Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра математики

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: «**Генерація зображень відповідно тексту**»

Виконав: студент 4-го року навчання,
Освітньої програми «Прикладна математика», 113

Хоптій Андрій Володимирович

Керівник Крюкова Г. В., _____

кандидат фіз.-мат. наук, доцент

Рецензент _____
(прізвище та ініціали)

Кваліфікаційна робота захищена
з оцінкою _____

Секретар ЕК _____
«____» _____ 20____ р.

Київ – 2024

Зміст

Вступ

У двадцять першому столітті людство зробило настільки великий технологічний ривок, що, якихось п’ятдесят років тому буденні інструменти, які працюють на штучному інтелекту, що зараз нас оточують і допомагають з навчанням, роботою, відпочинком, розвагами і купую інших задач, вважались науковою фантастикою. Ще століття назад люди і уявити собі не могли, що те майбутнє з книжок, про яке писали фантасти, у якому можна зустріти автопілоти в машинах, роботів, які вміють ходити по будь яким ділянкам, інколи навіть краще за людину, голосові помічники в телефонах і смарт-годинниках або які слухають що ти їм кажеш і будують з тобою діалог, ніби це справжня людина, з музикою яка була написана не людиною або картинами намальованими не людьми, що це все з’явиться так швидко. І такий результат ми отримали завдяки швидкому розвитку галузі досліджень штучного інтелекту, яка називається “Машине навчання”.

Звісно, що створення повноцінного штучного інтелекту, який буде мати свою свідомість, свою унікальну особистість і загалом думати один в один як людина поки не є можливим, бо хоч і наявні моделі штучних інтелектів виглядають “розумними”, але вони просто натренованими на великому обсязі інформації і виконують свої задачі всліпу, не розуміючи семантики питання яке перед ними поставлено і проблеми, що вони вирішують. І якщо з такими речами, як розпізнавання об’єктів, ходіння і навіть спілкуванням з реальними людьми у штучного інтелекту проблем немає, то з темою генерування зображень по текстовому запиту якраз проблема браку семантики в діях дуже добре відчувається, що призводить до інтересу дослідників до саме цієї теми і активному дослідженню наявних методів реалізації задля можливого покращення результату і пошуку абсолютно нових унікальних рішень.

Актуальність теми. Актуальність теми є дуже високою, бо вирішення проблеми в артефактах, при генеруванні зображень наблизить людство до

створення повністю свідомого штучного інтелекту, найпопулярнішим прикладом який тільки можна привести, то це при генеруванні зображення людини, часто можна помітити проблему з пальцями і очами, а точніше в їх формі і кількості. Ці випадки з шістьома пальцями або трьома очима трапляються через брак семантики, тому що різниця в розуміння слова “людина” від слова “яблуко” для штучного інтелекту буде лише в наборі певних характеристик, як от форма і колір, а не в тому, що яблуко це фрукт, який росте на дереві, а людина це живий організм, який його створив.

Мета дослідження. Метою цієї роботи буде розглянути різні підходи для вирішення задачі з генеруванні зображень по текстовому запиту, також будуть розглянути сильні і слабкі сторони кожного із них, вивчена проблематика і місця застосування цих підходів, а також буде обрано одну з моделей над якою буде вестись робота над її покращенням результатів після тесту в спеціалізованих для цього бенчмарках. Далі в роботі буде розглянуто чотири найпопулярніших моделей, на основі яких будуються всі сучасні генератори зображень. В наступних розділах будуть представлені GAN, CLIP, VQGAN і VAE.

Перелік умовних позначень, символів, одиниць вимірювання, скорочень

1) **GAN** (Generative Adversarial Network) — це тип архітектури нейронної мережі, який використовується для генерування нових даних, схожих на навчальний набір. Мережа складається з двох частин: генератора, який створює зразки, та дискримінатора, який оцінює їх на предмет схожості до реальних даних. Вони "змагаються" один з одним: генератор намагається обдурити дискримінатор, виробляючи все кращі імітації, а дискримінатор навчається краще визначати справжні дані від фальшивих.
2) **CLIP** (Contrastive Language–Image Pre-training) — це модель, розроблена OpenAI, яка використовується для зв'язування текстових та візуальних даних через контрастне навчання. Модель тренується на величезному наборі текстових описів та відповідних зображень, щоб вивчити загальні візуальні концепти від різноманітних джерел інформації. Завдяки цьому CLIP може ефективно ідентифікувати і класифікувати об'єкти на зображеннях, використовуючи природну мову як засіб запиту.
3) **VQGAN** (Vector Quantized Generative Adversarial Network) — це вид генеративної мережі, який поєднує ідеї векторного квантування з архітектурою GAN для створення високоякісних зображень. Модель використовує векторне квантування для стиснення зображень у просторі з низькою розмірністю, що дозволяє ефективно управляти великими наборами даних та зберігати більш точні деталі зображень. VQGAN зазвичай використовується у задачах, де потрібне генерування візуально привабливих і деталізованих зображень.
4) **VAE** (Variational Autoencoder) — це тип нейронної мережі, який використовується для генерування нових даних та зниження розмірності даних. Вона складається з двох основних частин: енкодера, який перетворює вхідні дані в набір параметрів у прихованому просторі, та декодера, який

використовує ці параметри для відтворення вхідних даних. Центральною особливістю VAE є її здатність управляти ступенем варіації у генерованих даних, що робить її корисною для задач, пов'язаних з моделюванням розподілів даних та нечітким генеруванням.

