Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка та дослідження методів одноразового навчання для мультимодальних даних
(тема)

Виконав:
студент 2 курсу, групи СШІм-21-1
Стахевич А.В.
(прізвище, ініціали)

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

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

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник к.т.н., доц. Турута О.П.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Стахевич Анні Віталіївні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та дослідження методів одноразового навчання для мультимодальних даних

затверджена наказом університету від 31 березня 20 23 р. № 306Ст

2. Термін подання студентом роботи до екзаменаційної комісії 19 травня 20 23 р.

3. Вихідні дані до роботи розроблені реалізації підписів для зображення англійською мовою, набір мультимодальних даних.

4. Перелік питань, що потрібно опрацювати в роботі розкриття теми предметної галузі, основна проблема досліджуваного питання, тенденції в розробці проблеми, опис очікуваної моделі, розробка структури моделі, розробка концептуальної моделі, постановка задач,аналіз існуючих рішень, аналіз сучасного стану дослідження, розгляд архітектури моделі, обробка вхідних даних та отримання ознак з даних, обробка тексту, обробка зображення, існуючі рішення моделей для задачі створення підписів, мультимодальне навчання, передавальне навчання, змагальне навчання, модель кодера-декодера, модели с внимательностью, модель трансформер, мета-навчання LSTM, методи аналізу для отриманих результатів, bert score, bleu-n, meteor,rouge, розробка моделей для підпису до зображення, набір даних, згорткова нейронна мережа, рекурентна нейронна модель, кодер-декодер, модель кодер-декодер з механізмом уваги, трансформер модель, аналіз результатів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) головна ідея моделі, навчання моделі для підпису зображення, принцип роботи моделі для підпису зображення, демонстрація підготовки даних для моделі, представлення блоку попередньої обробки, етап вирішення завдання комп'ютерного зору, кроки реалізації моделі, обробка тексту, обробка зображення, представлення моделі комп'ютерного зору, схема згорткової нейроної мережі, демонстрація етапу моделювання, архітектура мультимодальні нейронної мережі для підпису зображення, архітектура підпису для зображення на основі моделі кодеру декодеру, архітектура для підпису зображень на основі кодеру декодеру, та механізму уваги, демонстрація етапу оцінки моделі, архітектура RNNs моделі, архітектура LSTMs моделі, архітектура GRUs моделі, архітектура кодеру-декодеру, демонстрація Merge моделі, демонстрація обробленого тексту, архітектура кодеру-декодеру з механізмом уваги, базова архітектура моделі трансформер, демонстрація моделі трансформеру з механізмом уваги, демонстрація роботи Multi-head Attention, результат порівняння оцінок моделі та його партій, демонстрація результату при партії 32, демонстрація результату при партії 64, демонстрація результатів моделі з механізмом уваги, демонстрація результатів моделі трансформеру з механізмом уваги, приклад результатів кодер-декодер, приклад результатів кодер-декодер с механізмом уваги, приклад результатів трансформер модель с механізмом уваги.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	03.04.2023	Виконано
2	Аналіз предметної області та постановка завдання	04.04.2023-05.04.2023	Виконано
3	Аналіз існуючих рішень	06.04.2023-10.04.2023	Виконано
4	Проектування моделі для створення підписів	11.04.2023	Виконано
5	Реалізація моделі	14.042023-21.04.2023	Виконано
6	Проведення експериментів та оцінка результатів	24.04.2023-06.05.2023	Виконано
7	Обробка та оформлення результатів	08.05.2023	Виконано
8	Оформлення графічних матеріалів	09.05.2023	Виконано
9	Оформлення пояснювальної записки	10.05.2023-11.05.2023	Виконано
10	Попередній захист	12.05.2023	Виконано
11	Захист перед ЕК	19.05.2023	Виконано

Дата видачі завдання 3 квітня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ к.т.н., доц. Турута О.П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 92 с., 34 рис., 4 табл., 1 дод., 37 джерел.

ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, КОМП'ЮТЕРНИЙ ЗІР, МУЛЬТИМОДАЛЬНІ ДАНІ, ОБРОБКА ПРИРОДНОЮ МОВОЮ, РЕКУРЕНТНА НЕЙРОННА МЕРЕЖА.

Об’єкт дослідження – мультимодальні моделі для аналізу зображень і генерації текстів.

Мета роботи – дослідження і розробка моделей для створення опису зображень, які представляють невеликі класи, українською мовою

Методи дослідження – моделювання архітектури моделей, комбінація моделей, тренування моделей. Проведення аналізу існуючих рішень для задач комп’ютерного зору, обробки природної мови та методів оцінки результатів. Аналіз літератури та електронних ресурсів.

Результатом роботи є моделя для створення підпису до зображення, яка створює опис за допомогою української мови.

Сферою застосування даної роботи є український ринок нейронних мереж, які здатні створювати підпис до зображення.

Значимість роботи полягає в створенні моделі підпису до зображення україною мовою, збільшення кількість україномовних ресурсів.

ABSTRACT

Explanatory note: 92 p., 34 fig., 4 tabl., 1 ann., 37 sources.

COMPUTER VISION, CONVOLUTIONAL NEURAL NETWORKS, MULTIMODAL DATA, NATURAL LANGUAGE PROCESSING, RECURRENT NEURAL NETWORKS.

The object of research is multimodal models for image analysis and text generation.

The purpose of the work is research and development of models for creating a description of images that represent small classes in the Ukrainian language

Research methods – modeling of model architecture, combination of models, training of models. Analysis of existing solutions for computer vision, natural language processing and results evaluation methods. Analysis of literature and electronic resources.

The result of the certification work is a model for creating a caption for an image, which creates a description using the Ukrainian language.

The field of application of this work is the Ukrainian market of neural networks, which are capable of creating a caption for an image.

The significance of the work lies in the creation of a model of the caption for the image in the Ukrainian language, increasing the number of Ukrainian-language resources.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ШНМ – штучні нейронні мережі;

BERT score – bidirectional encoder representations from transformers – представлення двонаправленого кодера для трансформатору;

BLEU-n – bilingual evaluation understudy – двомовне оцінювання;

GAN – generative adversarial networks – генеративна змагальна мережа;

CNN – convolutional neural networks – згорткова нейронна мережа;

GRU – gated recurrent unit – закритий рекурентний блок;

LSTM – long short-term memory – довготривала короткочасна пам’ять;

METEOR – metric for evaluation of translation with explicit ordering – метрика для оцінки перекладу з явним упорядкуванням;

MNLM – multimodal neural language models – мультимодальні нейронні мовні моделі;

RNN – recurrent neural network – рекурентна нейронна мережа;

ROUGE – recall-oriented understudy for gisting evaluation – орієнтований на запам’ятовування дублер для оцінки гістінга.

ВСТУП

Оскільки дані та інформація відіграють дедалі важливішу роль у різних галузях і секторах, технології комп'ютерного зору та обробки зображень стають невід'ємною частиною нашого життя. У цьому контексті опис зображень, який дозволяє комп'ютерам аналізувати і розуміти зміст зображень так само, як це робить людина, є одним з основних напрямків розвитку цієї технології. У зв'язку з цим напрямок з опису зображень стають все більш затребувані серед фахівців, які хочуть поглибити свої знання та удосконалити професійні навички в галузі комп'ютерного зору.

Для створення надійної моделі, яка може описати детально зображення необхідна велика кількість надійних прикладі. Інколи це стає проблемою, оскільки не завжди є можливість отримати достатній набір даних. У цьому випадку необхідно розробляти більш надійні системи, які можуть навчатися не гірше інших моделей на маленькому наборі даних, та видавати більш точні результати. Існують багато різних технологій, які вже зарекомендували себе у даному напрямку. Однією з таких стала технологія одномоментного навчання. Дана технологія дає змогу навчити модель розпізнавати та описувати зображення при невеликому кількість прикладів. Це можно порівняти з інтуєтивним навчанням у людини, де людині показують декілька прикладів, а після цього вони можуть описати на основі попереднього досвіду.

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Розкриття теми предметної галузі

Останнім часом все більше компаній і організацій вирішують задачі пов’язані з навчанням моделей на невеликому наборі даних. Це пов’язано з тим, що не завжди можливо отримати доступ до великих даних, або через те, що малі дані є більш репрезентативними для конкретного завдання.

Недавні дослідження показали, що невеликий набір даних не обов’язково є перешкодою для навчання моделей машинного навчання високої точності. У статті 2021 року, опублікованій у Nature Communications, автори показали, що лише кілька десятків чи сотень прикладів можна використовувати для класифікації зображень у задачі розпізнавання обличчя. Для досягнення високої точності автори використовували методи збільшення даних, методи навчання передачі та полегшені архітектури нейронних мереж.

Навчання моделі на невеликому наборі даних має ряд переваг. По-перше, це значно скорочує час і вартість збору та розділення даних. По-друге, використання меншої кількості даних дозволяє уникнути проблеми надмірного навчання моделі, коли вона починає «запам’ятовувати» дані навчальної вибірки та не може узагальнити знання на нові дані. Крім того, навчання моделі на невеликому наборі даних може бути корисним, коли доступ до даних обмежено, наприклад, якщо дані конфіденційні або захищені авторським правом.

Щоб успішно навчити модель на невеликому наборі даних, необхідно враховувати низку особливостей. Важливо вибрати правильну архітектуру моделі, оптимізатор і функцію втрат, а також виконати доповнення даних, щоб розширити навчальну вибірку. Крім того, можуть бути використані

методики передавального навчання, коли модель, навчена на великому наборі даних, адаптується до нового завдання з невеликою кількістю даних.

Ще один метод, який може допомогти під час навчання на невеликому наборі даних є використання методів активного навчання. Це означає, що модель вибирає найбільш інформативні приклади для навчання та запитує експертів про нові приклади для додавання до набору даних.

Загалом навчання моделі на невеликому наборі даних є важливою тенденцією в машинному навчанні. Сучасні методи можуть надати високоточні моделі на невеликих наборах даних, що робить їх корисними в різних сферах. А саме від медичних додатків до фінансової аналітики та напрямках пов’язаних з розпізнаванням мови, комп’ютерного зору, обробки природної мови тощо. Однак необхідно враховувати особливості такого навчання і правильно підходити до вибору моделі і її параметрів.

Розкриваючи тему комп'ютерного зору та використання природної мови то можно бачити, що останні дослідження мають багато проривів у даному напрямку. Один з яскравих прикладів – це розробки в області машинного перекладу, такі як «DeepL», а також покращення перекладу за допомогою алгоритмів машинного перекладу "Google translate". Розвиток комп'ютерного зору теж не стоїть на місці. Багато фабрик, заводів впроваджують комп'ютерний зір для стеження за контролем якості продукції та створення механізмів які використовують комп'ютерний зір для покращення виготовлення продукції. Також значного поширення даної технології можно бачити при розробці нових моделей захоплення зображення, фільтрів у соціальних мережах, системах захисту тощо.

