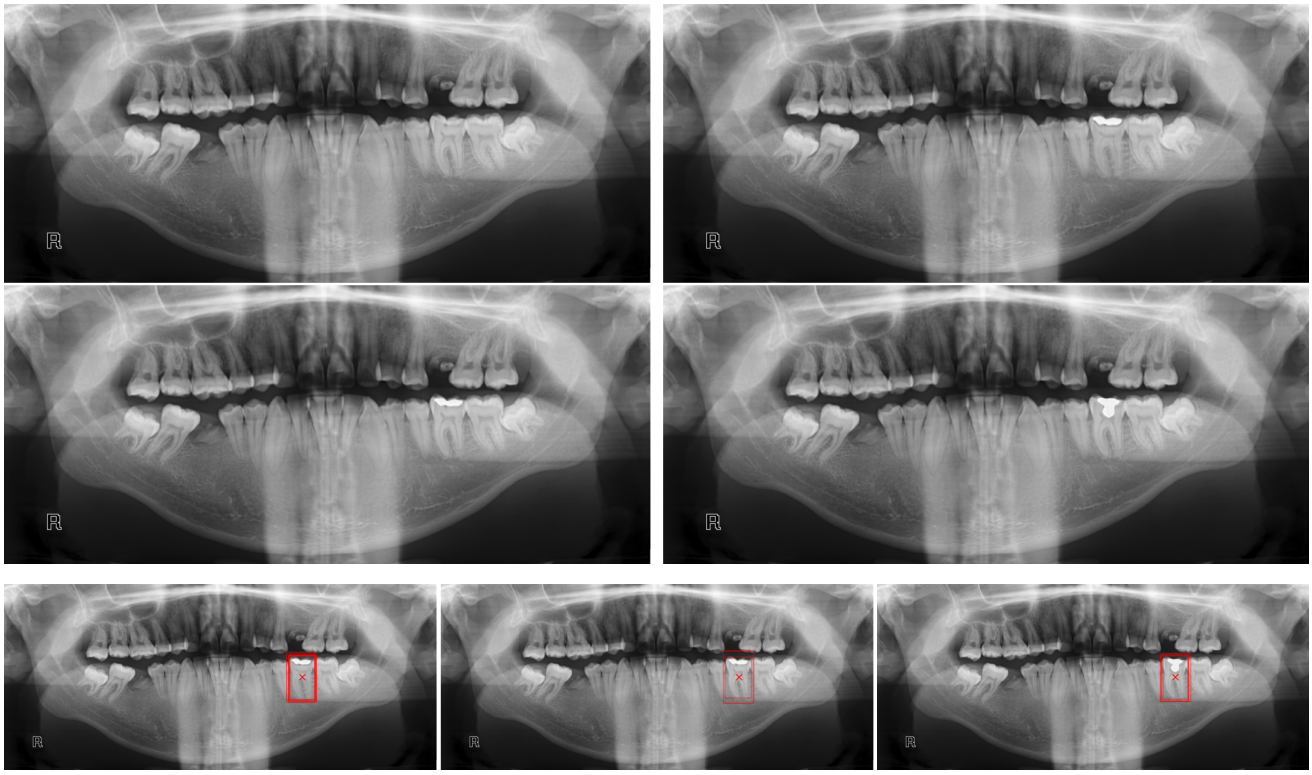
Для кожного з них додатково показано результати трансформацій за допомогою трьох генеративних моделей: 1) генеративна змагальна мережа на основі U-генератора (зображення згори праворуч); 2) та ж модель з постобробкою вихідного зображення та сегментаційної маски (зображення знизу ліворуч); 3) самовалідована генеративна змагальна мережа (зображення знизу праворуч).

Також відображено вибрані користувачами фрагменти, де, на їх думку, зображені згенеровані ділянки (для кожної моделі представлене одне зображення в ряду, де представлено сітку 1x3 панорамних знімка).

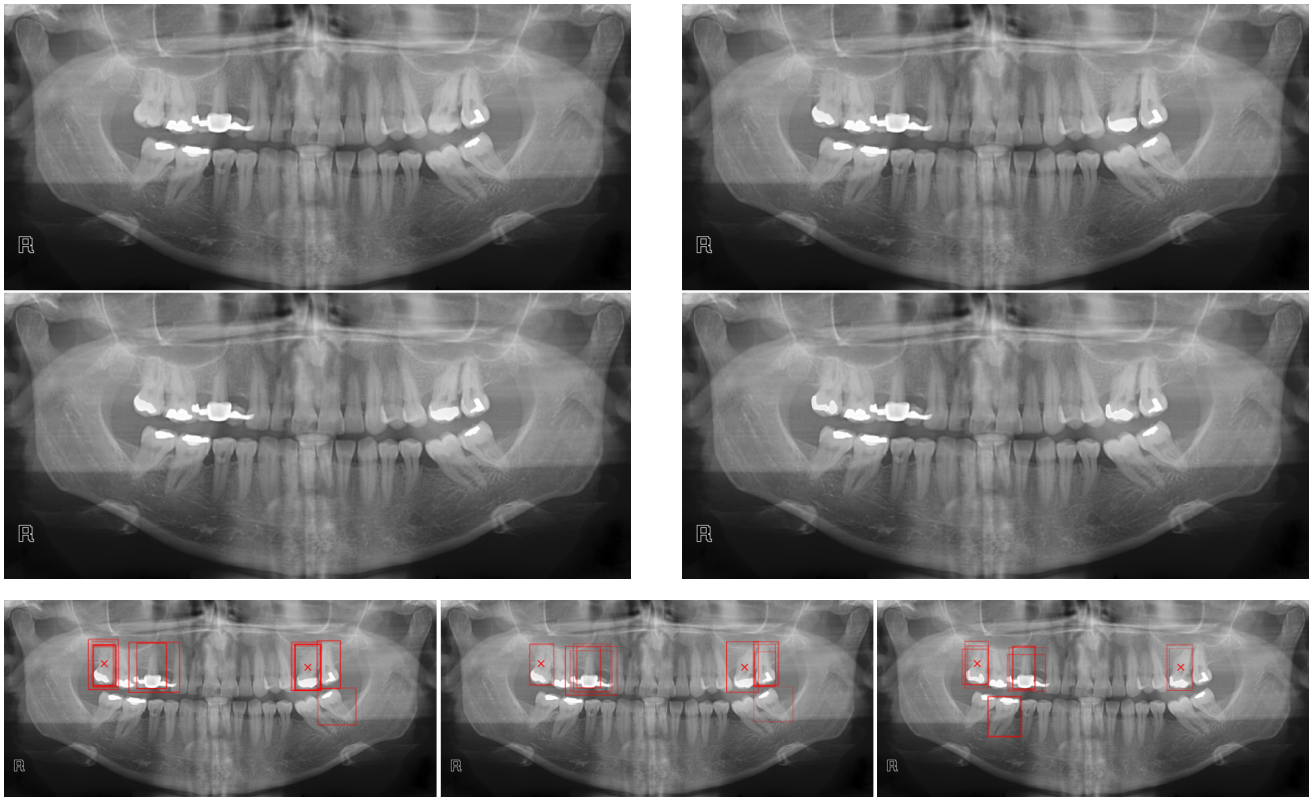
Приклад 1:



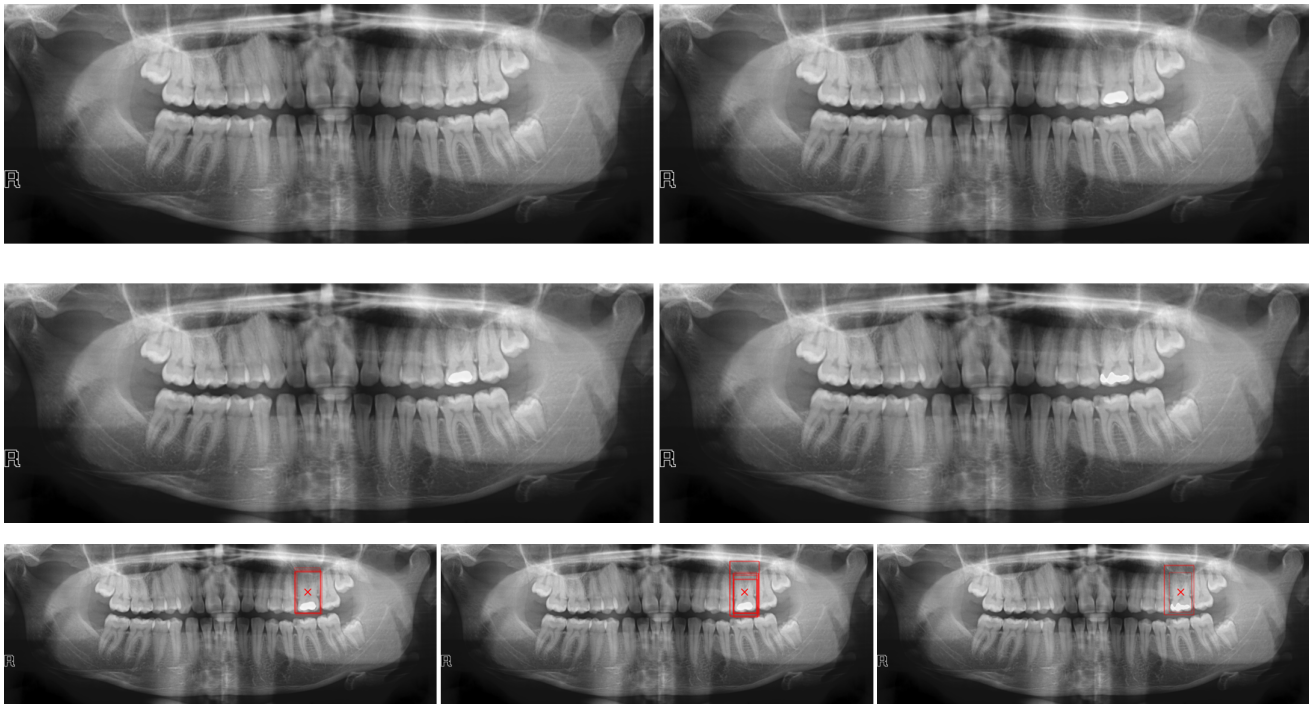
Приклад 2:



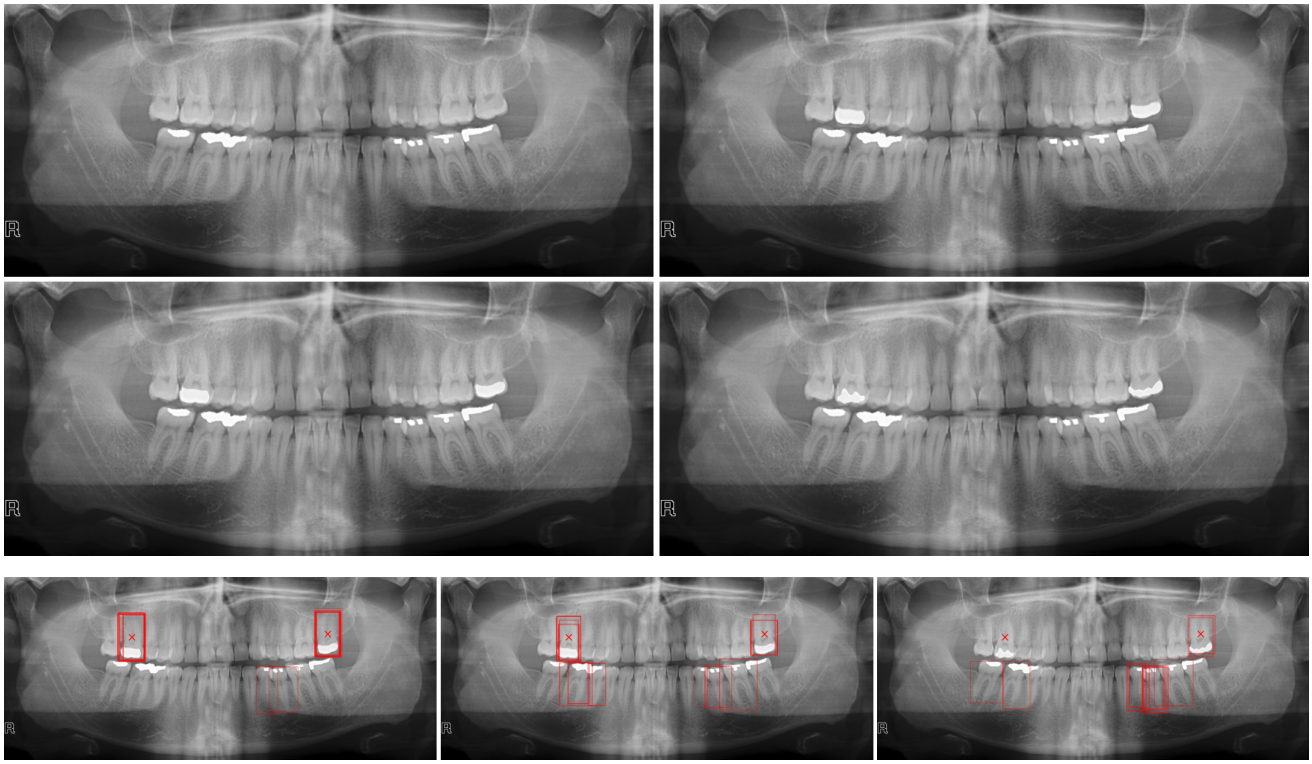
Приклад 3:



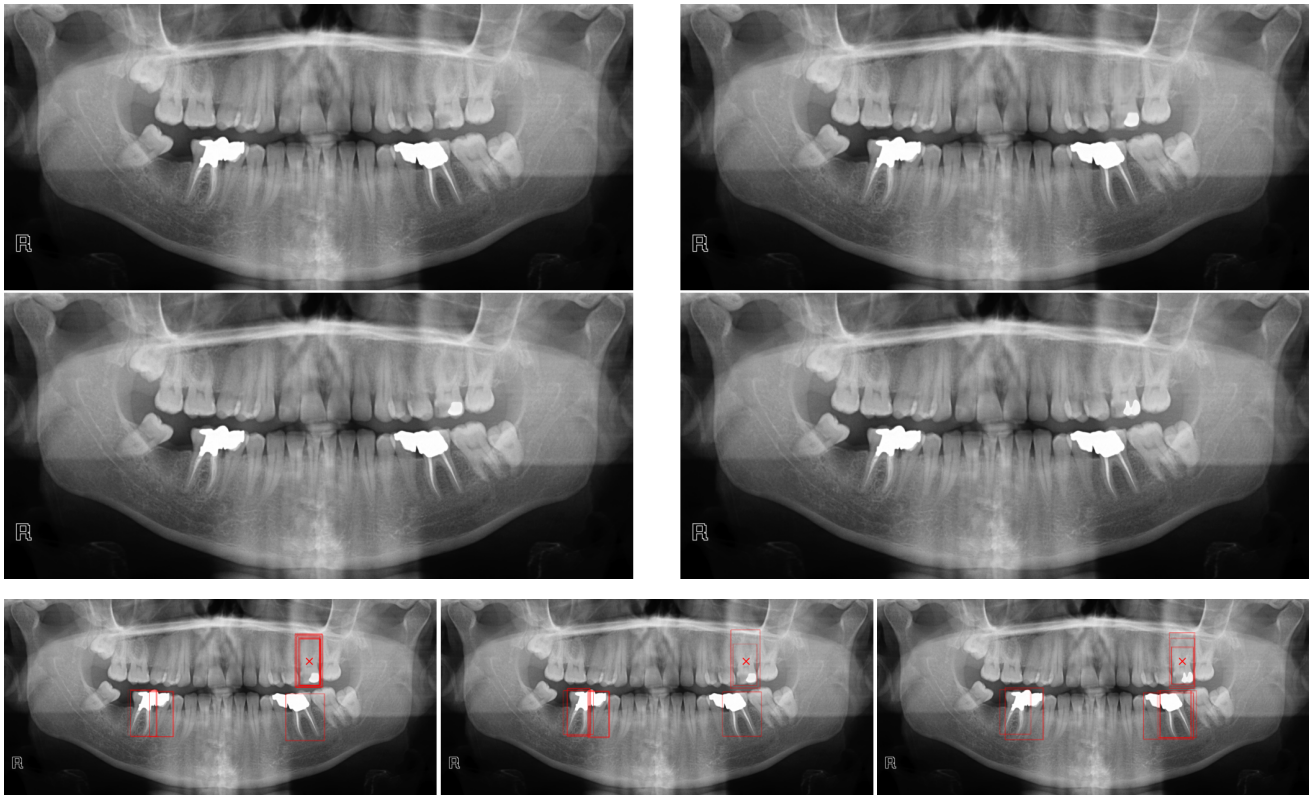
Приклад 4:



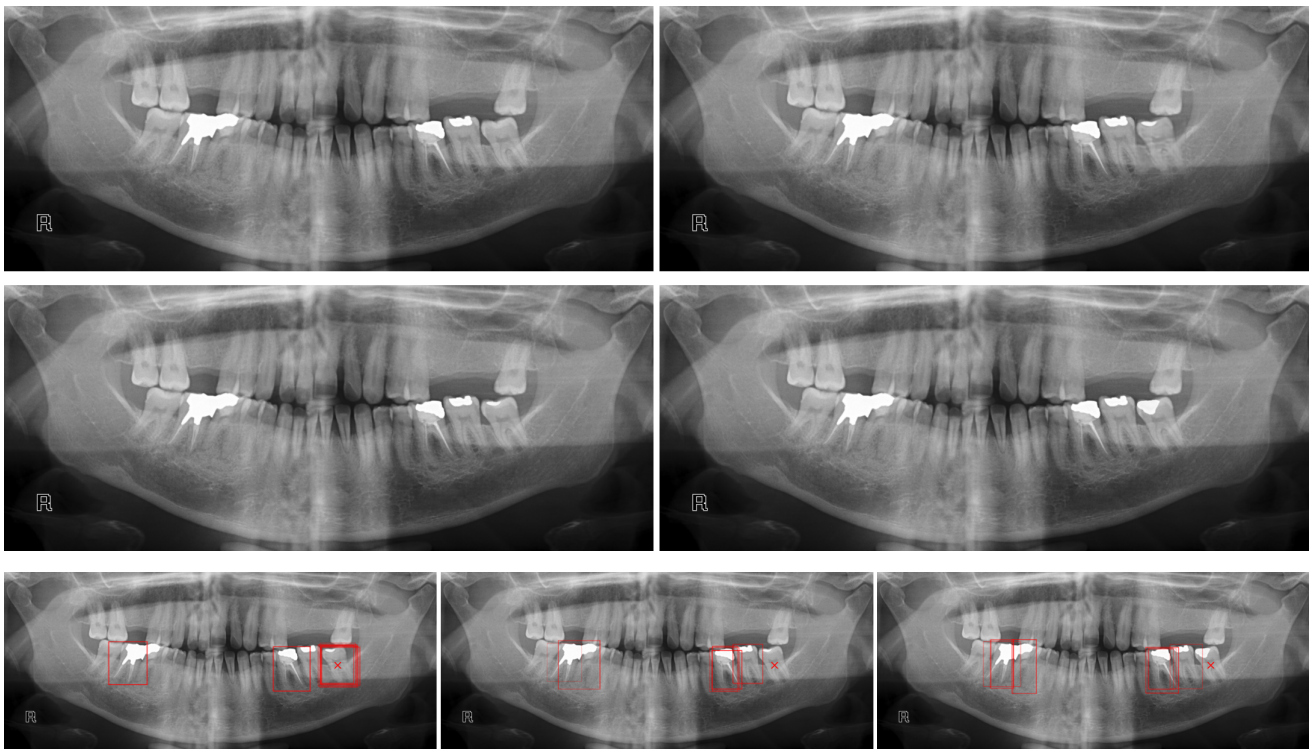
Приклад 5:



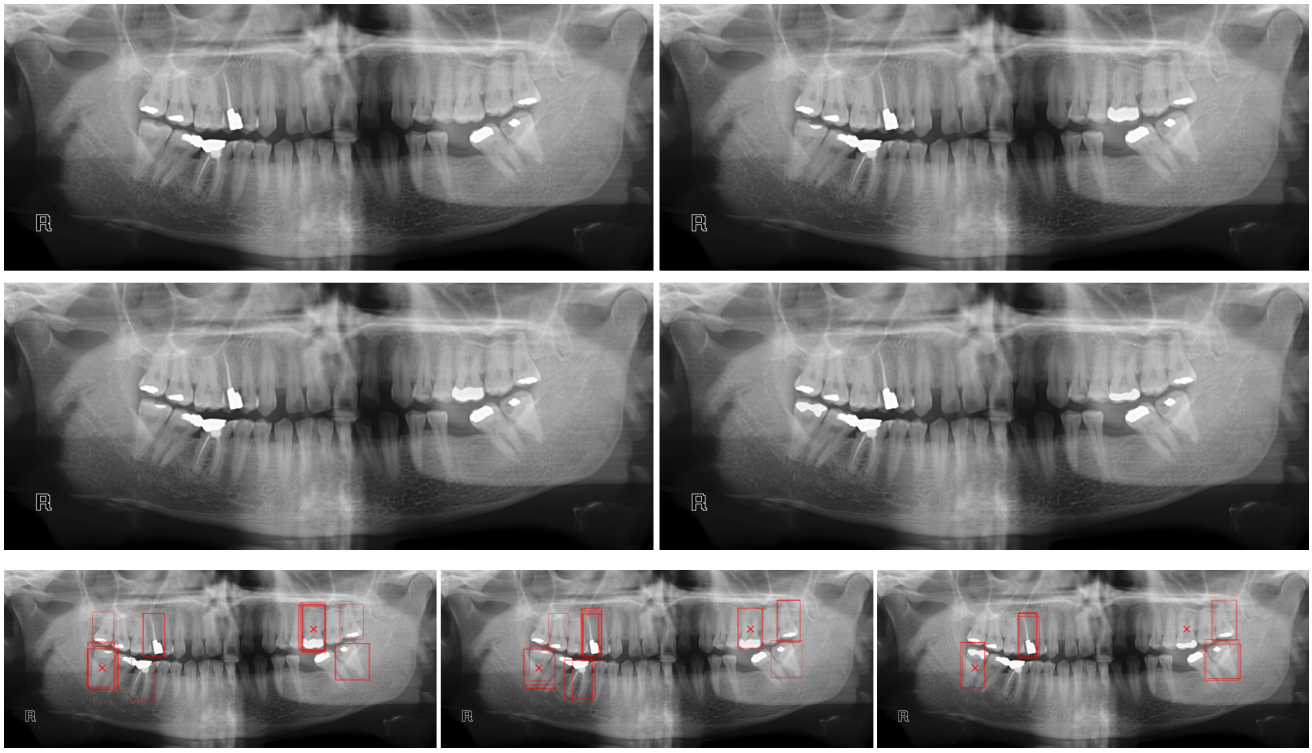
Приклад 6:



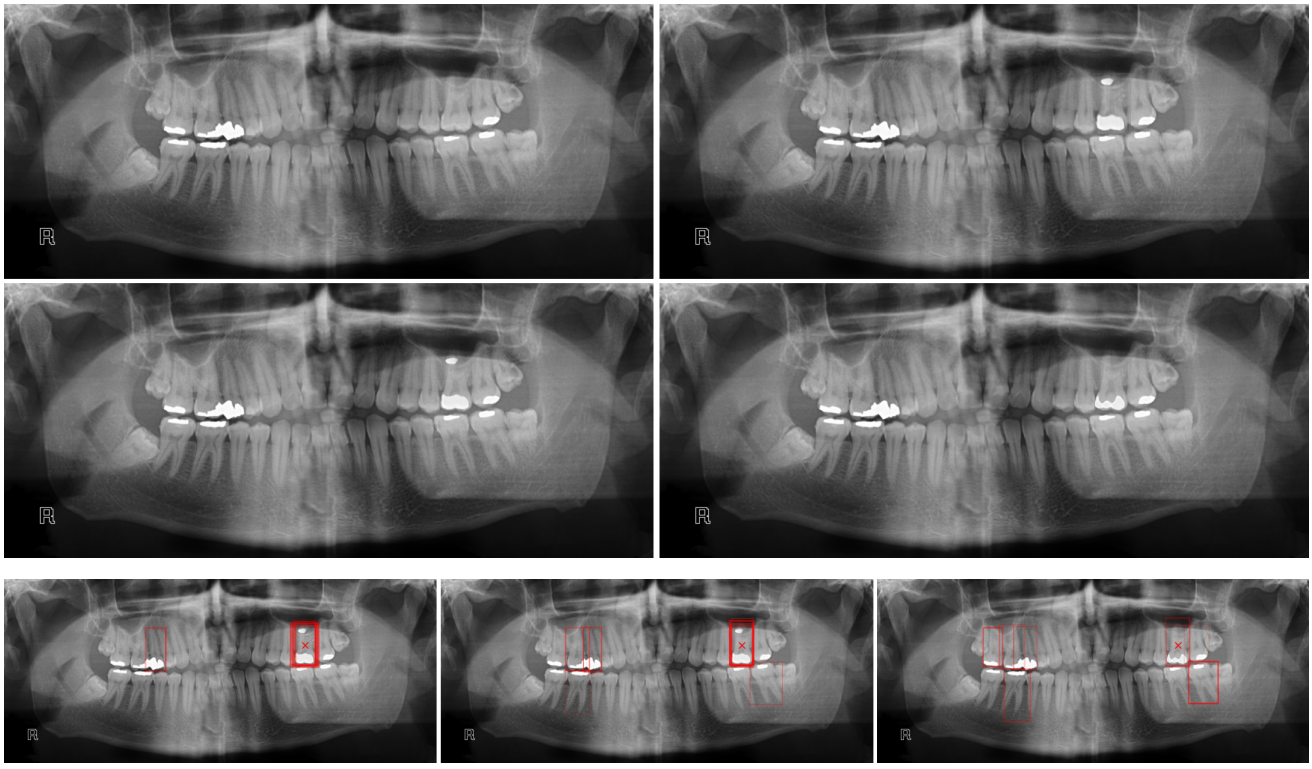
Приклад 7:



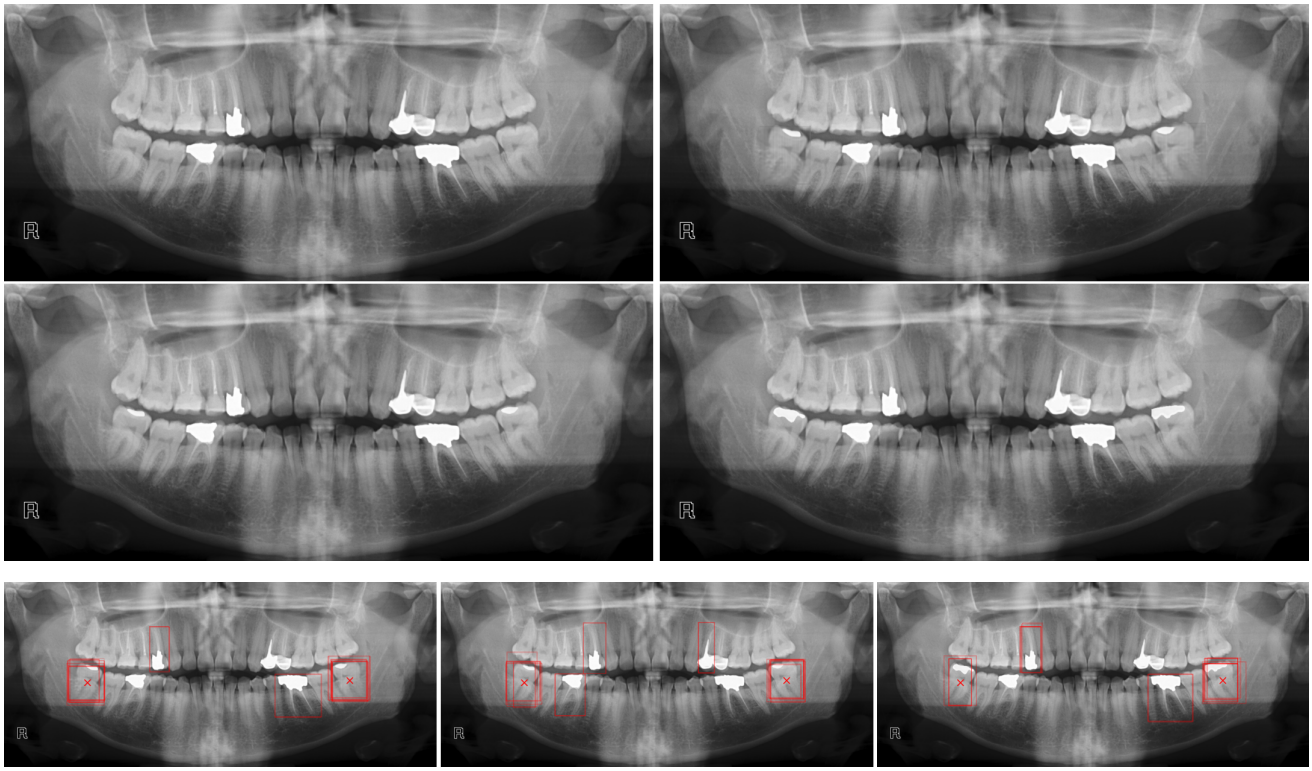
Приклад 8:



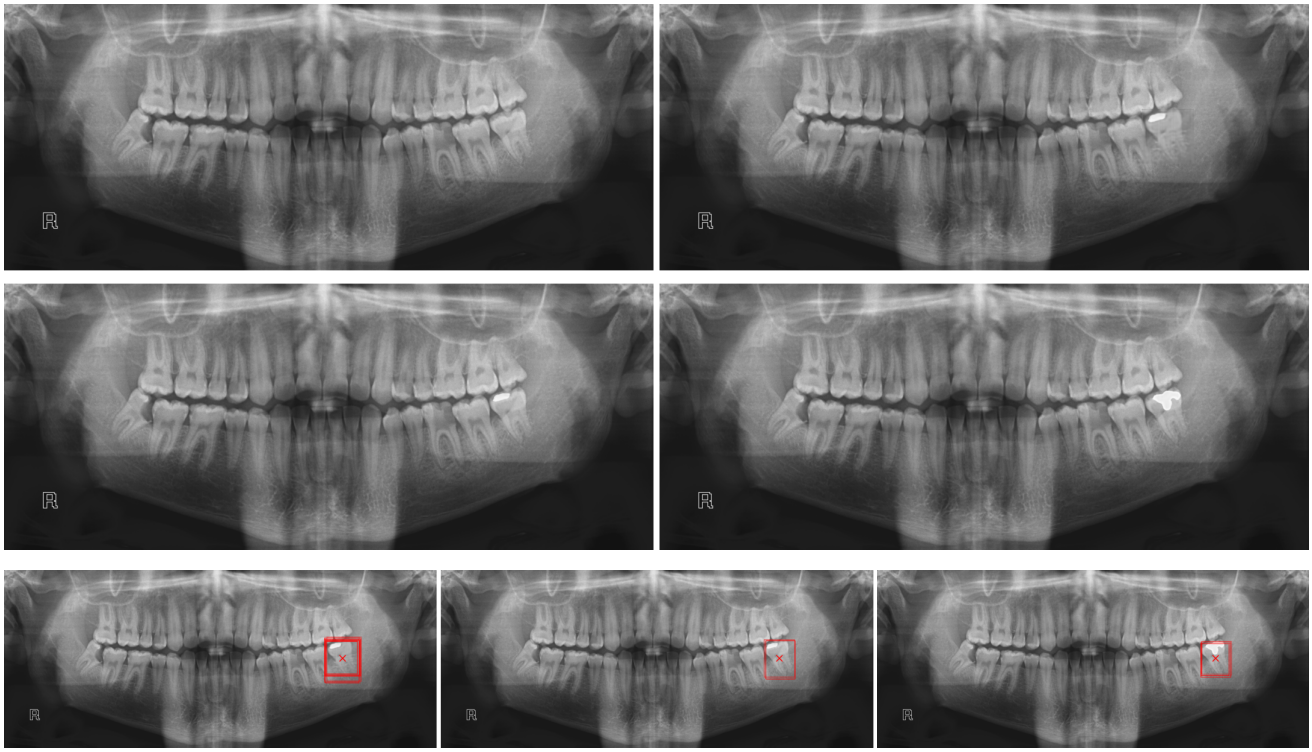
Приклад 9:



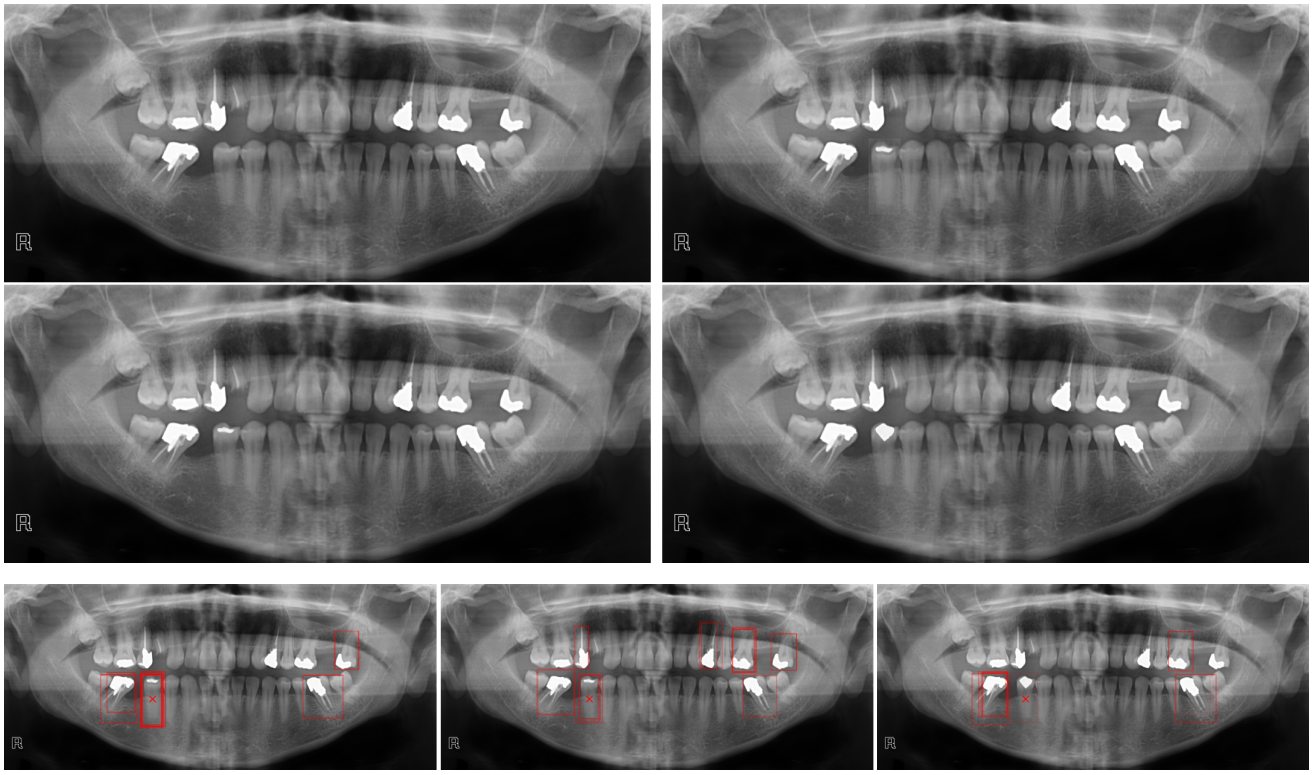
Приклад 10:



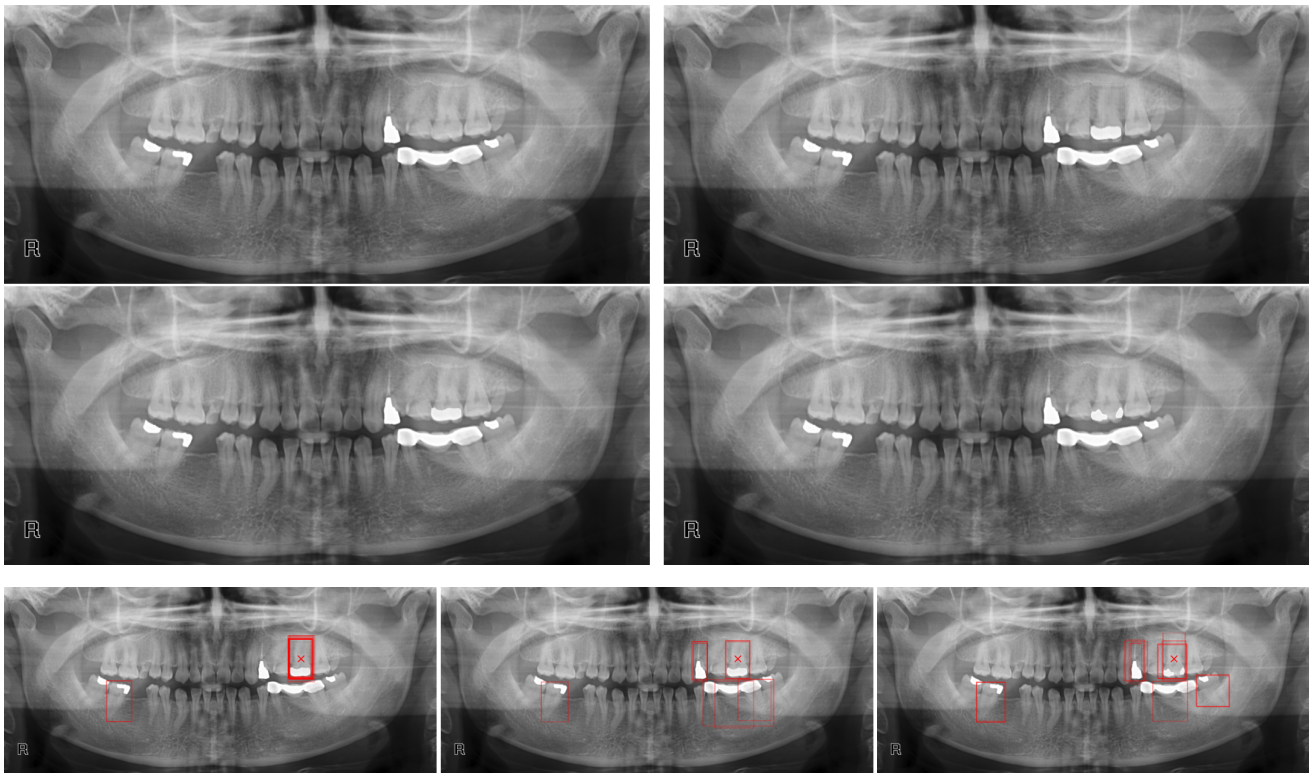
Приклад 11:



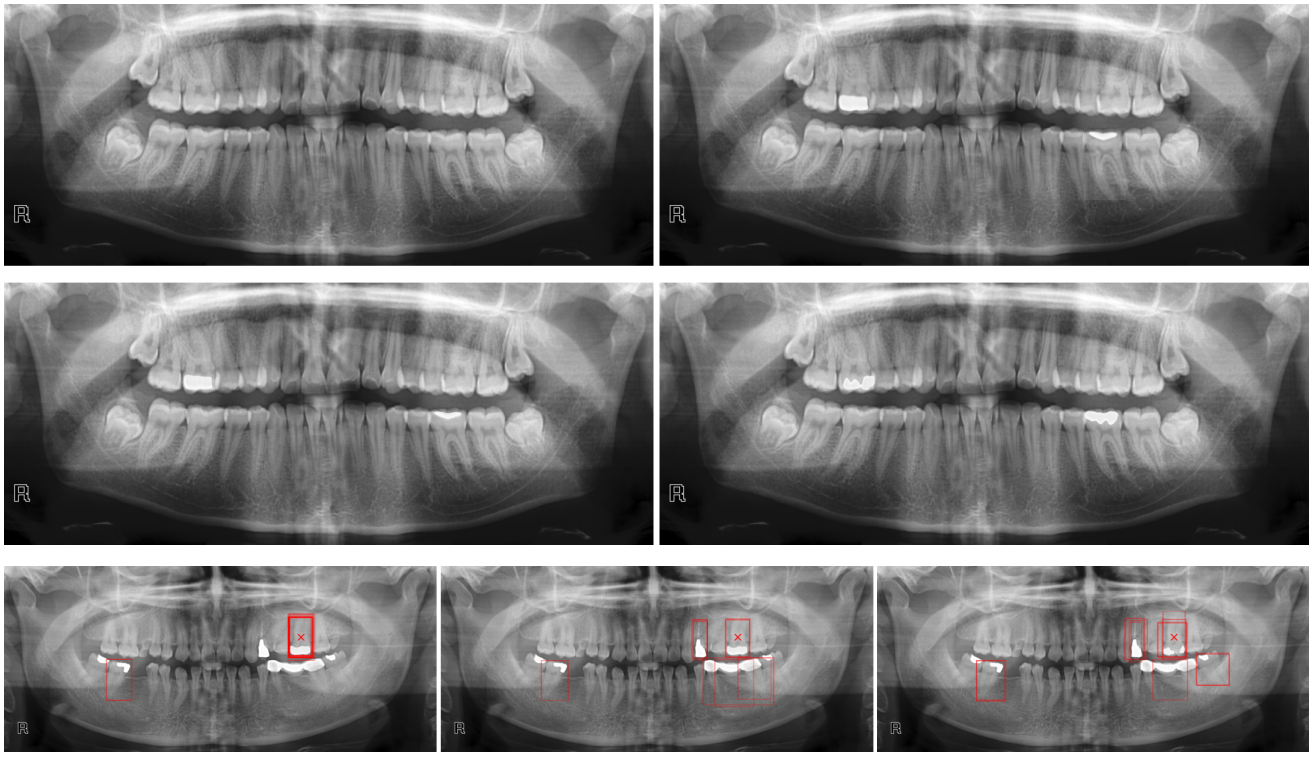
Приклад 12:



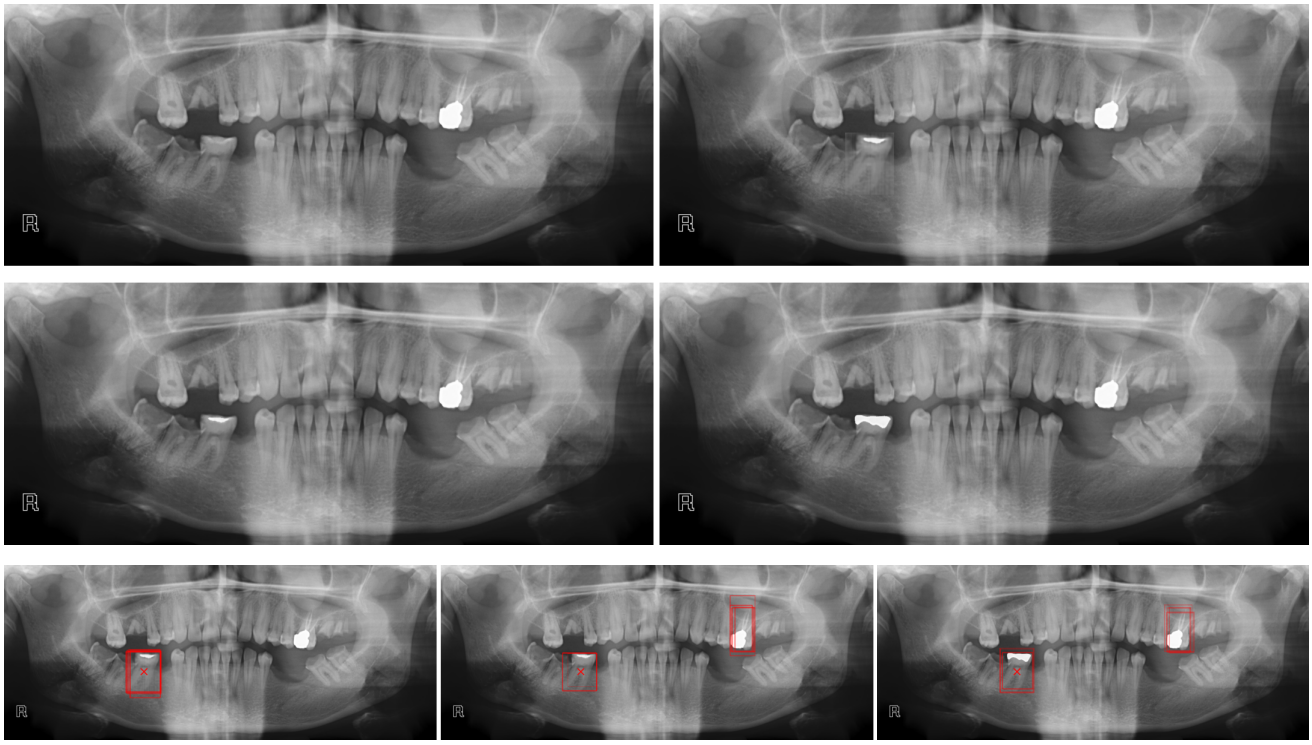
Приклад 13:



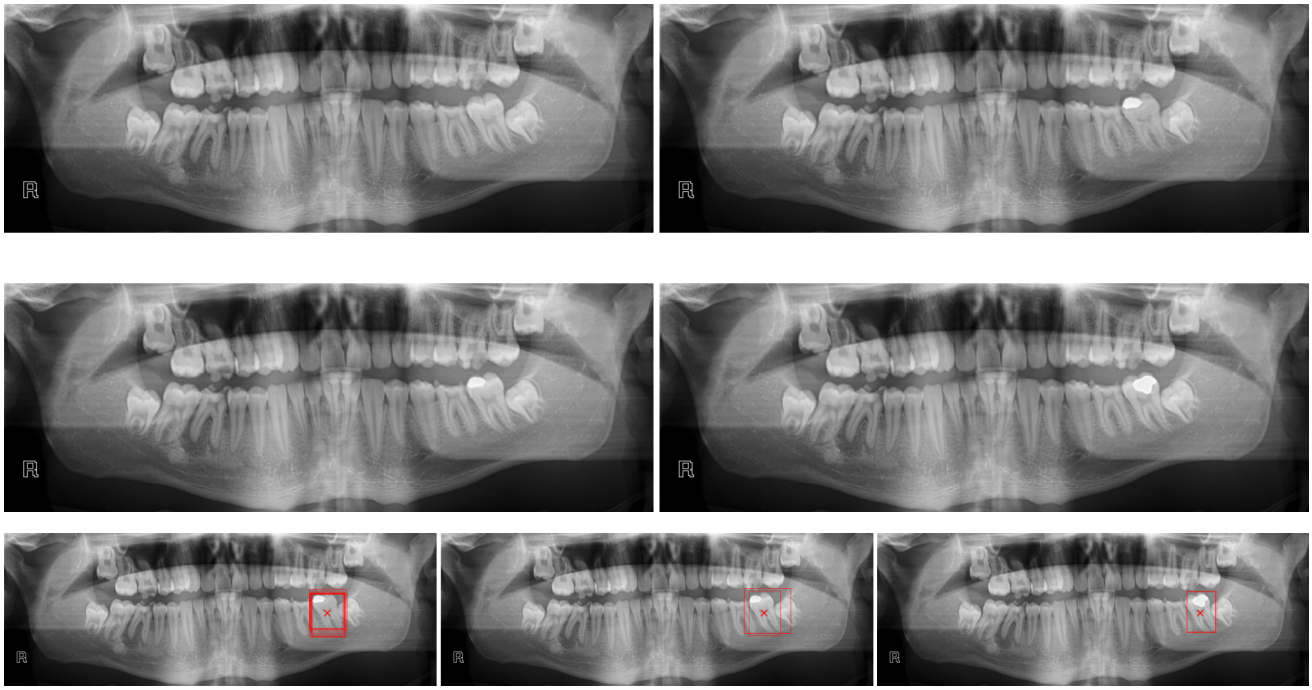
Приклад 14:



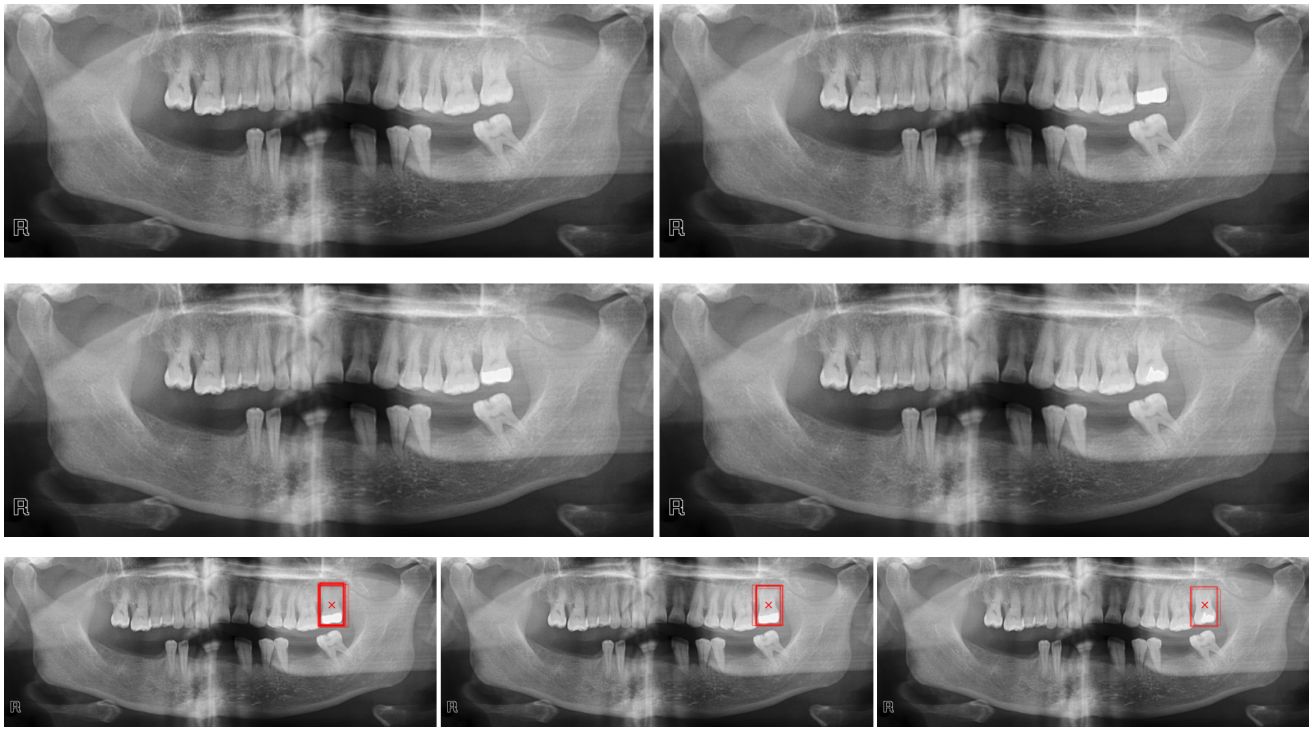
Приклад 15:



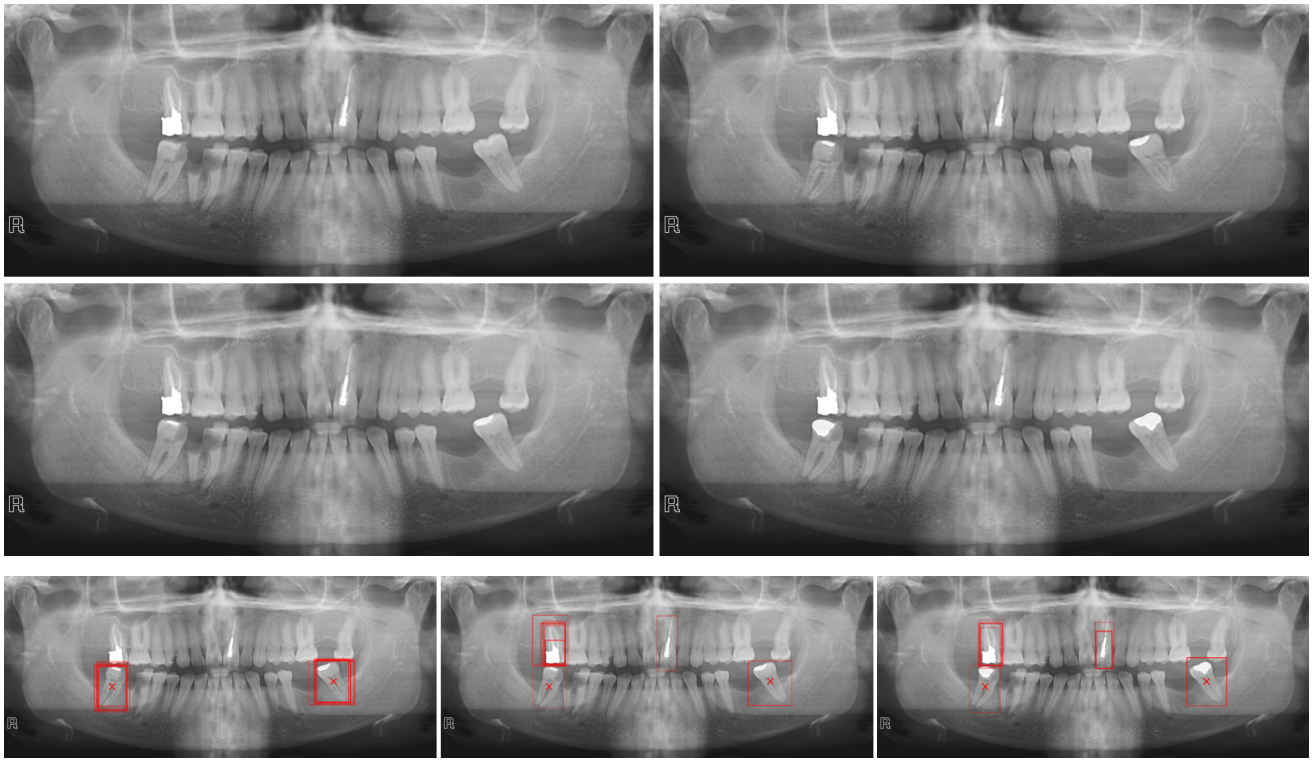
Приклад 16:



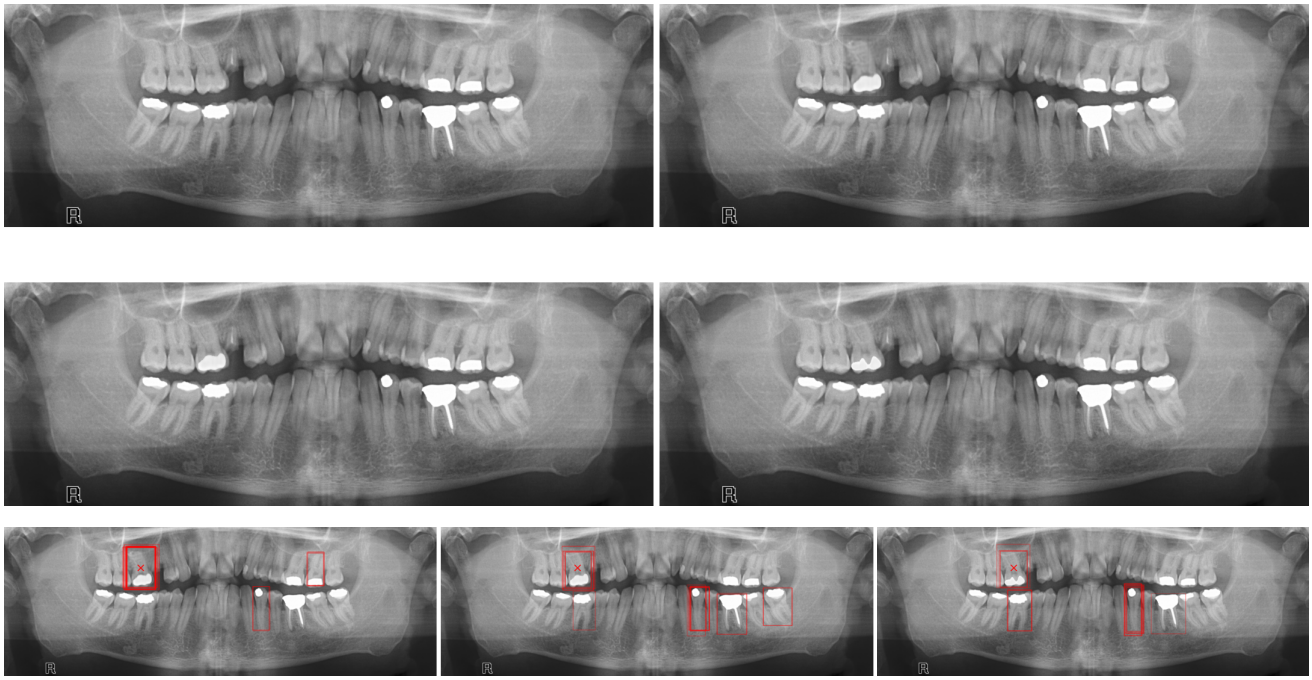
Приклад 17:



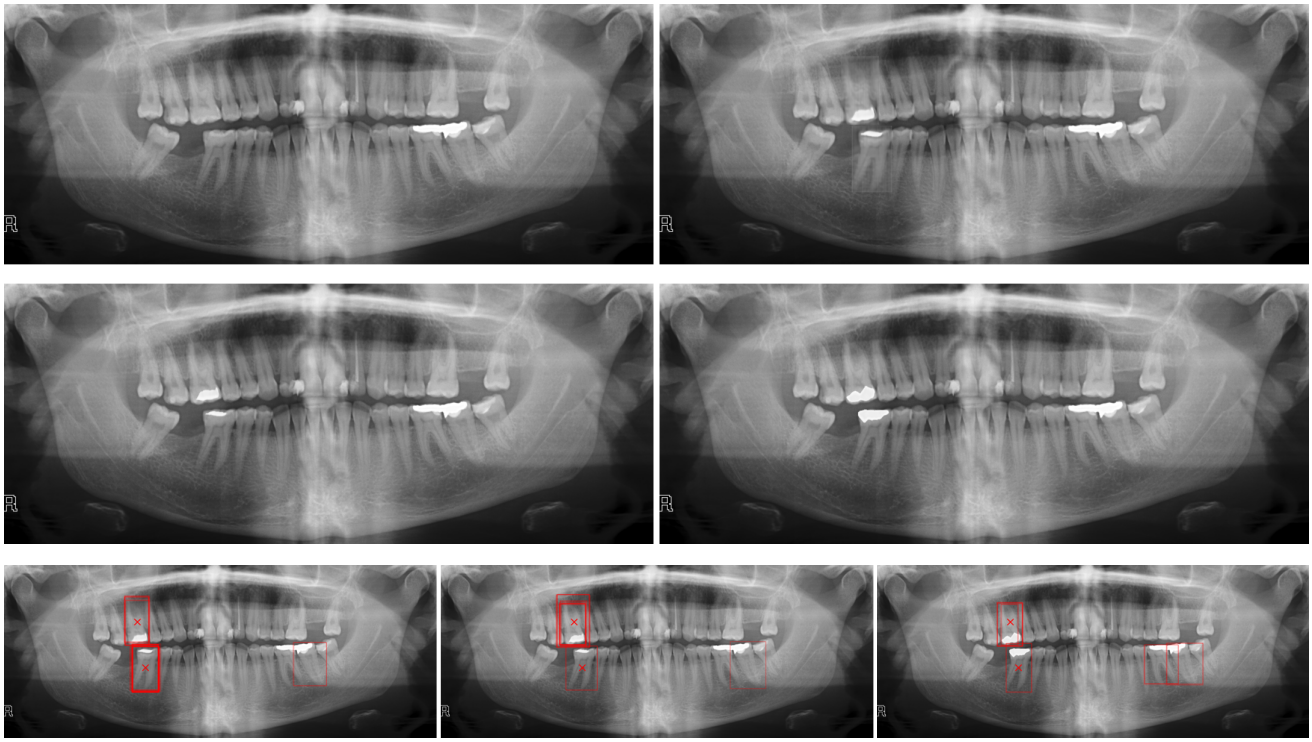
Приклад 18:



Приклад 19:



Приклад 20:



ДОДАТОК Б
ТАБЛИЦЯ ОЦІНОК ГЕНЕРАТИВНИХ МОДЕЛЕЙ ДЛЯ КОЖНОГО ЗОБРАЖЕННЯ

	F1			Влучність			Повнота		
Зображення	Мод. 1	Мод. 2	Мод. 3	Мод. 1	Мод. 2	Мод. 3	Мод. 1	Мод. 2	Мод. 3
1	0.800	0.167	0.033	0.800	0.167	0.033	0.800	0.167	0.033
2	0.867	0.200	0.500	0.867	0.200	0.500	0.867	0.200	0.500
3	0.640	0.111	0.244	0.644	0.167	0.333	0.667	0.083	0.200
4	0.367	0.367	0.067	0.367	0.367	0.067	0.367	0.367	0.067
5	0.900	0.206	0.133	0.900	0.250	0.133	0.900	0.183	0.133
6	0.633	0.067	0.167	0.633	0.067	0.167	0.633	0.067	0.167
7	0.800	0.000	0.000	0.800	0.000	0.000	0.800	0.000	0.000
8	0.444	0.200	0.000	0.500	0.200	0.000	0.417	0.200	0.000
9	0.878	0.644	0.200	0.867	0.567	0.200	0.900	0.800	0.200
10	0.550	0.518	0.300	0.550	0.733	0.300	0.550	0.417	0.300
11	1.000	0.233	0.267	1.000	0.233	0.267	1.000	0.233	0.267
12	0.733	0.267	0.022	0.683	0.267	0.017	0.833	0.267	0.033
13	0.956	0.233	0.133	0.933	0.233	0.100	1.000	0.233	0.200
14	0.867	0.417	0.444	0.867	0.583	0.633	0.867	0.333	0.350
15	0.800	0.200	0.333	0.800	0.200	0.333	0.800	0.200	0.333
16	1.000	0.133	0.200	1.000	0.133	0.200	1.000	0.133	0.200
17	1.000	0.733	0.333	1.000	0.733	0.333	1.000	0.733	0.333
18	0.711	0.039	0.167	0.833	0.050	0.233	0.650	0.033	0.133
19	0.822	0.356	0.200	0.767	0.333	0.183	0.933	0.400	0.233
20	0.756	0.400	0.244	0.900	0.600	0.367	0.683	0.300	0.183