5) **PAG** (Perceptually Aligned Gradients) — це метод оптимізації, який використовується у генеративно-змагальних мережах (GAN) для покращення візуальної якості зображень. Він враховує сприйняття людським оком, зменшуючи візуальні артефакти та досягаючи більш природних результатів у генерованих зображеннях.
6) **PGD** (Projected Gradient Descent) — це метод оптимізації, використовуваний для знаходження мінімальних або максимальних значень функції в обмеженому просторі. PGD виконує ітеративні кроки у напрямку антиградієнта функції, що означає рух проти напрямку найшвидшого зростання функції, після чого відбувається "проектування" (виправлення) цих кроків назад до дозволеної області. Цей метод широко застосовується в машинному навчанні для забезпечення того, щоб рішення залишалися в межах певних обмежень, і вважається ефективним засобом для підвищення стійкості моделей до атак з прикладами зі зміною.

Анотація

У цій роботі розглянуто розвиток технологій машинного навчання, починаючи з перших чатботів та систем розпізнавання зображень, до сучасних моделей генерації зображень. Описано чотири основні моделі: генеративні змагальні мережі (GAN), асоціативна модель CLIP, VQGAN та варіаційний автоенкодер (VAE). Розглянуто технічні особливості кожної з моделей, їх проблематику та сфери застосування. І в фінальній частині роботи запропоновано та протестовано нову модель CLIPPAG, яка поєднує методи перцептивно вирівняних градієнтів для покращення візуальної якості зображень. Описано процес тренування моделі на датасеті SBUcaptions. Проведено порівняння результатів генерації зображень різними моделями, підтверджено покращення якості і деталізації зображень при використанні моделі CLIPPAG.

Висновки підтверджують, що використання перцептивно вирівняних градієнтів під час тренування моделі CLIP значно покращує результати генерації зображень, забезпечуючи кращу передачу деталей та зменшення кількості артефактів. Робота демонструє перспективність подальших досліджень та вдосконалень у сфері генерації зображень на основі текстових запитів.

Ключові слова: машинне навчання, штучний інтелект, генеративні змагальні мережі, CLIP, VQGAN, VAE, перцептивно вирівняні градієнти, генерація зображень.

РОЗДІЛ 1. Заснування галузі Машинного навчання і його подальший розвиток

1.1 Історичний контекст і розвиток технологій

Сьогодні кожна дитина хоча б трошки чула про нейроні мережі, про ChatGPT, Dall-E, Midjourney та інші. Новини і самі додатки масово почали розповсюджуватись не так давно.

У 2022 році, а саме: 22 листопада, світу був представлений у загальний доступ ChatGPT і після нього активно почались з'являтись інші нейромережі для більш специфічних сфер: дизайн, програмування, тощо, тому може здаватись, що цей напрям тільки-тільки з'явився, але це не так. Розвиток Machine Learning (укр. Машине Навчання), який якраз відноситься до машинного навчання і є галуззю досліджень його, розпочав свій розвиток у 1950их, після розроблення тесту Тюрінга, який перевіряв машину на здатність появляти поведінку аналогічно до інтелектуальної поведінки людини. Тюрінг, Семюел, Маккарті, Мінський, Едмондс і Ньюелл – це люди які перші ввели термінологію "штучний інтелект" і "машине навчання"

1.2 Перший чатбот

Першим в історії чатботом була машина Єліза і вона була створена у далекому 1967 році в лабораторії штучного інтелекту Массачусетського технологічного інституту Джозефом Вайзенбаумом. Використовуючи метод аналізу pattern-mathing, чатбот відповідав користувачу шаблонними фразами, які викликали у користувачів відчуття розмови з реальною людиною.

Після цього багато інших програмістів почали додавати нові скрипти і робили цей чатбот більш вузько направлений. Одним із найвідомішим є версія DOCTOR, яка імітувала справжнього психотерапевта. Вона застосовували такі підходи як "активне слухання" і "перенаправлення питань", спонукаючи людину більше

говорити і міркувати про свої проблеми. Коли людина писала роботу, що вона сьогодні сумна, чатбот питав в чому полягає її сум і що його викликало сьогодні, тим самим направляючи людину до самостійного пошуку розв'язання її проблеми.

1.3 Перша робота з зображеннями

Перші спроби роботи з розпізнаванням зображень почались у 1960их – 1970их роках. Вони були доволі примітивними, так як використовували прості алгоритми і евристичні методи для розпізнавання символів.

Але вже у кінці 1980их і на початку1990их роках компанія IBM створила Multi-line Optical Character Reader (MLOCR) для US Postal Service. Ця машина виконувала задачу з розпізнаванням поштових індексів на конвертах для автоматизації сортування пошти для подальшої її відправки в потрібні адресантам місця. Основа роботи полягали в застосуванні оптичного розпізнавання символів (OCR).

Сам процес розпізнавання відбувався в 4 етапи:

1) **Сканування** – на спеціальних конвеєрах стояли сканери які зчитували зображення і передавали отримане зображення далі в систему
2) **Попередня обробка –** зображення оброблювались спеціальними алгоритмами які прибирали зайві шуми і підвищували якість зображення
3) **Аналіз і розпізнавання –** для розпізнавання були використані згорткові нейроні мережі (CNN), завдяки своєї особливості у витяганні локальних ознак і візерунків CNN показав хорошу ефективність і доцільних використання у розпізнавані рукописних зображень. На цьому етапі алгоритми виділяли окремі символі і правильно задавили відповідності до вже існуючих шаблонів

4) **Сортування –** після розпізнавання індексу на конверті він відправлявся конвеєром у відповідному напрямку, чим значно зменшив час на обробку пошти і навіть зменшив кількість помилок при сортуванні

Система MLOCR є одним із першим прикладів застосування штучного інтелекту і машинного навчання у бізнес процесах, що своїм прикладом надихнула на подальший розвиток цієї сфери програмування і привернення багато інвестицій для вивчення і аналізу.