За останні 5 років можно спостерігати тенденцію, як використання штучного інтелекту впливає на наше життя. Більшість технологій які використовуємо у повсякденному житті допомагають нам вправлятися з великою кількістю проблем. Наприклад вести соцмережі, прискорення написання коду, прискорення пошуку інформації, технології створення гарного відео, як професіональний відеофотограф, розпізнавання обличча,

рекомендаційні системи и рекламні системи, технології розпізнавання мови. Більшість людей навіть не уявляють, як сильно штучний інтелект допомагають нам у цьому. Наведені приклади відображають на скільки технології комп'ютерного зору та системи обробки природної мови вже застосовуються сьогодні.

Останні тенденції показують нам наскільки важливо створювати дослідження у даному напрямку. Одним із самих цікавих напрямків роботи комп'ютерного зору та методів обробки природної мови є опис до зображення. Гарним прикладом демонстрації цього напрямку є робота присвячена створенню тегів для зображення. Данна розробка дає можливість створювати якісні хештеги, які можуть бути використані для соцсетей, Ютубу, тощо.

Проаналізував даний напрямок більш детально можно побачити наскільки зараз актуальні системи, які спроможні створити опис для зображення. Проаналізував ринок користувачів, що зацікавлені у розробці підпису до зображення було виявлено наступне. Є багато цільових груп, такі як дизайнери,фотографи, ті хто володіють соціальними мережами та люди з обмеженими можливостями, що потребують у створенні систем, які можуть створювати опис до зображення. Гарним прикладом демонстрації актуальності розробки систем є люди з обмеженими можливостями. Ці групи потребують системи, які можуть зробити опис зображення, а потім озвучити, що є на зображенні. Розробка та дослідження у цьому напрямку є важливою, оскільки це покращує рівень життя та дає можливість жити набагато комфортніше.

Актуальність розвитку даного напрямку і є дуже важливим кроком, але що робити коли є концепт моделі, але немає великої кількості для навчання. У цьому випадку, розглядання теми коли коли є невеликі набір даних та створення підписів до зображення, є дуже важливою темою для маленьких компаній, які неспроможні зібрати велику кількість анотованих даних.

1.2 Основна проблема досліджуваного питання

Під час розробки різноманітних видів моделей може бути багато проблем, які обов’язково потребують вирішення для досягнення бажаного результату. Одним із найважливіших етапів перед початком роботи над моделлю це зробити постановку задачі. Коли перший крок був зроблений, розробники починають створювати концепт моделі та шукати набір даних, який повинен задовольняти вимоги замовника.

Найбільш розповсюдженою проблемою при пошуку набору даних є те, що компанії не збирають велику кількість даних за певних причин. Самі причини можуть бути різними та мати свої обмеження.

Першою причиною є конфіденційність даних. Деякі компанії працюють з конфіденційними даними й можуть не збирати їх великі обсяги, щоб запобігти витоку даних або порушенням конфіденційності.

Другою причиною є обмеження ресурсів. Збір великої кількості даних може бути дуже дорогим і потребує великих фінансових і людських ресурсів. Деякі компанії можуть не мати достатніх ресурсів для здійснення такого процесу.

Третьою причиною є неспроможність збирати дані. Деякі дані можуть бути недоступними або важко доступними, такі як медичні дані чи дані про покупки, які можуть бути захищені законом.

Попередні проблеми були пов’язані з необхідністю збирати дані. Наступні проблеми більш пов’язані з часом розробки та моделью та методом навчання.

Одним з прикладів є обмеження у часі, коли продукт повинен вийти на ринок за обмежений термін. Це може призвести до потреби в невеликому наборі даних для навчання моделі.

Ще одним прикладом є необхідність швидкого адаптивного навчання. Деякі компанії можуть працювати у швидкому та динамічному секторі, де

потрібне швидке адаптивне навчання моделі, і для швидкого запуску оновленої моделі може знадобитися використання невеликого набору даних.

Коли було стверджено підходящий набір даних для певної задачі, розробники переходять до створення концепту моделі та розробки моделі. Під час розробки певної моделі, у розробників може виникнути різні види проблем.

У контексті розгляду магістерської роботи, буде розглянуто,які проблеми можуть виникнути при розробці моделі, яка може створювати підпис для зображень.

Під час розробки моделі для підпису зараження певними проблемами може бути наступне.

Першою проблемою може бути недостатність даних. Оскільки для тренування певних модель необхідна достатня кількість зображень та їх відповідна кількість підписів. Якщо буде недостатньо даних, то модель може бути недостатньо точною. Неточність моделі може призвести до того, що розробнику потребується більше часу на розробку. Тим саме збільшивши витрати на розробку продукту.

Другим недоліком який може бути при розробці моделі, також стосується даних. Дані можуть бути не відповідними. Якщо підписав зображення не відповідає зображенню. Наприклад, на зображенні бачимо поле, а у підписі вказано гори. Навчившись на такому наборі даних, модель може зрозуміти неправильный контекст. Це призведе до того, що модель буде видавати недійсний контекст. Тому при роботі з даними потрібно бути дуже уважними та перевіряти весь зібраний набір даних на відповідність.

Наступною проблемою може стати проблеми із зображенням. Якщо модель не буде здатна розпізнати всі об'єкти на зображенні, то це може призвести до того, що для створення підпису не буде достатньо інформації. Тим самим погіршить підпис до зображення.

Останньою проблемою може бути, коли маємо підписи але вони є різними мовами. Якщо модель не передбачає того що вона може навчатися на декількох мовах, то при підписі зображення модель може генерувати не розуміли контекст використовуючи кілька мов. Це призведе до того що кінцевий підпис до зображення буде незрозумілого контексту.

1.3 Тенденції в розробці проблеми

Розробка моделей для підпису зображень є однією з активних областей досліджень у галузі комп'ютерного зору та машинного навчання. Для цього застосовуються різні підходи та методи, що дають можливість розпізнавати об'єкти на зображеннях та створювати підписи на їх основі. Щоб створити точний та зрозумілий підпис до зображення, потрібно враховувати багато чинників, таких як контекст зображення, відповідність темі, граматику та стиль написання.

Один з основних підходів – це використання глибоких нейронних мереж, таких як CNN. Ці моделі можуть ефективно визначати об'єкти на зображеннях та створювати підписи на їх основі. Однак, деякі дослідники використовують комбіновані моделі, які поєднують глибинні нейронні мережі з традиційними алгоритмами обробки зображень. Це дозволяє більш ефективно створювати підписи на основі різних методів розпізнавання об'єктів.

Техніки аугментації даних також можуть бути корисними при розробці моделей для підпису зображень. Вони дозволяють моделям навчитися робити підписи на більш широкому спектрі зображень шляхом зміни розміру зображень, яскравості, контрастності та додавання шуму до зображення.

Передавальне навчання є ще одним ефективним методом для навчання моделей для підпису зображень. Цей метод полягає в тому, що

навчена модель використовується для підтримки навчання нової моделі. Це дозволяє ефективно використовувати навчені моделі для підпису зображень в різних областях.

Крім того, активне навчання є ще одним напрямком досліджень, що дозволяє покращити якість підписів на зображеннях. Воно полягає в тому, що модель може взаємодіяти з людиною для отримання додаткової інформації та підтримки у навчанні. Наприклад, модель може запитувати у користувача підтвердження, чи є підпис, створений моделлю, правильним, що дозволяє покращити якість підпису.

Також, для розробки моделей для підпису зображень використовуються різноманітні архітектури, які базуються на різних підходах до обробки зображень. Наприклад, деякі дослідники використовують генеративні моделі, такі як GANs, щоб створювати підписи на основі штучно згенерованих зображень. Це дозволяє моделям вивчати ширший спектр зображень та покращувати якість підпису.

Один із способів розробки моделей для підпису зображень – це використання генеративних моделей, таких як глибокі нейронні мережі, які вчаться генерувати текстові описи зображень. Ці моделі можуть використовувати різні архітектури, наприклад, використовувати зворотний кодер для визначення контексту зображення та генеративний декодер для створення текстового опису.

Інші напрямки досліджень, пов'язані з розробкою моделей для підпису зображень, включають використання навчання з підкріпленням для створення підписів, які максимізують певну метрику якості, таку як зрозумілість або релевантність. Також можна використовувати моделі, які базуються на контексту зображення, щоб створювати більш точні та зрозумілі підписи. Крім того, можна поєднувати різні методи та алгоритми, щоб покращити якість підпису.

Загалом, розробка моделей для підпису зображень є складним завданням, яке вимагає використання різноманітних методів та підходів.

Досягнення прогресу в цій області може мати значний вплив на багато сфер, таких як обробка зображень, комп'ютерне зору, медична діагностика та багато інших.

1.4 Опис очікуваної моделі

Головною метою даної роботи це створення моделі здатна зробити опис зображення. Сам концепт роботи моделі складається з декількох пунктів.

Модель повинна приймати на вхід зображення, а на виході отримувати опис зображення. Це можно побачити на рисунку 1.1.

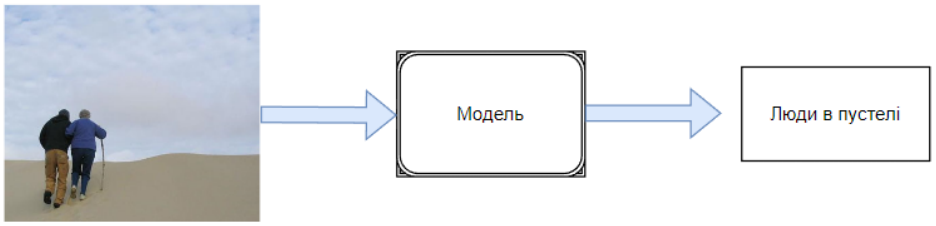

Рисунок 1.1 – Головна ідея моделі

Концепт розробки самої моделі дуже простий. На вході очикуємо дані з нашого набору даних. Самі дані є анатованими, та представлені у вигляді зображення та підпис. Модель знаходить взаємозв'язок зображення та підпису. Потім створює власний підпис до зображення. Оцінюємо на скільки створений підпис та переданий моделі підпис відповідають дійсності. Після повторюємо цикл до того моменту поки не закінчить навчання. На виході будемо отримувати опис, який створює наша модель.

РОЗРОБКА СТРУКТУРИ МОДЕЛІ

2.1. Розробка концептуальної моделі

Розробка моделі, яка здатна реалізувати підписи до задачі є дуже складною задачею. Але оскільки останнім часом все більше дослідників розробляють на покращують методи рішень було створено багато надійних моделей.

Основна ідея створення підпису до зображення складається з того, що вона має декілька етапів. Першим етапом є отримання або збір датасету. Другий етап це підготовка даних для моделі. Третій етап, це тренування моделі. четвертий етап це удосконалення моделі(але не є обов’язковим). П'ятий етап – це оцінка отриманих результатів.

Основуючись на цьому на рисунку 2.1 представлено схему приготування та тренування даних.