ДОДАТОК В
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковані основні наукові результати дисертації:

[1] Я. О. Ісаєнков і О. Б. Мокін, "Аналіз генеративних моделей глибокого навчання та особливостей їх реалізації на прикладі WGAN," *Вісник Вінницького політехнічного інституту*, том 160, вип. 1, с. 82–94, Берез. 2022. doi: 10.31649/1997-9266-2022-160-1-82-94. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

[2] O. Bychkov, L. Zubyk, D. Gololobov, Y. Isaienkov, G. Grynkevych, and A. Ivanytska, "The System for Recognizing Useful Information of the Client's ID-Card Based on Machine Learning Technologies," *CEUR Workshop Proceedings,* vol. 2845, pp. 159–169, Dec. 2023. (Входить до Scopus)

[3] Я. О. Ісаєнков і О. Б. Мокін, "Трансформація цільового класу для задачі сегментації з використанням U-GAN," *Вісник Вінницького політехнічного інституту*, том 172, вип. 1, с. 81–87, Лют. 2024. doi:10.31649/1997-9266-2024-172-1-81-87. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

[4] Я. О. Ісаєнков і О. Б. Мокін, "Самовалідований U-GAN для трансформації цільового класу в задачах сегментації," *Вісник Вінницького політехнічного інституту*, том 174, вип. 3, с. 102-111, Черв. 2024. doi: 10.31649/1997-9266-2024-174-3-102-111. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

[5] Я. О. Ісаєнков і О. Б. Мокін, "Метод оцінки частково згенерованих даних," *Наука, технології, інновації*, том 31, вип. 3, с. 114-126, Вер. 2024. doi: 10.35668/2520-6524-2024-3-13. (Index Copernicus, Наукове фахове видання України, категорія «Б» зі спеціальності 126)

Наукові праці, які засвідчують апробацію матеріалів дисертації:

[6] Я. Ісаєнков і О. Мокін, "Проблема оцінювання генеративних змагальних мереж," *у Матеріалах L Науково-технічної конференції підрозділів Вінницького національного технічного університету*, 2021, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2021/paper/view/12468.

[7] Я. Ісаєнков і О. Мокін, "Артефакт межі у генеративних змагальних мережах," у *Матеріалах LI Науково-технічної конференції підрозділів Вінницького національного технічного університету*, 2022, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/16039.

[8] Я. Ісаєнков, "Використання методів глибокого навчання для оцінки якості медичних записок пацієнтів," *у Матеріалах конференції Проблеми вищої математичної освіти: виклики*, 2022, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/pmovc/pmovc22/paper/view/15558.

[9] Я. Ісаєнков і О. "Мокін О, "Вплив гіперпараметрів дифузійних моделей на якість генерування," у *Матеріалах LII Науково-технічної конференції підрозділів Вінницького національного технічного університету*, 2023, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/17174.

[10] Я. Ісаєнков, О. Мокін, О. Хорошева, і А. Герасимович, "Розробка застосунку для часткової аугментації зображень на основі генеративних моделей," у *Матеріалах LIII Науково-технічної конференції підрозділів Вінницького національного технічного університету,* 2024, Вінницький національний технічний університет. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20614.

ДОДАТОК Г

АКТ ПРО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ

ЗАТВЕРДЖУЮ
Директор ТОВ «ОРНЕТІКА»
_____ Олена САМАРСЬКА
«6» вересня 2024 р.

АКТ
про впровадження результатів дисертаційної роботи
«Інформаційна технологія на основі самовалідованої генеративної змагальної мережі»
Ісаєнкова Ярослава Олександровича

Цей акт підтверджує, що результати дисертаційної роботи на здобуття наукового ступеня доктора філософії Ісаєнкова Я. О., рекомендовані до впровадження в практичну діяльність при розробці інтелектуальних систем для вирішення задач комп'ютерного зору. Зокрема цінними є наступні отримані результати дослідження Ісаєнкова Я. О:

- Генерування синтетичних зразків даних КТ знімків зубів з використанням самовалідованих генеративних змагальних мереж. Для тренування моделей машинного та глибокого навчання частою є проблема отримання достатньої кількості реальних даних, особливо з ознаками цільового класу на них. Ще більше це ускладняється обмеженими в використанні із-за питань анонімності зразками. Генеративні змагальні мережі допомагають в генерації синтетичних даних, які імітують реальні знімки, з метою збагачення зразками для навчання.
- Використання методу оцінки частково згенерованих даних для ситуацій коли необхідне використання частково згенерованих даних. Цей підхід дозволить більш якісно підійти до проблеми відбору кращих моделей/зразків даних для створення нових згенерованих наборів даних.
- Проектування системи часткового аугментування даних, що може дозволити більш ефективно та економніше створювати нові зразки даних для навчання моделей.
- Тренування сегментаційних моделей для вирішення задачі ідентифікації об'єктів на зображеннях.

Даний акт не є основою для проведення фінансових взаєморозрахунків.

Голова комісії _____ Олена САМАРСЬКА

Члени комісії _____ Андрій РЕМОВСЬКИЙ

_____ Костянтин ВОВК

ДОДАТОК Д

АКТ ПРО ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ У НАВЧАЛЬНИЙ ПРОЦЕС ВНТУ

ЗАТВЕРДЖУЮ
Проректор з науково-педагогічної роботи та організації освітнього процесу Вінницького національного технічного університету

_____ доц. Олександр ПЕТРОВ
«19» вересня 2024 р.

АКТ
про впровадження результатів дисертаційної роботи
Ісаєнкова Ярослава Олександровича
«Інформаційна технологія на основі самовалідованої генеративної змагальної мережі», представленої до захисту на здобуття наукового ступеня доктора філософії зі спеціальності 126 Інформаційні системи та технології, в навчальному процесі

Комісія Вінницького національного технічного університету у складі:

голова комісії – декан Факультету інтелектуальних інформаційних технологій та автоматизації (ФІІТА), к.т.н, доцент Володимир СЕВАСТЬЯНОВ та члени комісії – заступник декана з навчально-методичної роботи ФІІТА, ст. викладач каф. системного аналізу та інформаційних технологій (САІТ) Василь ПРИСЯЖНЮК, заступник завідувача кафедри САІТ ФІІТА, к.т.н., доцент Євгеній КРИЖАНОВСЬКИЙ, склала цей акт про підтвердження того, що дійсно у Вінницькому національному технічному університеті під час викладання освітніх компонентів «Інтернет речей та інтелектуальний аналіз даних» та «Інформаційні технології моніторингу та аналізу стану складних систем» здобувачам, які навчаються за освітньо-науковою програмою «Інформаційні системи та технології» третього (освітньо-наукового) рівня вищої освіти зі спеціальності 126 Інформаційні системи та технології, впроваджено результати дисертаційної роботи на здобуття доктора філософії аспіранта кафедри САІТ Ісаєнкова Ярослава Олександровича у ряді лекцій та практичних занять з питань інформаційних технологій.

При викладанні цих дисциплін використовуються такі результати досліджень, отримані у дисертаційній роботі Ісаєнкова Я. О.:

- підготовка набору даних для навчання генеративних моделей: вибір потрібного набору даних, первинний аналіз даних, розмітка, візуалізація розподілів, огляд сучасних основних Python-бібліотек для вирішення цих задач (cv2, albumentations, matplotlib, seaborn, pandas);
- тренування та використання моделей генерування, таких як autoencoder, variational autoencoder (VAE), Deep Convolutional Generative Adversarial Networks (DCGAN), Wasserstein GAN (WGAN) та StyleGAN2, з використанням сучасних інструментів та фреймворків на мові програмування Python (Keras, PyTorch);
- дослідження методів оцінки результатів генерування як за допомогою ручних методів (HYPE), так і автоматичних (Inception Score, Fréchet Inception Distance).

Використання зазначених результатів дозволило підвищити якість навчального процесу із згаданих дисциплін.

Голова комісії _____ Володимир СЕВАСТЬЯНОВ

Члени комісії _____ Василь ПРИСЯЖНЮК

_____ Євгеній КРИЖАНОВСЬКИЙ