1.4 Перші text-to-image генератори

Не зважаючи на те, що розвиток Машинного навчання розпочалось більше як 70 років тому, як було зазначено вище, лише в 2021 перший text-to-image генератор набув широкої популярності і це був DALL-E від лабораторії досліджень штучного інтелекту OpenAi. Це була справжня революція в області штучного інтелекту та генерації зображень. У 2017 році була представлена нова трансформерна архітектура, яку використовував ChatGPT-3 і на базі якого і працює DALL-E.

Вражаючою особливістю в згенерованих результатах від DALL-E була здатність працювати в різних стилях, в тому числі абстрактних і футуристичних. Ця особливість навіть підкреслена в самій назві, бо назва складається з так званої ігри слів двох імен – відомого іспанського художника і архітектора Сальводара Далі, який якраз і працював і стилі сюрреалізм і відомого анімаційного персонажа WALL-E з однойменного науко-фантастичного фільму від Pixar який був випущен у 2008 році.

Як було зазначено в попередньому абзаці використовується архітектура трансформерів “під капотом”, але особливість від ChatGPT-3 в використанні величезного датасету під час навчання який складається з наборів пар тексту і зображень.

Звісно процес навчання такої модель потребує величезну кількість ресурсів, тренування моделі яка може працювати з текстом і зображеннями одночасно потребує використання кластерів з суперкомп'ютерів і звісно ж DALL-E не є виключенням. Лідером у забезпеченні графічних процесорів і суперкомп'ютерів створених на їх основі для таких задач є компанія NVIDIA. OpenAI використовував суперкомп'ютер Selene – це один з найпотужніших обчислювальних апаратів які зараз існують. Він побудований на найкращої лінійки відеокарт A100 і налічує в собі більше тисячі таких графічних процесорів в собі. Кількість ядер налічує в собі більше 500 тисяч, а максимальна потужність отримана потужність є майже 64 петафлопс, хоча теоретична сягає майже до 80.

В подальшій моїй роботі також було використано спеціальний суперкомп'ютер для тренування нової моделі, про це детальніше буде розписано в розділі 3.

Розділ 2. Детальний розбір моделей для генерації зображень, а також результати їх роботи

2.1 Загальнодоступні моделі для роботи з зображеннями

Звісно не тільки компанія OpenAi досліджує і представляє свої моделі і підходи для роботи з зображеннями в штучному інтелекті. На сьогоднішній день існує 4 найпопулярніші моделі і мережі для цього:

- Генеративно змагальна мережа GAN – створена у 2014 році Іаном Гудфеллоу та його колегами в університеті Монреаля і представлена у науковій статті “Generative Adversarial Nets”
- Модель CLIP – створена вищезгаданою компанією OpenAi у 2021 році і представлена у статті “Learning Transferable Visual Models From Natural Language Supervision”
- Модель VQGAN – створена у колаборації між Ганноверським університетом і філіалом Google Research у 2021 і представлена у науковій статті “Taming Transformers for High-Resolution Image Synthesis”
- Модель VAE – створена у Амстердамському університеті у 2013 році і представлена у науковій статті “ Auto-Encoding Variational Bayes”

В наступних підрозділах буде розглянуто всі технічні особливості, проблематики, сфери застосування кожної з моделей і подальша аргументація вибору моделі CLIP для створення нової кращої моделі з застосуванням перцептивно вирівняними градієнтами.

2.2 Генеративні адверсальні мережі (GAN), їх компоненти і проблематика

Генеративні адверсальні мережі відносяться до серйозних інноваційних досягнень у сфері машинного навчання. Вони складаються з двох компонентів: генератора і дискримінатора.

Генератор є нейромережою і відповідає за створення нових даних, які мають бути максимально реалістичними до заданого запиту. На вхід генератор отримує абсолютно випадковий шум, а на вихід повертає на результат його роботи.

Дискримінатор, який також є нейромережою, в свою чергу відповідає за класифікацію роботи генератора. Дискримінатор може повернути тільки 2 значення: зразок справжній або зразок був штучно згенерований.

Ці два компонента грають між собою в гру “злодій-поліцейський” і під час тренування кожен з них намагається покращити роботу своїх алгоритмів, щоб отримати перемогу над протилежним компонентом. Такий тип навчання називається адверсальне навчання і доволі часто застосовується і в інших нейромережах, але таке навчання також має свої проблеми.

По перше, балансу між генератором і дискримінантом дуже важко досягти, тому поширеною проблемою є нестабільність навчання, бо одна з мереж швидко почне помітно переважати іншу.

По друге, можливий випадок злиття градієнтів двох компонентів, у такому випадку подальші епохи навчання стають неефективними і генератор перестає навчатись

По трете, генератор може почати створювати обмежений набір зразків, який не буде відображати повного обсягу даних з датасету. Такий випадок називається mode collapse

І останньою загальною проблемою є потреба в великому і якісному датасеті, щоб забезпечити ефективне тренування моделі. Рекомендовано використовувати датасети які нараховують в собі десятки тисяч зразків, але для складних задач

розмір датасету зачасту нараховує сотні тисяч або навіть і мільйони зразків для навчання.