Рисунок 2.1 – Навчання моделі для підпису зображення

На вхід очікується пара даних, таких як зображення та опис до нього. у модель неможливо передати дані у тому вигляді які вони є. Перед тим їх треба обробити. Зображення обробити за допомогою моделі комп'ютерного зору та на виході отримати вектор ознак. Текст закодувати у послідовність чисел за допомогою токенайзеру, у якому зберігається слово та його числове представлення. Підготовлені дані передаються на вхід до моделі. Та за допомогою методів природної обробки мови модель навчається

передбачати підпис зображення на основі отриманих ознак зображення. Створена модель необхідно оцінити на ефективність створення підписів для зображення. Для цього було розроблено багато метрик, які будуть розглянуті у роботі.

По закінченню тренування ми отримаємо готову модель, яка стпроможна створювати підписи для зображнння. На рисунку 2.2 можно бачити роботу моделі. На вході ми отримуємо зображення. За допомогою моделі комп'ютерного зору отримаємо вектор ознак. На основі отриманих даних модель буде намагатися створити числову послідовність тексту, яку пропускають через токінезатор, для отримання тексту, який буде описувати зображення.

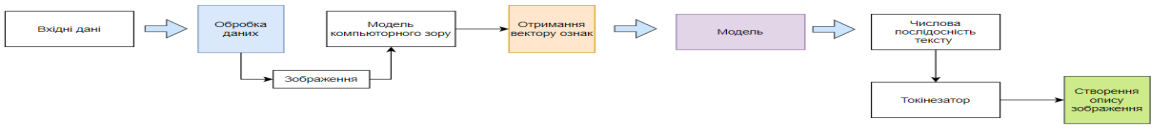

Рисунок 2.2 – Принцип роботи моделі для підпису зображення

2.2 Постановка задачі

У ході проведеного аналізу предметної галузі та можливостей для розробки було створено наступний план виконання роботи:

– розробити архітектуру моделі;

– проаналізувати та дослідити існуючі алгоритми, нейронні мережі та сучасні рішення;

– вибрати датасети для навчання нейронних мереж;

– розробити моделі на основі обраних нейронних мереж;

– провести тестування отриманих молдей і зробити висновки на базі отриманих результатів.

АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

3.1 Аналіз сучасного стану дослідження

З 2011 року було досягнуто значного прогресу у вирішенні складних завдань у напрямку комп'ютерного зору, значною мірою завдяки застосуванню моделей глибокого навчання та великій кількості даних, доступних для дослідників. Останнім часом схожий процес, схоже, відбувається і в галузі обробки природною мовою.

Не дивно, що ці розробки як і в комп'ютерному зорі, так і в обробці природною мовою також викликали нові хвилі міждисциплінарних дослідницьких проблем, що охоплюють обидві області, з яких опис зображень є дуже хорошим прикладом. Як результат, спільноти комп'ютерного зору та обробки природною мовою все більше співпрацюють і організовують спільні семінари протягом останніх кількох років. Ці зусилля призвели до створення нових моделей, наборів даних і метрик, що знайшло відображення у збільшенні кількості публікацій, особливо з 2014 року.

Щоб полегшити огляд, розуміння і порівняння зростаючої кількості досліджень на цю тему, існуючі дослідження пропонують різні схеми класифікації використовуваних моделей. З одного боку, у дослідженні Бернарді та ін. 2017 року [19] запропоновано систему класифікації, що базується лише на двох вимірах і трьох категоріях. З іншого боку, Віа і Ахн 2018 року [20] організували існуючі дослідження відповідно до типу використовуваної архітектури або фреймворку, що призвело до більш детальної класифікації з вісьмома категоріями. Після порівняння обох підходів до класифікації існуючих досліджень було вирішено, що підхід, прийнятий Ваі та Ахн 2018 року [20], є більш точним і описовим і призводить до більш тонкої деталізації, тоді як класифікація Бернарді та ін.

2017[19] є більш абстрактною і призводить до більш грубої деталізації, надаючи перевагу підходу Bai та Ан 2018 року [20]; Нещодавнє дослідження Hossain та ін. 2019 року [21] використовує той самий підхід, що й Bai and An 2018 року [20], з більш сфокусованим оглядом моделей на основі глибокого навчання та посиланнями на найновіші дослідження, опубліковані на сьогоднішній день.

3.2 Розгляд архітектури моделі

Задача створення підписів є однією з тих задач, де модель повинна розуміти зображення, що є задачею комп'ютерного зору. І в той самий момент створена модель повинна не тільки вирішувати задачу комп'ютерного зору, а і створювати зв'язні речення, на основі визначених об’єктів та продемонструвати зв’язок поміж ними використовуючи природну мову.

Даний тип задачі став викликом для алгоритмів машинного навчання і дуже тривалий час був дуже складною задачею. Сама задача повинна була імітувати здатність людини інтерпретувати візуальну інформацію у письмову мову.

Оскільки задача підпису до зображення є складною задачею. То для того щоб її вирішити необхідно поділити її на двоетапний процес. Перший етап передбачає, що модель буде розуміти повний контекст візуального вмісту зображення. Другим етапом є перекладання інформації отриманої від зображення на опис природною мовою.

Вилучення візуальної інформації включає у себе виявлення та розпізнавання об'єктів. Також у цю задачу входить ідентифікація зв’язків поміж вилученою інформацією.

Коли говоримо про етап формулювання опису, говоримо що отриманий опис відповідає трьом властивостям. Перша властивість

відповідає релевантності опису. Іншими слова, це опис який відповідає елементу зображення. Друга властивість відповідає за граматично правильні речення. Третє це опис, який є вичерпними, але водночас короткими, тобто опис має бути спрямований на підсумовування важливих елементів зображення, а не просто на його опис.

Розкриваючи тему генерації тексту, з точки зору обробки природною мовою, завдання є дуже складним. Данне завдання передбачає що для цього необхідно зробити вибір змісту, впорядкування за змістом та генерація семантичного та правильно граматичного речення.

Якщо роздивитись модель детальніше то її можно поділита на декілька кроків. Першим кроком буде створення набору даних для задачі. Цей самий трудомісткий та фінансово затратний процес. Тому більшість віддає перевагу вже готовим наборам даних. Але це залежить від складності та специфіки самого завдання. Другим кроком є попередня обробка зображення. Оскільки більшість наборів даних є необробленими завжди необхідно обробляти дані, щоб бути впевненими у тому, що на вході будемо мати стандартизовані дані, які не будуть мати зайвих відхилень, та створіть модель більш устойчивую для подальшої роботи. Наступним кроком буде попередня обробка тексту. Як із ситуацією із зображеннями, текст, теж необхідно привести до одного виду. Виділити синтаксично не важливі змінення, наприклад, увесь запропонований текст повинен бути нижнього регістру, прибрати лишні пунктуаційні знаки тощо. Ця обробка даних допоможе моделі позбавитись додаткових витрат на обчислення та більш точно робити передбачення без акценту на не важливі деталі. Оскільки моделі не розуміють сучасної мови, а розуміє мову чисел. Іншими словами весь текст необхідно перевести для машини з нашої мови на мову чисел.

Після попередньої обробки можем поділити увесь набір даних на тренувальний валідаційний та тестувальний. Даний поділ необхідний для того, щоб забезпечити об'єктивну оцінку ефективності моделі машинного

навчання та уникнути перенавчання. Тренувальний набір використовується для навчання моделі на вхідних даних та знаходження оптимальних параметрів моделі. Валідаційний набір використовується для налаштування гіпер параметрів моделі та оцінки її ефективності. Тестувальний набір використовується для остаточної оцінки ефективності моделі.Розбиття даних на тренувальний, валідаційний та тестувальний набори дозволяє забезпечити незалежність відповідних наборів даних для тренування, настройки та оцінки моделі, що дає можливість об'єктивно оцінити ефективність моделі на реальних даних. Без такого розбиття, може статися перенавчання моделі, коли вона буде надмірно точно працювати на тренувальному наборі, але буде погано працювати на нових, раніше невиданих даних.

Наступним кроком буде реалізація моделі, яка здатна створювати підписи до заданого зображення.

Останнім кроком розробки є оцінка продуктивності моделі на тестувальному наборі за допомогою таких показників які здатні оцінити та порівняти підпис зображення який був створений моделлю та запропонований підпис з тестувального набору даних.

Якщо роздивитися більш детально обробку тексту, то для моделі важливо врахувати наступні етапи: токенізація, відступ та вбудовування. Під токенезацією мається на увазі розділення тексту на окремі слова та підслова. Під поняттям відступ мається на увазі забезпечення однакової довжини кожного підпису шляхом додавання нулів у кінці коротших підписів. Під поняттями вбудування мається на увазі перетворення токенів у числове представлення за допомогою вбудовування слів, наприклад Word2Vec або GloVe, або інші. На виході після обробки даних будемо мати послідовність слів у реченні, які будуть мати однаковий розмір з максимальною довжиною речення. Словар, який буде складатися із слів та його математичного представлення.

Якщо роздивитись обробку зображення, то для моделі важливо врахувати наступні етапи: зміна розміру, нормалізація, вилучення функцій. Зміна розміру є необхідним кроком, оскільки дані можуть мати різні розміри. Тому завжди необхідно змінювати зображення до єдиного розміру. Розмір зображення обирається виходячи з архітектури CNN і наявних обчислювальних ресурсів. Загальні розміри для підписів до зображень – 224x224, 256x256 або 299x299 пікселів. Змінити розмір можна за допомогою методів інтерполяції, таких як білінійний або найближчий сусід. Нормалізація теж є невід'ємним кроком, яка забезпечує надійність роботи архітектури CNN із зображенням. Після зміни розміру значення пікселів зображення нормалізуються до діапазону між 0 і 1. Це важливо, оскільки кодер CNN очікує вхідних значень у цьому діапазоні. Нормалізацію можна виконати, поділивши значення пікселів на 255, що є максимальним значенням, яке може мати піксель у 8-бітному зображенні. Крім того, значення пікселів можна нормалізувати за допомогою інших методів, таких як віднімання середнього значення та ділення на стандартне відхилення. Для архітектури отримання опису до зображення важливим кроком буде вилучення ознак. Вхідне зображення проходить через кілька згорткових шарів, які застосовують фільтри до зображення та витягують карти функцій. Потім карти функцій зменшуються за допомогою об’єднання шарів, щоб зменшити розмір карт функцій. Вихід кодера CNN є вектором ознак, який представляє зображення. Потім цей вектор подається в мовний декодер для створення підпису для зображення.

Передбачається, що модель підпису до зображення зазвичай складається з двох частин: кодера зображення та декодера мови. Кодер зображень – це згорточна нейронна мережа (CNN), яка обробляє вхідне зображення та виділяє відповідні характеристики. Декодер мови – це рекурентна нейронна мережа (RNN), яка генерує заголовок слово за словом. Вихідні дані CNN подаються в RNN, який потім передбачає наступне слово в підписі на основі попередніх слів і особливостей

зображення. Процес повторюється для кожного слова в підписі, доки не буде згенеровано маркер кінця речення.

Функція втрат є важливим механізмом при навчанні моделі. Функція втрат використовується для вимірювання різниці між прогнозованим заголовком і основним заголовком. Функція втрат може бути крос-ентропійною втратою або іншим варіантом. Під час навчання модель мінімізує функцію втрат, регулюючи ваги кодера CNN і декодера RNN за допомогою зворотного поширення.