Підсумовуючи всі проблеми які були перераховані вище, варто зазначити, що GAN є все одно дуже перспективним і популярним рішенням, нові рішення і підходи покращення стабільності і продуктивність постійно з'являються. GAN знайшов сферу свого застосування у багатьох галузях:

- Комп'ютерний зір – генерація зображень, збільшення роздільної здатності, розфарбування чорно-білих зображень, перетворення стилів
- Синтаксична обробка – генерація тексту, переклад тексту, синтез мови
- Створення контенту – генерація музики, створення реалістичних спец ефектів, створення 3д моделей

2.3 Асоціативна модель CLIP, потенціал і застосування в різних сферах

Асоціативна модель CLIP є загальнодоступною моделлю від компанії OpenAI, вона є новаторською бо поєднує в собі обробку синтаксису і зображень. Ця модель також складається з двох компонентів: Текстовий енкодер і Візуальний енкодер

Текстовий енкодер потрібен для перетворення тексту на проекції векторів використовуючи трансформери, які не раз вже були згадані в роботах пов'язаних з компанією OpenAI. Таке перетворення на вектори потрібне моделі для розуміння змісту текстової частини і контексту про що іде мова.

Візуальний енкодер також перетворює зображення на проекції векторів. Для цього застосовується CNN, який також був вище згаданий у успішному прикладі застосування на US Postal Service для розпізнавання поштових індексів на конвертах.

За рахунок спільної нормалізації до векторів модель починає тренування з зв'язування пар тексту і зображень. В результаті отримується модель, яка має

широкий спектр використання у будь яких задачах пов’язаних з одночасною роботою з текстом і зображеннями. Це може як розпізнавання і класифікація зображень, так і генерування зображень відповідно до тексту. Варто зазначити, що окрім універсальності модель є дуже гнучкою і адаптивної, це випливає з того, що при додаванні нових даних до датасету або виконання інших задач немає загальної потреби з нуля перенавчати модель, достатньо буде просто довчити на новому датасеті. CLIP застосовується в схожих сферах, що і GAN:

- Комп’ютерний зір – генерація зображень, розпізнавання об’єктів, пошук об’єктів
- Синтаксична обробка – аналіз тексту разом з зображенням, для кращого розуміння контексту, автоматична генерація описів
- Створення контенту – створення відео по текстовому сценарію

2.4 Поєднання векторного квантування і генеративних адверсальних мереж (VQGAN), високоякісні зображення і словник векторів замість вивчення напрямку пікселів

Дослідження можливості покращення GAN призвели до створення нової моделі VQGAN яка поєднує в собі векторне квантування і генеративно адверсальні мережі. Ця модель замість звичайного аналізу напрямку пікселів застосовує словник векторів, що призводить до можливості генерування зображень з високою якістю. VQGAN складається з 3ьох компонентів: Векторного квантування, генеративних адверсальних мереж, ендокеда-декодера.

Векторне квантування створює словник у якому певний фрагмент зображення представлений як вектор. Мета навчання моделі полягає в навчені моделі оперувати цими векторами для створення нових зображень

Принцип роботи GAN було розглянуто в розділі 2.2. Єдиною відмінністю в цій моделі полягає в тому, що сам генератор при роботі використовує вектори з словника задля генерування екземплярів для змагання з дискримінатором

Енкодер і Декодер в цій моделі потрібні для перетворення зображень на словник векторів і після тренування для перетворення векторів у фрагменти зображень.

Ця модель має велику кількість переваг, а саме: висока якість зображень з збереженням правильних деталях в текстурах і структурах, використання векторів зменшує розмір моделі і кількість необхідних параметрів для зберігання, збільшує швидкість навчання і генерування зображень.

Поєднання векторного квантування і генеративних адверсальних мереж застосовується у таких сферах:

- Мистецтво і дизайн – генерація художніх картин, рекламних постерів, NFT
- Створення контенту - створення реалістичних текстур для 3д моделей, створення реалістичних відеоефектів і анімацій
- Комп’ютерний зір – генерування зображень відповідно до тексту

2.5 Варіаційний автоенкодер (VAE), розширення поточного датасету і байєровські методи

Варіаційний автоенкодер це інструмент який як правило застосовується в Машинному Навчанні для розширення навчального датасету, в тому числі для генерування зображень. Він поєднує в собі принципи глибинного навчання разом з байєровськими методами, такий метод дозволяє на основі початково зразку створити декілька подібних нових.

Складається варіаційний енкодер з трьох компонентів: енкодера, декодер і прихований простір.

Енкодер потрібен для перетворення вхідних даних на параметри розподілу, як правило це середнє і дисперсія. Ці параметри застосовуються у прихованому просторі

Прихований простір, використовуючи вище зазначенні параметри обирає витаскує випадкові зразки і використовує їх для відновлення даних у декодері

Декодер працює по стандартному принципу, а саме отримає зразки з Прихованого простору і перетворює їх на нові вихідні данні які є подібними до вхідних

Як було зазначено вище, ціль варіаційного автоенкодера в розширенні наявних даних, створюючи нові синтетичні зразки. Це активно застосовується для аугментації даних, тобто створення нових екземплярів для тренування моделі, бо чим більше розмір датасету, тим кращий результат роботи моделі отримуємо на виході. Також VAE активно застосовується в генерації варіацій, це доволі часто потрібно у творчих індустріях таких як: дизайн, музика і мистецтво. А також доволі креативне застування VAE це відновлення втрачених або пошкоджених даних, так як схожі зразки можуть заповнити пропуски.

Варто зазначити що використання байєровських методів у VAE є поширеною практикою, яка позитивно впливає на тренування моделей завдяки апроксимації розподілу прихованих змін досягається більш ефективне навчання моделей, що призводить до зменшення часу під час тренуванні, особливо на складних і високовимірних даних.

2.6 Порівняння і підсумки методів генерування зображень

Розібравши найпопулярніші механізми і моделі для генерування зображень, було надано особливу увагу розвитку моделі CLIP. Цю модель була обрано через те, що сучасна проблема ШІ полягає в браку розуміння контексту і лексики. Ця модель вже застосовує унікальні способи тренування, які сфокусовані якраз на вирішення цієї проблеми. Також за рахунок того, що тренування складних

моделей, які працюють з зображеннями потребують великих обчислювальних потужностей, а робота з CLIP не потребує перенавчанню з нуля, достатньо буде довчити на новому датасеті, щоб отримати новий результат робить цю модель ідеальною для подальшого дослідження і створення нової більш ефективної моделі

Розділ 3. Покращення моделі CLIP за допомогою перцептивно вирівняних градієнтів (PAG)

3.1 Ознайомлення з перцептивно вирівняними градієнтами (PAG)

Перцептивно вирівняні градієнти – це один з методів покращення візуального сприйняття даних. Принцип роботи полягає в тому, що корегуються градієнти кольорів, утворюючи більш плавний і рівномірний перехід кольорів, що відповідає сприйняття картинки людським оком.

3.2 Теоретичне результати при додавання перцептивно вирівняних градієнтів до моделі CLIP

Теоретично, можна спробувати покращити лексичне розуміння моделі CLIP під час тренування, якщо використати перцептивно вирівняти градієнти які зроблять нові вхідні екземпляри з більш чітко окресленими деталями які враховують сприйняття людського ока під час дотренування моделі CLIP на новому датасеті.

CLIP вже використовує контрастивне навчання. Контрастивне навчання є підходом у машинному навчанні, зокрема в області обробки зображень та обробки природної мови, який спрямований на навчання моделей шляхом порівняння подібностей та відмінностей між парами прикладів. Основна ідея полягає в тому, щоб моделі вчилися розрізняти, що робить один приклад подібним чи відмінним від іншого. Це досягається шляхом навчання на позитивних і негативних парах Такий метод оптимізації під час тренування забезпечить покращення якості зображення, зменшення кількості артефактів і отримання природніх форм на згенерованих зображеннях.

3.3 Обрання датасету для тренування

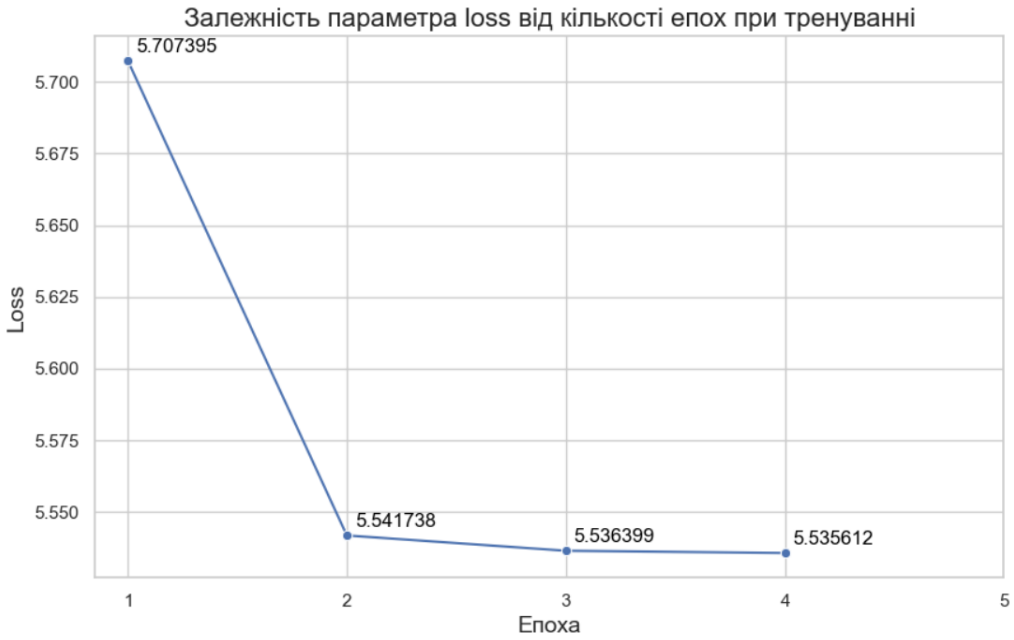
Для початку треба обрати датасет для тренування нової моделі яка буде називатись в подальшій роботі CLIPPAG. Мій вибір прийшовся на датасет

SBUcaptions. Це один з найбільших публічно доступних датасетів який складається з 1 мільйону зображень і підписів до них. Особливість цього датасету полягає в тому, що зображення і підписи було зібрано з соц. мереж від реальних користувачів, тому датасет містить в собі неформальну мову та синтаксичні помилки. Такі данні є дуже корисними під час тренування моделі, яка повинна працювати з реальними даними, тому цей датасет ідеально підходить для тренування CLIPPAG

3.4 Тренування моделі

Для тренуванні моделі було використано сервіс RunPod, який дозволяє орендувати їх потужності. Було використано кластер з графічним процесором Nvidia L40, який має 48 гігабайтів відео, також кластер мав в собі 58 гігабайтів оперативної пам’яті і SDD на 300 гігабайтів. Таких великих потужностей все одно не вистачало для нормального тренування моделі, тому під час розробки застосовувались різні методи оптимізації і паралельне програмування задля зменшення часу тренування.