Вхід даних моделі є послідовність слів, які складають підпис для вхідного зображення. Щоб перетворити передбачення моделі в текст, зазвичай використовується такий алгоритм декодування, як пошук за променем. Пошук за променем генерує кілька можливих заголовків і вибирає той із найвищою ймовірністю на основі функції підрахунку балів.

Коли процес навчання був завершений, модель необхідно оцінити. Оцінка проходить за допомогою порівняння прогнозованого та реальними підписами. Показники можуть виміряти якість створених субтитрів з точку зору схожості з еталонними прикладами.

Проаналізува увесь концепт роботи моделі, яка описує зображення було отримано схему зображену на рисунку 3.1.

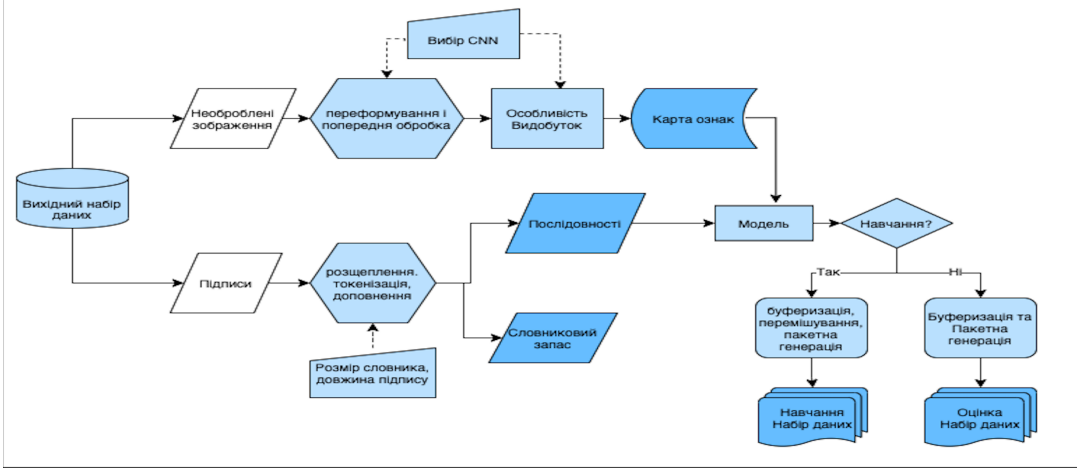

Рисунок 3.1 – Демонстрація підготовки даних для моделі

Дану задачу можно поділити на декілька блоків. Перший блок складається з даних, які були зібрані для задачі. Наступним блоком є поділення даних на зображення та текст. Після цього кожен поділений блок додатком обробляється для моделі. Та останнім кроком є створення самої моделі, яка навчається на оброблених даних.

Натренерована модель, яка навчилась способом зіставлення зображень та підписів за допомогою обраного підходу, подається на процес оцінювання.

3.3 Обробка вхідних даних та отримання ознак з даних

Для роботи з даними необхідно їх завжди обробляти. Це забезпечить надійну та стійку модель. Кожен вид даних обробляється за допомогою своїх технік та прийомів залежно від архітектури обраної моделі.

У розроблені нашій моделі обробка даних є першим етапом. На рисунку 3.2 видно, що обробка даних необхідним кроком. Оброблені зображення передаються до моделі комп'ютерного зору, а текст до токінайзеру.

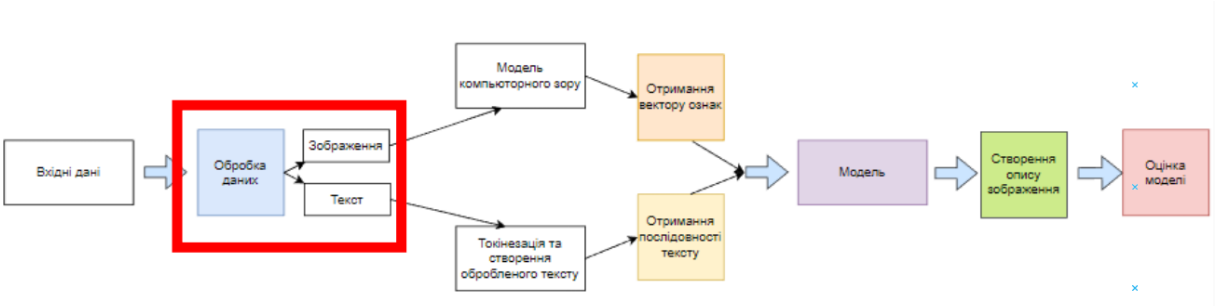

Рисунок 3.2– Представлення блоку попередньої обробки

Даний тип задачі став викликом для алгоритмів машинного навчання і дуже тривалий час був дуже складною задачею. Сама задача повинна була

імітувати здатність людини інтерпретувати візуальну інформацію у письмову мову.

Даний тип задачі став викликом для алгоритмів машинного навчання і дуже тривалий час був дуже складною задачею. Сама задача повинна була імітувати здатність людини інтерпретувати візуальну інформацію у письмову мову.

Оскільки задача підпису до зображення є складною задачею. То для того щоб її вирішити необхідно поділити її на двоетапний процес (рис. 3.3). Перший етап передбачає, що модель буде розуміти повний контекст візуального вмісту зображення. Другим етапом є перекладання інформації отриманої від зображення на опис природною мовою.

Рисунок 3.3 – Етап вирішення завдання комп'ютерного зору

Вилучення візуальної інформації включає у себе виявлення та розпізнавання об'єктів. Також у цю задачу входить ідентифікація зв’язків поміж вилученою інформацією.

Коли говоримо про етап формулювання опису, говоримо що отриманий опис відповідає трьом властивостям. Перша властивість відповідає релевантності опису. Іншими слова, це опис який відповідає елементу зображення. Друга властивість відповідає за граматично правильні речення. Третє це опис, який є вичерпними, але водночас

короткими, тобто опис має бути спрямований на підсумовування важливих елементів зображення, а не просто на його опис.

Розкриваючи тему генерації тексту, з точки зору обробки природною мовою, завдання є дуже складним. Данне завдання передбачає що для цього необхідно зробити вибір змісту, впорядкування за змістом та генерація семантичного та правильно граматичного речення.

Якщо роздивитись модель детальніше то її можно поділита на декілька кроків (рис. 3.4).

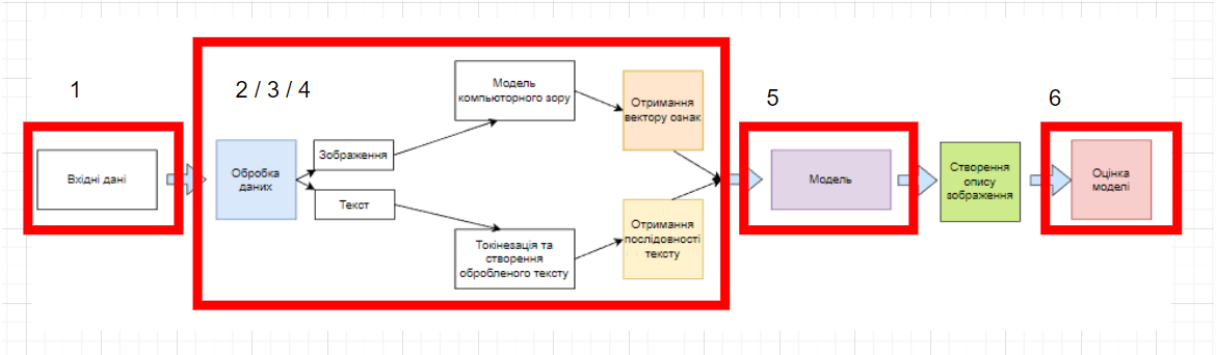

Рисунок 3.4 – Кроки реалізації моделі

Першим кроком буде створення набору даних для задачі. Цей самий трудомісткий та фінансово затратний процес. Тому більшість віддає перевагу вже готовим наборам даних. Але це залежить від складності та специфіки самого завдання.

Другим кроком є попередня обробка зображення. Оскільки більшість наборів даних є необробленими завжди необхідно обробляти дані, щоб бути впевненими у тому, що на вході будемо мати стандартизовані дані, які не будуть мати зайвих відхилень, та створіть модель більш устойчивую для подальшої роботи.

Наступним кроком буде попередня обробка тексту. Як із ситуацією із зображеннями, текст, теж необхідно привести до одного виду. Виділити

синтаксично не важливі змінення, наприклад, увесь запропонований текст повинен бути нижнього регістру, прибрати лишні пунктуаційні знаки тощо. Ця обробка даних допоможе моделі позбавитись додаткових витрат на обчислення та більш точно робити передбачення без акценту на не важливі деталі. Оскільки моделі не розуміють сучасної мови, а розуміє мову чисел. Іншими словами весь текст необхідно перевести для машини з нашої мови на мову чисел.

Після попередньої обробки можем поділити увесь набір даних на тренувальний валідаційний та тестувальний. Даний поділ необхідний для того, щоб забезпечити об'єктивну оцінку ефективності моделі машинного навчання та уникнути перенавчання. Тренувальний набір використовується для навчання моделі на вхідних даних та знаходження оптимальних параметрів моделі. Валідаційний набір використовується для налаштування гіпер параметрів моделі та оцінки її ефективності. Тестувальний набір використовується для остаточної оцінки ефективності моделі. Розбиття даних на тренувальний, валідаційний та тестувальний набори дозволяє забезпечити незалежність відповідних наборів даних для тренування, настройки та оцінки моделі, що дає можливість об'єктивно оцінити ефективність моделі на реальних даних. Без такого розбиття, може статися перенавчання моделі, коли вона буде надмірно точно працювати на тренувальному наборі, але буде погано працювати на нових, раніше невиданих даних.

Наступним кроком буде реалізація моделі, яка здатна створювати підписи до заданого зображення.

Останнім кроком розробки є оцінка продуктивності моделі на тестувальному наборі за допомогою таких показників які здатні оцінити та порівняти підпис зображення який був створений моделлю та запропонований підпис з тестувального набору даних.

Якщо роздивитися більш детально обробку тексту (рис. 3.5), то для моделі важливо врахувати наступні етапи: токенізація, відступ та

вбудовування. Під токенезацією мається на увазі розділення тексту на окремі слова та підслова. Під поняттям відступ мається на увазі забезпечення однакової довжини кожного підпису шляхом додавання нулів у кінці коротших підписів. Під поняттями вбудування мається на увазі перетворення токенів у числове представлення за допомогою вбудовування слів, наприклад Word2Vec або GloVe, або інші. На виході після обробки даних будемо мати послідовність слів у реченні, які будуть мати однаковий розмір з максимальною довжиною речення. Словар, який буде складатися із слів та його математичного представлення.