Модель тренувалась в 4 епохи, саме така кількість є оптимальною, так як параметр Loss після 2ої епохи майже перестав зменшуватись. Наглядно це можна побачити на графіку 1. Де по осі Y зазначено Loss, а по осі X кількість епох. Значення Loss в межах 5 є нормою для моделі яка працює з текстом і зображеннями одночасно, але завдяки епохам все одно вдалось трохи зменшити цей показник. У контексті машинного навчання та штучного інтелекту, параметр "Loss" вказує на різницю між прогнозованими результатами моделі та її фактичними, правильними результатами. Це ключовий показник, який використовується під час тренування моделей для оцінки того, наскільки добре модель виконує своє завдання.



Графік 1. Залежність параметра loss від кількості епох

3.5 Застосування проєктованого градієнтного спуску

Під час тренування даної моделі була застосовано PGD-атака. PGD атака це різновидність змагальної атаки, яка додає невеликі навмисні зміни до оригінального зображення, щоб заплутати модель за допомогою модифікацій градієнту. Наглядну роботу змагальної атаки можна побачити на Зображенні 1.

Зображення 1. Приклад застосування змагальної атаки

Після застосування атаки модель-класифікатор перестає розпізнавати вхідне зображення свині як свиню і починає вважати, що це літак. Ось так PGD атака виглядає математичною формулою, де

$$x_{t+1} = Proj_{\mathcal{X}}\left(x_t + \alpha \cdot sign(\nabla_x J(\theta, x_t, y))\right)$$

Формула 1. PGD атака

x_{t+1} - оновлене значення зображення на ітерації $t + 1$

$Proj_X$ — проекція на допустимий набір даних, в рамках атаки на зображення це буде обмеженням пікселів у самому зображенні.

x_t - значення зображення на поточній ітерації

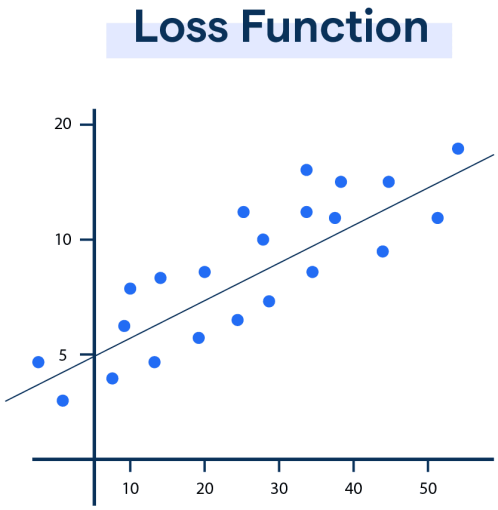
α — крок атаки

$\text{sign}\nabla x J(\theta, xt, y)$ — знак градієнта по x функції втрат J від параметрів моделі θ, поточного вхідного прикладу x_t та правильної мітки y

3.6 Градієнт функції витрат

Градієнт функції витрат є вектором часткових похідних функції витрат відносно кожного з її параметрів. У контексті машинного навчання та оптимізації, функція витрат вимірює, наскільки добре модель передбачає цільові значення.

На графіку 2 зображено саму функцію втрат. На графіку показано, як розсіювання точок даних відносно лінії регресії визначає величину втрат. Чим ближче точки до лінії, тим менша функція витрат, що вказує на кращу відповідність моделі даним.



Графік 2. Функція витрат

Математично запис градієнту функції витрат виглядає наступним чином:

$$\nabla_x J(\theta, x, y) = \left(\frac{\partial J(\theta, x, y)}{\partial x_1}, \frac{\partial J(\theta, x, y)}{\partial x_2}, \ldots, \frac{\partial J(\theta, x, y)}{\partial x_n} \right)$$

Формула 2. Градієнт функції витрат

Де J — це функція витрат

θ — параметри моделі

x – вхідні дані

y — вихідні дані

$x1$, $x2$,…, xn — компоненти вектору вхідних даних x.

3.7 Інтегрування PAG у PGD атаку

В попередній підрозділах було розібрано, що таке PGD атака, але не було вказано, що якраз ця атака дозволяє перцептивно вирівняти градієнти на вхідних зображеннях. Щоб перетворити з PGD атаки на PAG, достатньо помножити градієнт функції витрат на ваговий коефіцієнт, який враховує перцептивні особливості.

$$x_{t+1} = \mathrm{Proj}_{\mathcal{X}}\left(x_t + \alpha \cdot \mathrm{sign}(\nabla_x J(\theta, x_t, y) \cdot w)\right)$$

Формула 3. PGD атака яка перцептивно вирівнює градієнти

Для такого випадку можна використовувати декілька метрик вагових коефіцієнтів, а саме: LPIPS або SSIM

- LPIPS використовує нейронні мережі, навчені на великій кількості зображень, щоб виміряти перцептивну схожість. Вона обчислює різницю між активаціями глибоких шарів попередньо натренованої нейронної мережі. Саме цю метрику я і обрав

- SSIM обчислюється на основі трьох компонентів: яскравості, контрасту і структури. Вона порівнює локальні патерни пікселів вікна, що рухається по зображенню, для обчислення середнього значення, дисперсії та коваріації.

Завдяки LPIPS градієнт під час атаки створює вразливі прикладів шляхом максимізації втрат моделі і враховує перцептуальні характеристики зображень або іншими словами як зображення сприймається саме людиною.

3.8 Порівняння отриманих результатів

Мною було натреновано дві моделі. Звичайна дотренована модель CLIP на датасеті SBUcaptions і CLIPPAG – модель яка також дотренована на SBUcaptions, але з урахуванням перцептуально вирівняних алгоритмів. Моделі порівнювались в рівних умовах з однаковим текстом запитом на генерування і однаковою кількістю ітерацій, а також було використано нейромережу класифікатор, яка підтверджувала результати метрик.

Отже за запитом “A painting of an apple in a fruit bowl”, що перекладається як картина яблока в мисці з фруктами можемо порівняти 3 результати. На Зображенні 2 результати зліва-направо використання звичайної моделі CLIP, CLIP (SBUcaptions) і CLIPPAG. Розберемо кожне зображення окремо.

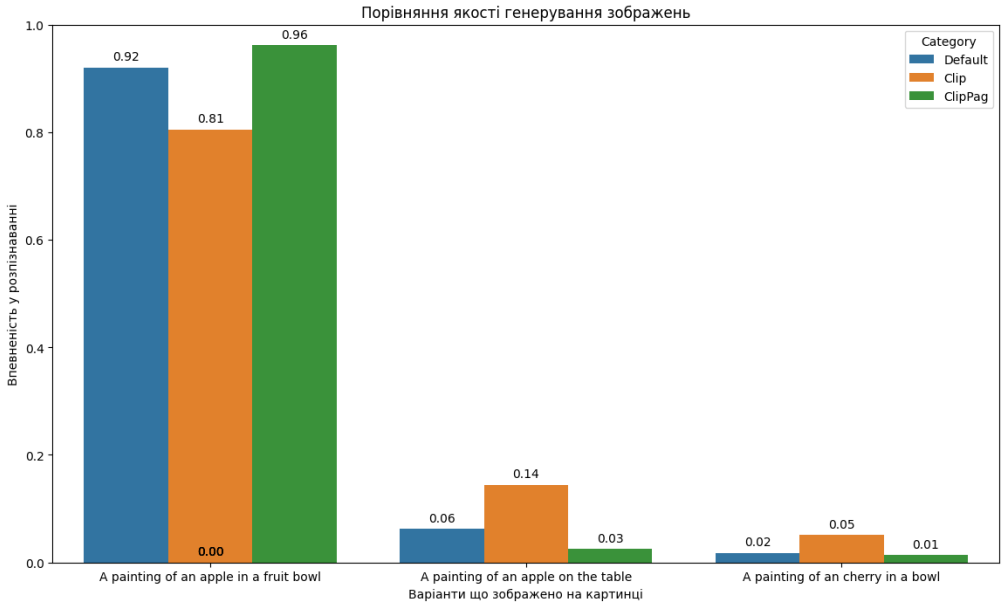
Зображення 2. Результати генерація запиту “A painting of an apple in a fruit bowl”

На лівому зображенні можна побачити чисельні артефакти, такі як деформація форми миски, деформація форми яблука, а також об’єкт який більше схожий на помідор, ніж на яблуко

На центральному зображенні можна побачити покращення форми миски і навіть розгледіти правдоподібну текстуру дерева на ній, але все ще забагато артефактів, особливо листок, хоча форма яблука більш кругла, ніж у лівого зображення

На правому зображенні, де застосовувались перцептивно вирівнянні градієнти можна розгледіти більшу увагу до деталей. Маємо правильну перспективу і форму яблука зліва, навіть було додано деталь як чорна точка, яка часто зустрічається на шкурці яблука і помітно різні нерівності. Також варто зазначити кращу передачу кольорів самої миски і майже ідеальну текстуру дерева.

Але щоб бути об’єктивним, застосуємо вище згаданий класифікатор. Цей класифікатор на вхід отримує 3 вхідних текста і зображення. Він має розтавити вірогідність пари зображення до тексту

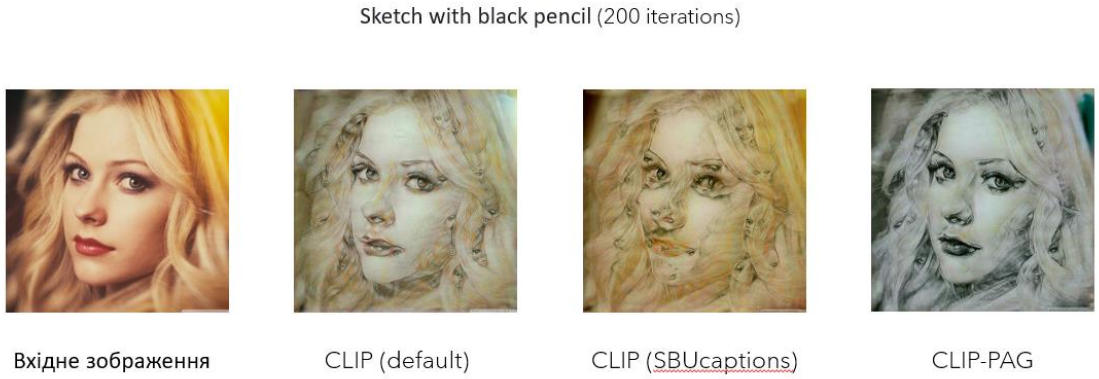


Графік 3. Метрика розпізнавання зображень класифікатором

Отже, синім кольором виділено класифікацію звичайної CLIP моделі, помаранчевим – CLIP (SBUcaptions) і зеленим CLIP-PAG. Найгірший результат показала модель CLIP (SBUcaptions), а найкращий CLIP-PAG, що підтверджує ефективність використання PAG під час тренування моделі CLIP