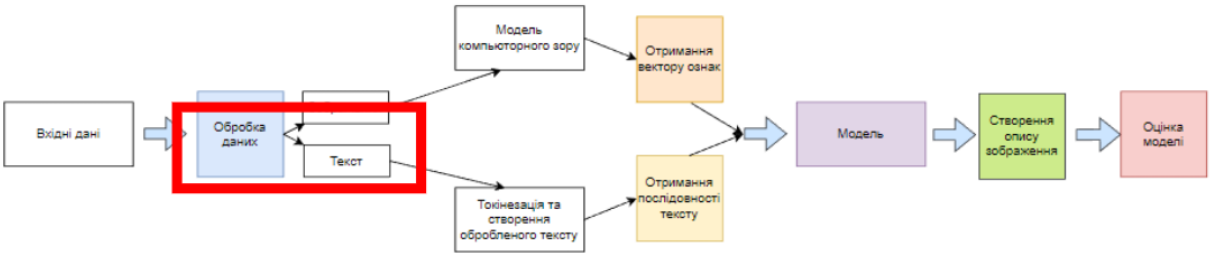

Рисунок 3.5 – Обробка тексту

Якщо роздивитись обробку зображення (рис. 3.6), то для моделі важливо врахувати наступні етапи: зміна розміру, нормалізація, вилучення функцій. Зміна розміру є необхідним кроком, оскільки дані можуть мати різні розміри. Тому завжди необхідно змінювати зображення до єдиного розміру. Розмір зображення обирається виходячи з архітектури CNN і наявних обчислювальних ресурсів. Загальні розміри для підписів до зображень – 224x224, 256x256 або 299x299 пікселів. Змінити розмір можна за допомогою методів інтерполяції, таких як білінійний або найближчий сусід. Нормалізація теж є невід'ємним кроком, яка забезпечує надійність роботи архітектури CNN із зображенням. Після зміни розміру значення пікселів зображення нормалізуються до діапазону між 0 і 1. Це важливо,

оскільки кодер CNN очікує вхідних значень у цьому діапазоні. Нормалізацію можна виконати, поділивши значення пікселів на 255, що є максимальним значенням, яке може мати піксель у 8-бітному зображенні. Крім того, значення пікселів можна нормалізувати за допомогою інших методів, таких як віднімання середнього значення та ділення на стандартне відхилення. Для архітектури отримання опису до зображення важливим кроком буде вилучення ознак. Вхідне зображення проходить через кілька згорткових шарів, які застосовують фільтри до зображення та витягують карти функцій. Потім карти функцій зменшуються за допомогою об'єднання шарів, щоб зменшити розмір карт функцій. Вихід кодера CNN є вектором ознак, який представляє зображення. Потім цей вектор подається в мовний декодер для створення підпису для зображення.

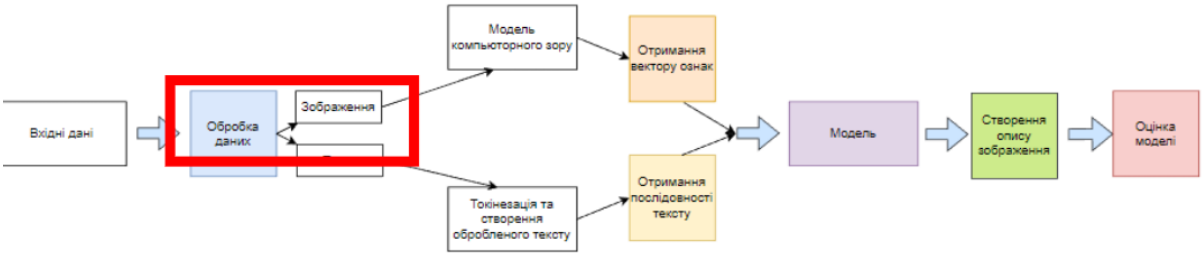

Рисунок 3.6 – Обробка зображення

Передбачається, що модель підпису до зображення зазвичай складається з двох частин: кодера зображення та декодера мови. Кодер зображень – це згорточна нейронна мережа (CNN), яка обробляє вхідне зображення та виділяє відповідні характеристики. Декодер мови – це рекурентна нейронна мережа (RNN), яка генерує заголовок слово за словом. Вихідні дані CNN подаються в RNN, який потім передбачає наступне слово в підписі на основі попередніх слів і особливостей зображення. Процес повторюється для кожного слова в підписі, доки не буде згенеровано маркер кінця речення.

3.3.1 Обробка тексту

Підписом до зображеення можно важати послідовність слов зібрані у речення природною мовою. Кінцевою метою проблеми підписів до зображень є створення підписів для нових зображень, не включених до навчального набору даних. Розроблений нами концепт моделі у якому можно побачити етап реалізації обробки текста перед роботою моделлю на рисунку 3.7.

Рисунок 3.7 – Обробка тексту перед тренуванням моделі.

Таким чином, завдання генерації титрів можна розглядати як завдання прогнозування послідовності, яке передбачає вивчення мовної моделі. Щоб створити такі моделі, нам потрібно перетворити текст у формат, який можна оптимізувати, що вимагає представлення мовних речень у вигляді числових векторів. Ці вектори забезпечують числове представлення мовних символів.

По-перше, нам потрібно визначити рівень деталізації, який використовується для представлення мовних символів. Однією зі стратегій є розглядати кожне слово як унікальну сутність, як запропоновано Солтоном та ін. 1975 року . На іншому кінці спектру знаходиться стратегія передбачення одного символу за раз, як запропоновано Лінгом та ін. 2015 року .

Процес створення підписів для нових зображень, які не є частиною навчального набору даних, передбачає створення коротких речень природною мовою. Щоб досягти цього, повинні спочатку перетворити текст у формат, який можна оптимізувати, що передбачає представлення речень у вигляді числових векторів. Це вимагає токенізації підписів, що включає їх поділ на окремі слова з видаленням знаків пунктуації та неалфавітно-цифрових символів. Потім цим словам присвоюються числові індекси на основі їх частоти в корпусі, і створюється спеціальний словник, який включає маркери для початку та кінця речень, а також маркер для невідомих слів і маркер заповнення для речень різної довжини. Цей словник обмежений, щоб зменшити складність і підвищити ефективність, і застосовується до всіх підписів у наборі навчальних даних.

3.3.2 Обробка зображення

Робота із зображенням – це завжди було завданням комп'ютерного зору. Для того щоб працювати із зображенням було розроблено багато методів та практик які полегшують роботу із зображенням На розробленій моделі, яку ми розробили, ми маємо архітектуру комп'ютерного зору (рис. 3.8), яку у даному параграфі будемо розглядати.

Рисунок 3.8 – Представлення моделі комп'ютерного зору

Для роботи із зображенням було розроблено CNN. Даний метод позволяет працювати із зображенням для таких задач, як классификация зображення, виявлення об’єкті та іншу задачі пов’язані ііз комп’ютерним зіром.

Архітектура CNN завжди складається із кількох рівнів. А саме згорткові , об’єднані та з’єднані шари. Для виділення локальних особливостей використовують набір фільтрів до вхідного зображення. зменшити розмірність даних і забезпечити інваріантність трансляції використовують шари об’єднання. Тим самим зменшують дискретизацію карт функцій. В кінці мережі використовують повність з'єднані шари, які класифікують зображення на основі вилучених ознак (рис. 3.9)

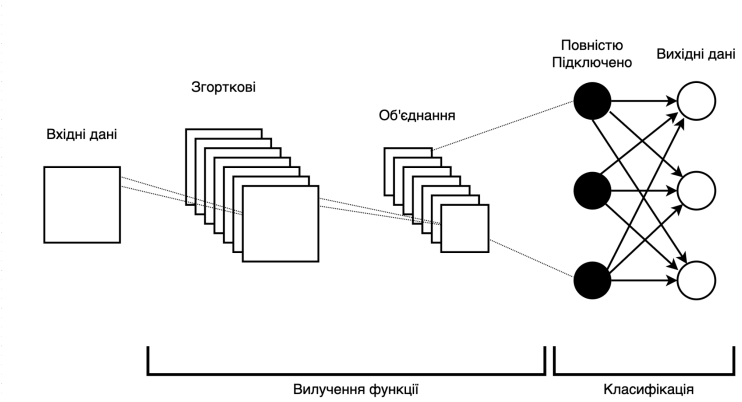

Рисунок 3.9 – Схема згорткової нейроної мережі

Зображення для задачі підпису теж необхідно опрацювати, щоб можно було поясниті машині, що нам необхідно зробити. Тому для знашої задачі необхідно отримати візуальни характеристики які необхідні декодеру. Це можно зробити а допомогою архітектури CNN. Для того щоб зобрадення можно було передати у CNN його необхідно попередньо обробляти. Це зміна розміру, форми, та специфічні налаштування. Деякі моделі використовують зображення зі значеннями в діапазоні від 0 до 1, інші від -1 до 1.

Розробляти та навчати CNN заново це дуже складний процес який потребує багато часу, розробки та налаштувань, тому було розроблено багато CNN моделей, які можуть попередньо обробляти зображеня. В таблиці 3.1. представлено характеристики найпопулярніших архітектур, які знаходяться у відкритому доступі, та навчанні на наборі даних ImageNet.

Таблиця 3.1 – Підсумок популярних мереж ConvNet, попередньо навчених на наборі даних ImageNet

Моделі	Розмір зображен-ня	Розмір весів	Топ -1 acc.	Топ -1 acc.	Параметри	Гли-бина
Xception	299 x 299	88 MB	0.790	0.945	22,910,480	126
VGG16	224 x 224	528 MB	0.715	0.901	138,357,544	23
VGG19	224 x 224	549 MB	0.727	0.910	143,667,240	26
ResNet50	224 x 224	99 MB	0.759	0.929	25,636,712	168
InceptionV2	299 x 299	92 MB	0.788	0.944	23,851,784	159
InceptResNet V2	299 x 299	215 MB	0.804	0.953	55,873,736	572
MobileNet	224 x 224	17 MB	0.665	0.871	4,253,864	88

Xception (Extreme Inception) – це модель глибокого навчання, розроблена командою дослідників з Google у 2016 році. Вона є модифікацією більш ранньої моделі Inception v3 і була створена з метою покращення швидкості та точності класифікації зображень. Основна ідея полягає у використанні внутрішніх згорткових шарів, що дозволяє зменшити кількість параметрів та знизити обчислювальну складність. При цьому точність моделі залишається на високому рівні. Плюсом Xception є висока швидкість та точність класифікації зображень, а також невелика кількість параметрів моделі. Мінусом може бути складність з реалізацією та вимогливість до обчислювальних ресурсів.

VGG16 і VGG19 були розроблені Групою візуальної геометрії (VGG) Оксфордського університету в 2014 році. Обидва вони базуються на одній архітектурі, яка складається з кількох згорткових шарів, за якими слідують повністю з’єднані шари. Ключова відмінність між двома моделями полягає в тому, що VGG19 має більше шарів, ніж VGG16. Обидві моделі досягли високої точності під час завдання ImageNet, але VGG19 потребує більше часу та ресурсів на обчислення через більший розмір.