Наступним прикладом для порівняння було трансформування вхідного зображення у різні стилі. Аналогічно як на з Зображенням 2 маємо аналогічний порядок моделей на Зображенні 3. Запит який використовується для генерування – “Sketch

with black pencil” який перекладається як малюнок чорним олівцем і на генерування кожного з зображень було виділено по 200 терацій



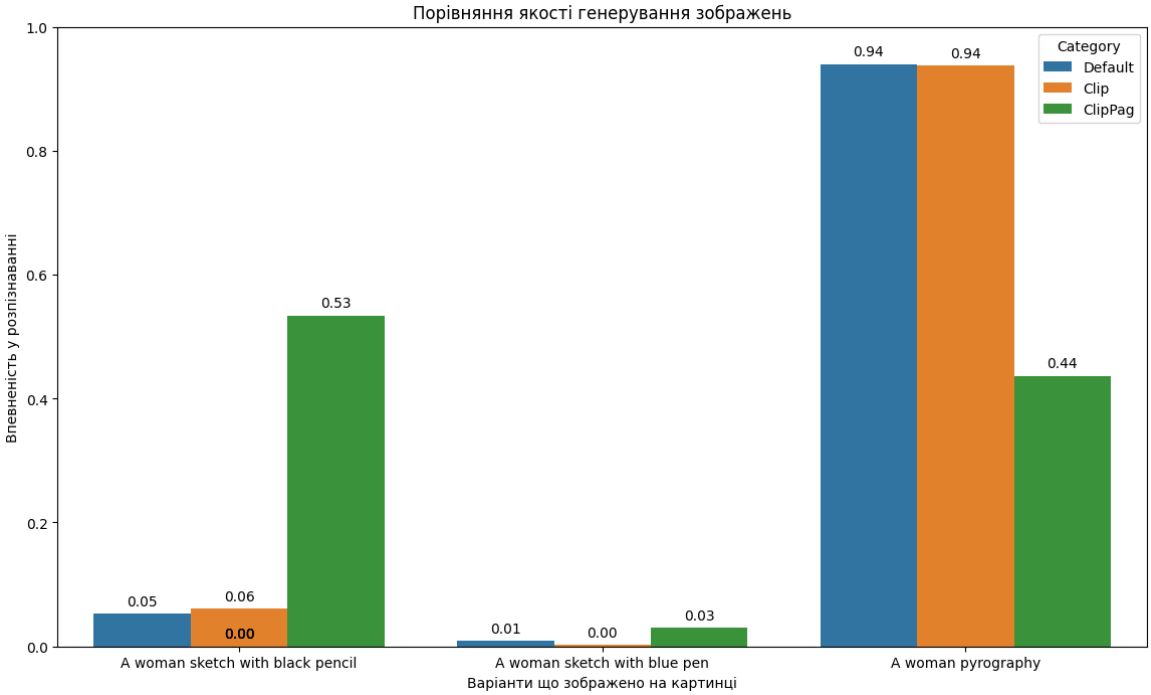
Зображення 3 Результат генерації по запиту “Sketch with black pencil”

На лівому (не вхідному) зображені бачимо середню кількість артефактів, зображення виглядає як малюнок, але передача кольору більш жовта ніж чорна.

На центральному зображені бачимо високу кількість артефактів, абсолютно неправильний колір, але в порівняння з лівим трохи краще промальовані деталі в очах

На правому зображенні відразу помітна правильна передача кольорів, найменша кількість артефактів, краще виділення локонів на фоні, а також деталі на віях, очах і губах.

Знову повертаючись до метрики бачимо на Графіку 4 цікавий результат, що лише модель CLIP-PAG було правильно класифікована, дві інші моделі проварили свою класифікацію через неправильну передачу кольору і були розпізнані не як малюнок чорним олівцем, а як випалювання по дереву



Графік 4. Метрика генерації зміни стиля вхідного зображення

Висновок

Застосування перцептивно вирівняних градієнтів під час тренування моделі CLIP демонструє значні покращення в результатах генерування зображень. Отримані результати демонструють увагу до дрібних деталей, правильні геометричні форми, додавання реалістичної текстур матеріалів, зменшення кількості артефактів, також тести в різних задачах показали, що модель зберігає свою універсальність і краще передає правильно кольори.

Джерела

1) CLIPAG: Towards Generator-Free Text-to-Image Generation [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: https://arxiv.org/abs/2306.16805.
2) History and evolution of machine learning: A timeline [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: https://www.techtarget.com/whatis/A-Timeline-of-Machine-Learning-History.
3) ELIZA: a very basic Rogerian psychotherapist chatbot [Електронний ресурс] – Режим доступу до ресурсу: https://web.njit.edu/~ronkowit/eliza.html.
4) SELENE - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHZ, NVIDIA A100, MELLANOX HDR INFINIBAND [Електронний ресурс] – Режим доступу до ресурсу: https://www.top500.org/system/179842/.
5) CLIP [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/openai/CLIP.
6) Gradient-based Adversarial Attacks : An Introduction [Електронний ресурс] // 2020 – Режим доступу до ресурсу: https://medium.com/swlh/gradient-based-adversarial-attacks-an-introduction-526238660dc9.
7) RunPod [Електронний ресурс] – Режим доступу до ресурсу: https://www.runpod.io/.
8) Are Perceptually-Aligned Gradients a General Property of Robust Classifiers? [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://arxiv.org/abs/1910.08640.
9) SBUcaptions dataset [Електронний ресурс] – Режим доступу до ресурсу: https://paperswithcode.com/dataset/sbu-captions-dataset.