ResNet50: ResNet (Residual Network) – це глибока нейромережа, яка була розроблена в 2015 році Microsoft Research . Вона має 50 шарів і використовує "residual connections", що дозволяє зменшити проблему зникнення градієнту. ResNet50 є дуже ефективною для багатьох завдань, включаючи розпізнавання образів та відео.

InceptionV3: Inception є глибокою нейромережею, розробленою в 2015 році дослідниками Google , яка має декілька версій. InceptionV3 – це третя версія, яка використовує 3x3 та 5x5 свертки замість стандартних сверток, що дозволяє отримати кращі результати для більш складних завдань.

Inception-ResNetV2 є глибокою згортковою нейромережею, що поєднує в собі інноваційний Inception блок та ResNet архітектуру. Ця модель була розроблена в 2016 році компанією Google Brain. Вона є однією з найкращих моделей для класифікації зображень, з найвищою точністю на даний момент. Архітектура Inception-ResNetV2 містить багато глибоких згорткових шарів та максимальну кількість параметрів.Головним плюсом є найвища точність серед згорткових нейромереж, висока ефективність завдяки використанню Inception та ResNet архітектур.Мінусом є вимоги до обчислювальної потужності, багато параметрів, потребує багато даних для навчання.

MobileNet є згортковою нейромережею, призначеною для використання на мобільних пристроях та на пристроях з обмеженим ресурсами. Модель була розроблена в 2017 році компанією Google з метою

зменшення розміру нейромережі та збереження точності. Архітектура MobileNet базується на згорткових шарах з груповою згорткою та глибокими згортковими блоками, що дозволяє зменшити розмір нейромережі та кількість параметрів. Головним плюсом є мала кількість параметрів, швидка та легка модель, що дозволяє використовувати її на мобільних та вбудованих пристроях. Мінусом є менша точність порівняно з більш великими моделями, що були описані вище.

Представлені моделі можно використати для вилучення вектору ознак тавикористати для тренування моделідля опису зображення.

Перед навчанням можна використати моделі та закодувати зображення в карти ознак, які можна зберігати на диску і використовувати пізніше. Цей процес передбачає використання ознак, отриманих з останнього шару згортки попередньо навченої мережі CNN, за винятком повністю з'єднаних частин, які використовуються для класифікації.

Отримані карти ознак відрізняються за розміром залежно від використовуваної CNN і кодуються як сітки, що складаються з ділянок зображення з певними розмірами, такими як ширина W, висота H і глибина D. Глибиною вважають кількість функцій на ділянку та дорівнює вона кількості каналів останнього шару згортки. InceptionV3 створює карти функцій із формою (8 × 8 × 2048).

Зрештою, ознаки зображення пропускаються через повністю з'єднаний шар для отримання 2D тензора. Вибір попередньо вивченої мережі CNN для використання є гіперпараметром моделі і може мати значення «vgg16» , «resnet50» , «inception» .

3.4 Існуючі рішення моделей для задачи створення підписів

У розглянутій концепції нашої моделі ми можемо побачити (рис. 3.10), що після обробки усіх даних, та представлення їх у вигляді,

який розуміє комп'ютер можно переходити до обговорення методів реалізації самої моделі.

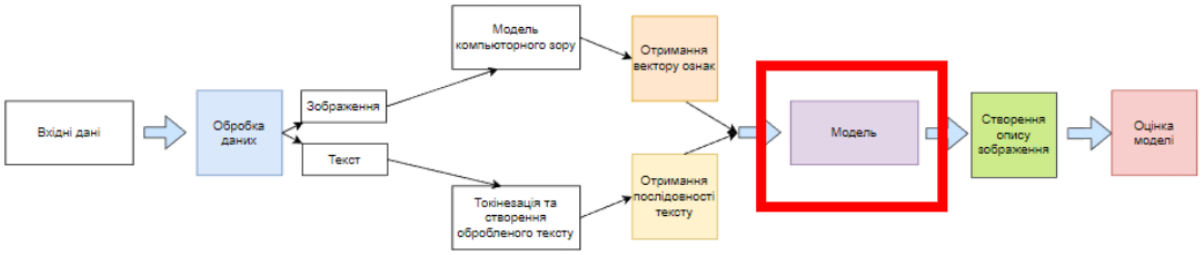

Рисунок 3.10 – Демонстрація етапу моделювання

3.4.1 Мультимодальне навчання

Методи пошуку та підписування зображень на основі шаблонів накладають обмеження на здатність описувати зображення у формі шаблонів, структурованих передбачень та/або синтаксичних дерев. Удосконалення нейронних мереж принесло нові підходи, які можуть подолати ці обмеження. Ці техніки можуть створювати речення з багатшою структурою, виразністю та гнучкістю. Мультимодальні моделі нейронної мови є одним із підходів до цієї проблеми з точки зору навчання. Загалом ці моделі є двонаправленими, тобто вони здатні генерувати нові підписи зображень, але їх також можна застосовувати до завдань пошуку зображень і речень.

Загальна структура мультимодального нейронного навчання показана на рисунку 3.11. По-перше, функції зображення витягуються за допомогою екстрактора функцій (зазвичай CNN). Витягнуті ознаки потім надсилаються до нейронної моделі, яка відображає ознаки зображення в загальному просторі слів. Нарешті, модель передбачає нові слова на основі характеристик зображення та попередньо згенерованих контекстних слів.

У роботі, яка була представлена у 2014 році Кіросом та ін. [7] було використано мультимодальні нейронно мовні моделі (MNLM), іншими словами моделі природної мови, які можуть залежати від інших модальностей. Автори представили дві адаптовані моделі Лог-білінійної мовної моделі, запропонованої Mnih і Hinton 2007 року [8]. У випадку моделювання тексту на основі зображення можно спільно вивчити представлення слів і особливості зображення, навчаючи моделі разом із CNN, такі як «AlexNet» .

Рисунок 3.11 – Архітектура мультимодальні нейронної мережі для підпису зображення

Mao та ін. 2014 року [9] та Mao і Yuille 2015 року [10] представили модель мультимодальної рекурентної нейронної мережі (m-RNN) для створення підписів до зображення. Він моделює розподіл ймовірностей словотворення безпосередньо з попередніх слів і зображень. і використовуйте цей розподіл для створення підписів. Модель складається з двох підмереж: глибокої RNN для речень і глибокої CNN для зображень. Ці дві підмережі взаємодіють одна з одною на мультимодальному рівні, щоб сформувати всю модель m-RNN. Модель m-RNN також можна використовувати для вилучення зображень або речень. Ця модель забезпечує найсучасніші результати в підписах до зображень.

У роботі представленій Карпаті та Фей-Фей 2015 року [11] пропонують інший мультимодальний підхід у якому пропонують

використовувати глибокі нейронні мережі. У роботі вони пропонубть вбудовувати зображення та текст природної мови, де завданням є двонаправлений пошук зображень та речень. Дний метод дає можливість працювати на більш високому рівні з фрагментами зображень та речень. Запропонований метод застосовує Region-CNN, щоб розбити зображення на кілька об’єктів, RNN над реченнями, представленими зв’язками дерева залежностей (DTR)

У роботі Чен і Зітнік 2015 року [12] вони також розглядали завдання двонаправленого пошуку для опису зображення. У цьому підході використовували RNN яка динамічно намагалась зробити представлення зображення на створення текстового опису. Модель намагалась надовго запам'ятовувати візуальні особливості. Таким чином модель може створювати підписи до зображення, та і навпаки реконструювати зображення на основі опису.

3.4.2 Передавальне навчання

У машинному навчанні часто застосовують техніку передавального навчання, яка полягає у повторному використанні попередньо навченої моделі для нової задачі. Це дозволяє використати знання, отримані від попереднього навчання, для покращення узагальнення на нові дані. Наприклад, якщо навчаємо класифікатор визначати, чи є на зображенні їжа, то можемо використати ці знання для розпізнавання напоїв. Це приклад перехідного навчання.

Головною ідеєю передавального навчання є використання знань, отриманих з попереднього завдання, для поліпшення узагальнення в іншому. Замість того, щоб починати навчання з нуля, будемо використовувати шаблони, отримані від попередніх пов’язаних задач.

Цей метод часто застосовується в задачах обробки комп’ютерного зору та природної мови, таких як аналіз настроїв, завдяки великій обчислювальній потужності.

Хоча передавалье навчання не є самостійною технікою машинного навчання, його можна вважати методологією проектування, яка застосовується в активному навчанні та інших галузях. Він є популярним у поєднанні з нейронними мережами, що потребують великі обсяги даних та обчислювальну потужність.

Передавальне навчання має декілька переваг, але основні з них полягають у скороченні часу на навчання, збільшенні продуктивності нейронних мереж (у більшості випадків) та необхідності у меншій кількості даних. Зазвичай для навчання нейронної мережі з нуля необхідно багато даних, але іноді доступ до таких даних обмежений. У такому випадку, передавалье навчання стає корисним і дозволяє створити надійну модель машинного навчання за допомогою значно меншої кількості навчальних даних, оскільки модель вже навчена. Це особливо корисно для обробки природної мови, де для створення великих наборів даних з мітками потрібні експертні знання. Крім того, передавальне навчання дозволяє скоротити час навчання, що особливо важливо для складних завдань, які можуть займати дні або навіть тижні для навчання глибокої нейронної мережі з нуля.

Є декілька підходів до передавального навчання, які використовуються у сучасній розробці. Перше це навчання моделі на повторному використанні. Друге використання попередньо навченої моделі Третє отримання функцій з моделі.

Задачі, які використовують підхід, де повторно використовують попередньо навчену модель, це коли мало даних для навчання. Розробники беруть за основу навчену модель, модифікують та вирішують свою задачу за допомогою свого набору даних. В обох задачах використовуються однакові вхідні дані. Можна повторно використовувати модель та

прогнозувати нові вхідні дані. Крім того, розглядають можливість змінювати та перепідготовлювати різні рівні, пов’язані із задачею, і вихідний рівень.

Другий підхід до вирішення проблеми полягає в застосуванні попередньо навченої моделі. На ринку існує велика кількість таких моделей, тому важливо провести дослідження та визначити, яка з них найбільш підходить для поставленої задачі. Кількість шарів, які слід повторно використовувати, та ті, які потрібно перенавчити, залежать від конкретної задачі.

Один із способів досягнення кращої продуктивності в машинному навчанні полягає в застосуванні глибокого навчання для виявлення найкращого представлення вашої проблеми, шляхом вилучення ознак та перенесення навчання. Цей підхід дозволяє автоматично витягувати функції, але все ж потребує деякої розробки функцій та знання домену. Однак, нейронні мережі можуть дізнаватися, які функції дійсно важливі, алгоритм навчання репрезентації може знайти хорошу комбінацію функцій для складних завдань. Крім того, вивчене представлення можна використовувати для інших проблем, просто використовуючи перші шари та проміжні рівні, як представлення необроблених даних. Цей підхід часто використовується в комп’ютерному зорі, що дозволяє зменшити розмір набору даних та зробити обчислення більш придатними для традиційних алгоритмів.

3.4.3 Змагальне навчання

Останніми роками спостерігається значний прогрес у дослідженні рекурентних нейронних мереж (RNN) та їхньої здатності генерувати речення. Однак речення, що генеруються, часто не мають варіативності через те, що ШНМ навчаються на реальних текстах і мають тенденцію до

їх точної імітації. Це обмеження ще більше посилюється метриками оцінювання, такими як BLEU та METEOR.

Генеративні змагальні мережі (Generative Adversarial Networks, GANs) пропонують потенційне вирішення цієї проблеми, вивчаючи глибокі особливості немаркованих даних і генеруючи більш різноманітні та чіткі підписи за допомогою конкурентного процесу між мережею-генератором і мережею-дискримінатором. Одним з таких методів є умовний ШН (Conditional GAN, CGAN), який спільно навчає генератор створювати описи на основі зображень, а оцінювач – оцінювати якість цих описів. Однак GAN стикаються з такими труднощами, як зворотне поширення та зникаючі градієнти, які можна вирішити, запровадивши механізми градієнтів політики та розгортання Монте-Карло.

Інший метод на основі GAN, запропонований Шетті та ін. 2017 року [17], може генерувати кілька підписів для одного зображення, використовуючи семплер Гумбела замість апроксимації функцією. Лі та ін. (2018)[18] пропонують структуру порівняльного змагального навчання під назвою "Мережа порівняльного змагального навчання (CAL)", яка оцінює якість підписів шляхом порівняння набору підписів у спільному просторі зображення та підпису. Цей підхід ефективно збільшує різноманітність згенерованих підписів у всьому корпусі.

3.4.4 Модель кодера-декодера

Архітектура кодера-декодера (рис. 3.12) була розроблена для перекладу речень між різними мовами[13]. Тому наткнувшись самою архітектурою багато дослідників вирішили використовувати даний підхід і для підпису зображень. Головною ідеєю для адаптації цієї задачі було у змінені модуля у задачі.

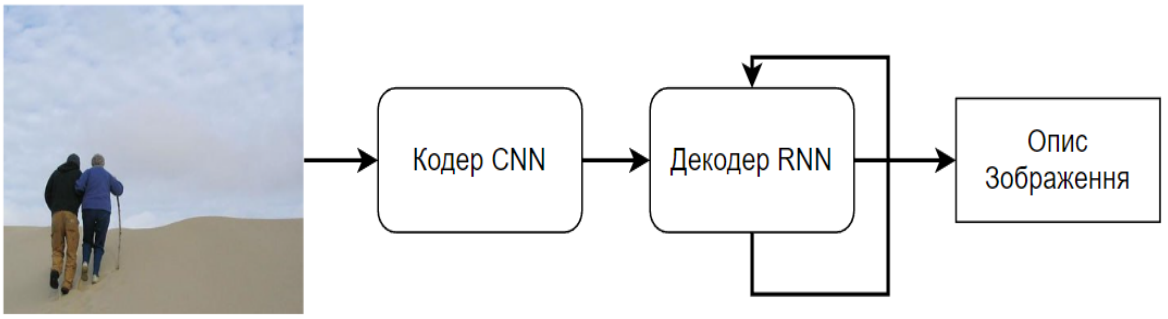

Рисунок 3.12 – Архітектура підпису для зображень на основі моделі кодеру-декодеру

Першими хто використав ідею кодера-декодера були Кірос та ін. [14]. Головною ідеєю було отримувати на вході зображення, а на виході генерувати речення слово за словом як при перекладі. Їх модель використовує довготривалу короткочасну пам'ять (LSTM) для кодування текстових даних і CNN для кодування візуальних даних. Потім, шляхом оптимізації втрат попарного ранжування, закодовані візуальні дані проектуються у простір вбудовування, що охоплює приховані стани LSTM, які кодують текстові дані.

Для створення реалістичних підписів до зображень було використано Structure-Content Neural Language Model (SC-NLM) – мультимодальну нейромовну модель, яка декодує візуальні ознаки за допомогою контекстуальних векторів ознак слів та генерує речення слово за словом. Метод, запропонований Вінялсом та іншими в 2015 році [15], є подібним до методу Кіроса, але використовує CNN для представлення зображень та LSTM для генерації підписів до зображень.

Проблемою попередніх методів кодування-декодування є їхня чутливість до проблеми зникаючого градієнта, через те що інформація про зображення подається лише на початку процесу, тому підписи мають тенденцію до погіршення релевантності при генеруванні довгих речень.

Для подолання цієї проблеми, Донахью та ін. запропонували модель довготривалої рекурентної згорткової мережі (LRCN), яка може обробляти статичні та динамічні зображення та послідовності, використовуючи копії статичного зображення та попереднього слова безпосередньо як вхідні дані. Крім того, деякі дослідники пропонують методи розширення кодувально-декодувальної структури за рахунок включення високорівневих семантичних елементів.

Дослідники пропонують удосконалити модель LSTM, використовуючи семантичну інформацію з зображень як додатковий вхідний сигнал. Це дозволяє моделі зосередитися на більш релевантних рішеннях, особливо під час генерування довгих речень. Семантична інформація може бути витягнута різними способами, наприклад, використовуючи кросмодальне пошукове завдання або мультимодальний простір вбудовування. Крім того, автори досліджують різні стратегії нормалізації довжини для пошуку за променем, щоб уникнути упередженості в бік коротких речень.

Ву та ін. [16] використовують семантичні концепції високого рівня в структурі кодера-декодера для створення субтитрів до зображень. Вони використовують набір семантичних атрибутів та класифікатор з кількома мітками, щоб передбачити ймовірність наявності об’єктів на зображенні. Для створення підпису вектор передбачення передається до LSTM. Можуть використовуватися різні мовні моделі для різних завдань, включаючи мультимодальну мовну модель на основі регіону. Модель також може включати зовнішню семантичну інформацію.

Для підходу кодера-декодера обмеженням більшості розглянутих підходів може стати обмежена доступність підписів для частини зображень.

3.4.5 Модель з механізмом уваги

Для опису зображень необхідно використовувати короткі, але інформативні підписи, які відображають найбільш вагомі елементи зображення та уникають деталей. Одним із підходів до генерації описів зображень є використання механізму уваги, що ґрунтується на механізмах візуальної уваги приматів та людей. Інтеграція уваги до рамки кодувальник-декодувальник дозволяє згенерувати речення, яке буде залежати від прихованих станів, що обчислюються на основі механізму уваги.

Загальна структура методів опису зображень за допомогою механізму уваги показана на рисунку 3.13. У таких методах використовуються механізми уваги, що базуються на різних типах сигналів з вхідного зображення, для того, щоб при генерації опису зображення фокусуватись на конкретних аспектах вхідного зображення. Успіхи інших задач, які використовують механізм уваги у роботі Mnihта ін [24]), були додатковою мотивацією для використання цього підходу.

Рисунок 3.13 – Архітектура для підпису зображень на основі кодеру декодеру, та механізму уваги

Перші, хто використали механізм уваги для автоматичного підписування зображень, були Lei Bata ін. та Xuta ін. [17]. Їхня модель, яка є варіацією на тему зразка кодування-декодування, може динамічно уважати на найважливіші ділянки зображення під час генерації опису. Більшість підходів, що використовують кодувальники зображень на основі CNN, використовують верхній шар ConvNet для вилучення ознак зображення, але цей метод генерує контекстні вектори, використовуючи ознаки, які були вивчені на більш низьких шарах ConvNet. Ідея полягає в тому, що використання верхнього шару може призвести до втрати деталей, які можуть бути корисними для генерації описів зображень. Xuta ін. спробували дві різні техніки для симуляції уваги: стохастичну тверду увагу та детерміновану м'яку увагу.

Багато методів, використовуваних у підписуванні зображень, можна класифікувати як низ вгору або верх донизу. У верхньому підході (Donahue та ін.[25]; Karpathy and Fei-Fei [11]; Chen and Zitnick [26]; Mao and Yuille, 2015; Mao та ін. [27]; Vinyals ін. [28]) спочатку вилучають візуальні ознаки та використовують їх для вибору або генерації підпису. У нижньому підході (Farhadi та ін. [29]; Kulkarni та ін. [30]; Li та ін. [31]; Kuznetsova та ін. [32]; Elliott and Keller [33]; Lebret та ін [34]) спочатку вилучають візуальні концепти (наприклад, регіони, об'єкти та атрибути) та пізніше поєднують їх для генератора.

Янг та ін. [35] пропонують мережу перегляду, як вирішення проблеми відсутності можливостей глобального моделювання в моделях кодерів-декодерів, з використанням механізму уваги. Мережа з механізмом уваги складається з модуля уваги, який виконує кілька кроків перегляду з механізмом уваги на прихованих станах кодера для генерації векторів думок, які фіксують глобальні властивості зображення. Ці вектори використовуються як вхідні дані для механізму уваги в декодері. На етапі виконання механізму уваги можно отримати інформацію про об'єкти, їхнє взаємне розташування та загальний контекст зображення.

Педерсолі та ін. [36] пропонують механізм уваги на основі області для підписів до зображень, який дозволяє встановити прямий зв'язок між словами підпису та областями зображення. Вони представляють мережі просторового перетворення, які дозволяють створювати зони уваги, специфічні для зображень, і можуть навчатися разом з рештою мережі.

Лу та ін. [37] пропонують адаптивний механізм уваги, який динамічно визначає, коли дивитися на зображення, а коли покладатися на мовну модель для генерації наступного слова. Ця модель поєднує в собі нову просторову модель, яка визначає, куди дивитися, і сторожові ворота, які вирішують, чи дивитися на зображення, чи покладатися на візуальний сторож для генерації підпису. Ці механізми уваги покращують гнучкість і виразність моделей глибокого навчання для створення підписів до зображень і містять релевантну інформацію для покращення обробки природної мови в задачах, пов'язаних із зображеннями та природною мовою.

3.4.6 Модель трансформер

Модель трансформер, як кодер-декодер були розроблені для завдань обробки природної мови. Створення даної моделі відкрила нові перспективи у вирішенні багатьох завдань.

Для вирішення завдань підпису зображень було розроблено багато підходів на основі трансформер моделей.Для адаптації моделі Transformer для субтитрів зображень було запропоновано кілька підходів. Одним із найпопулярніших підходів є використання комбінації Vision Transformer (ViT) і моделі Transformer. Модель ViT спочатку попередньо навчена на великому наборі даних зображення, щоб витягти візуальні характеристики з вхідного зображення. Потім ці візуальні функції вводяться в модель Transformer, яка генерує відповідний заголовок.

Інший підхід передбачає використання CNN для вилучення візуальних характеристик із вхідного зображення, які потім подаються в модель Transformer для генерації підпису. Візуальні характеристики, отримані з CNN, можуть бути додатково оброблені за допомогою Multi-Head Self-Attention для захоплення більш детальної візуальної інформації.

Модель Transformer для субтитрів до зображень була ретельно вивчена та оцінена на кількох контрольних наборах даних, таких як COCO, Flickr30k і Visual Genome. Він продемонстрував надзвичайну ефективність порівняно з іншими найсучаснішими моделями, особливо щодо створення описових і змістовних підписів.

3.4.7 Мета-навчання LSTM

LSTM (Long Short-Term Memory) – це тип рекурентної нейронної мережі, яка використовується в одномоментних методах навчання для мультимодальних підписів зображень даних. LSTM корисні, оскільки вони можуть обробляти послідовні дані, такі як мова, і створювати підписи для зображень.

У контексті навчання для підписів до зображень LSTM можна використовувати кількома способами. Одним із підходів є використання LSTM для кодування текстового опису зображення, а потім поєднання цього кодування з візуальними характеристиками зображення для створення підпису. Інший підхід полягає у використанні LSTM для створення підпису слово за словом, де кожне слово залежить від попередніх слів і візуальних особливостей зображення.

Ідея мета-навчання LSTM полягає в тому, щоб навчити cell LSTM вивчати правило оновлення для нашого вихідного завдання. З точки зору

структури мета-навчання, cell LSTM буде використовуватися як мета-навчання.

Щоб реалізувати метанавчальний LSTM для однократного підписування зображень, необхідно наступне.

Першим кроком буде підготовка пар даних зображення-підпис. Далі обробити дані наступним чином. Зміна розміру та нормалізацію зображень, а також токенізацію та додавання підписів.

Після цих кроків можемо навчити мета-навчальний LSTM на попередньо оброблених парах зображення і підписів, використовуючи, наприклад, функцію перехресних ентропійних втрат і оптимізацію стохастичного градієнтного спуску (SGD). Під час навчання можно використати методи доповнення даних, такі як випадкове обрізання та перевертання, щоб покращити продуктивність узагальнення моделі.

LSTM зберігають усю попередню інформацію за допомогою різних воріт. Існують різні варіації стохастичного градієнтного спуску (SGD), такі як momentum, RMSprop, Adam та багато інших, які по суті зберігають інформацію про минуле навчання (у вигляді градієнтів) для кращої оптимізації. Тому, логічно, cell LSTM можна розглядати як стратегію кращої оптимізації, яка дозволяє моделі фіксувати знання як про короткострокову перспективу конкретної задачі, так і про загальну довгострокову перспективу.

Як мета-навчальний LSTM завершен, маємо змогу використати його для генерації підписів до нових зображень з того ж завдання. Під час генерації підписів можемо використовувати пошук по променю, щоб згенерувати кілька підписів і вибрати найкращий з них на основі оцінки мовної моделі.

Розгляд Архітектурі мета-навчального LSTM можно почати з методу оновлення градієнтного спуску.

$$\theta_t = \theta_{t-1} - \alpha_t \nabla L_t, \tag{3.1}$$

де θ_t – параметр на часовому кроці t;

α_t – швидкість навчання в момент часу t;

∇L_t – градієнт втрат в момент часу t.

Тоді, як cell LSTM виглядає наступним чином.

$$c_t = f_t \odot c_{t-1} + i_t \odot \bar{c_t}. \tag{3.2}$$

Це оновлення виглядає дуже схожим на те, як оновлюються cell в LSTM. Автори мета-навчання LSTM припустили, що якщо підставити в рівняння оновлення cell наступні значення, то отримаємо правило оновлення за градієнтним спуском:

$$f_t = 1 \mid c_{t-1} = \theta_{t-1} \mid i_t = \alpha_t \mid \bar{c_t} = \nabla L_t. \tag{3.3}$$

Враховуючи це, логічно, що навчають у даній моделі i_t, оскільки це аналогічно оцінці швидкості навчання градієнтного спуску. Отже, мета-навчальник LSTM визначає i_t наступним чином:

$$i_t = \sigma(W_I [\nabla L_t, L_t, \theta_{t-1}, i_{t-1}] + b_I). \tag{3.4}$$

Як бачимо, i_t визначається як сигмоїдна функція з комбінацією поточного градієнта, поточних втрат і попереднього навчання швидкість,

i_{t-1}.

Для f_t вона має дорівнювати 1, але щоб уникнути проблем зі зменшенням градієнтів, її було визначено наступним чином:

$$f_t = \sigma(W_{F.}[\nabla L_t, L_t, \theta_{t-1}, f_{t-1}] + b_F). \quad (3.5)$$

Як бачимо, f_t визначається як сигмоїдна функція з комбінацією градієнта, поточних втрат та gete forget. Якщо придивитися уважніше, то i_t іf_t були обрані як функція поточного градієнта та поточних втрат. Це було зроблено навмисно, щоб мета-навчання міг контролювати швидкість навчання, щоб навчити основну модель за менший час.

3.5 Методи аналізу для отриманих результатів

Після створення кінцевого результату у вигляді моделі, яка може створювати підпис до зображення ми підійшли до кінцевого етапу, а саме оцінці результату, це видно на рисунку 3.14.

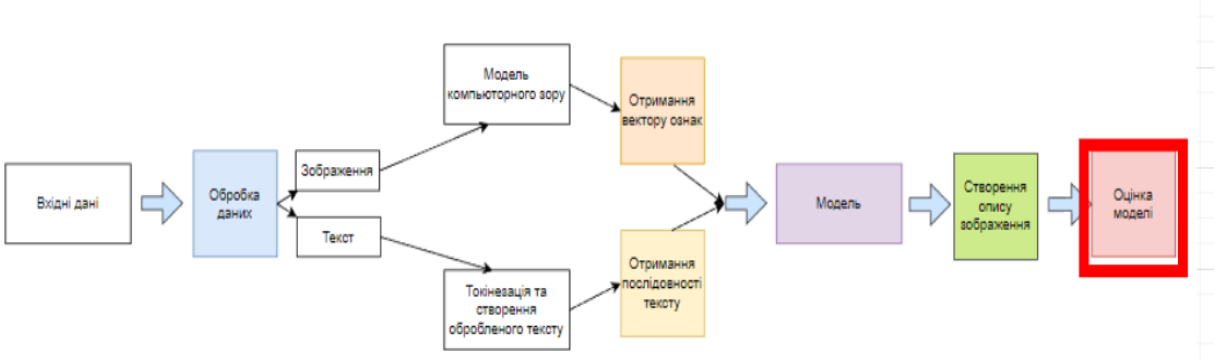

Рисунок 3.14 – Демонстрація етапу оцінки моделі

3.5.1 BERT score

BERT score – це метод автоматичного оцінювання згенерованого тексту природної мови.Метою цього методу полягає у тому, щоб оцінювати семантичну єквівалентність. велика кількість людей використовує метрику BLEU-n, оскільки є найпоширенішою метрикою машинного перекладу, просто підраховує n-грамове перекриття між кандидатом і еталонним. Хоча це забезпечує просту та загальну міру, воно не враховує лексичне та композиційне розмаїття, що зберігає значення.

BERT score вимірює подубність між єталоним реченням та згенерованим реченням, за основу для оцінки він використовуючи контекстуалізовані вбудовування токенів та обчислюючи суму косинусних подібностей між ними. Даний підхід вирішує дві найпоширеніші проблеми метрики оцінки тексту. По перше він не штрафує речення, де є семантично правильні фрази за допомогою виявлення парафрази та уникає не дооцінки речення. По-друге, контекстуалізовані вбудовування токенів можуть вловлювати віддалені залежності та покарати семантично критичні зміни порядку. BERT score показав добрі результати в експериментах з машинним перекладом та створенням підписів до зображень, корелюючи дуже сильно з людськими оцінками.

Архітектура BERT score порівнює речення x та речення кандидат $\hat{x}$. Кожне речення подається в попередньо підготовлену модель BERT для генерування контекстних вставок для кожного слова. Після отримання вкладень для всіх слів оцінка подібності обчислюється між кожним словом у еталонному реченні та кожним словом у реченні-кандидаті, що призводить до обчислення n-квадрат.Пара слів із найбільшою подібністю вибирається та використовується для обчислення recall, precision, і F1-оцінки (що є середнім гармонійним значенням точності та запам’ятовування).

Recall розраховується за формулою:

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} max_{\widehat{x_j} \in \widehat{x}} x_j^T \widehat{x_j} . \quad (3.6)$$

Precision розраховується за формулою:

$$P_{BERT} = \frac{1}{|\widehat{x}|} \sum_{\widehat{x_j} \in \widehat{x}} max_{x_i \in x} x_j^T \widehat{x_j}. \quad (3.7)$$

F1-оцінка розраховується за формулою:

$$F_{BERT} = 2\frac{P_{BERT} * R_{BERT}}{P_{BERT} + R_{BERT}} . \quad (3.8)$$

У деяких випадках рідкісні слова можуть вказувати на схожість речень. Тому з наведеними вище рівняннями BERTScore можна використовувати інверсну частоту документа (IDF). Це необов'язково, і залежно від домену тексту та доступних даних це може або не може сприяти кінцевим результатом. Таким чином, більше розуміння зважування важливості є відкритою областю дослідження. Враховуючи M посилальних речень {x (i)}M та i=1, оцінка IDF лексеми W дорівнює

IDF-оцінка розраховується за формулою:

$$idf(w) = - log \frac{1}{M} \sum_{i=1}^{M} I[w \in x^{(i)}] , \quad (3.9)$$

де I[·] – індикаторна функція. Не використовується повна міра tf-idf, оскільки обробляємо окремі речення, де частота терміну (tf), швидше за все, дорівнює 1.

3.5.2 BLEU-n

BLEU від IBM – це набір показників, створений для заміни людських суддів у оцінюванні завдань МТ, тож окрім того, що він швидкий, недорогий і не залежить від мови, він був розроблений, щоб продемонструвати хорошу кореляцію з людське судження. У цій групі показників використовується середньозважене значення збігів фраз різної довжини порівняно з перекладами, написаними людьми, щоб визначити їх близькість. BLEU починається з базової метрики, з якої виводиться низка метрик для різних розмірів n-грамів: BLEU-1 порівнює кандидатське речення з реченнями в уніграмі, BLEU-2 порівнює кандидатське речення з реченнями в біграмі і так далі, поки BELU-4, який має найкращу кореляцію з людськими судженнями. Тоді як уніграмні оцінки враховують адекватність, вищі n-грамові бали враховують вільне мовлення. BLEU популярний, тому що це була перша метрика, популяризована для завдань МП, задовго до того, як почалися дослідження поля автоматичного створення зображень. Хоча BLEU показує розумну кореляцію з людськими судженнями, здається, він працює добре лише тоді, коли згенерований текст є сортованим і може навіть статися, що збільшення балу BLEU не відповідає збільшенню якості.

Оцінка BLEU обчислюється на основі точності n-грамів, де n – це довжина n-грамів, суміжна послідовність з n слів (зазвичай від 1 до 4), між машинним перекладом і еталонним перекладом, тоді як стисліть вимірює, наскільки добре машинний переклад приблизно відповідає довжині еталонного перекладу. Формула для обчислення точності n-грам має такий вигляд:

$$precision\_n = \frac{(number\ of\ n-gram\ matches)}{(total\ number\ of\ n-grams)}. \quad (3.10)$$

Загальна кількість n-грамів – це кількість n-грамів у створеному машиною тексті, тоді як кількість збігів із n-грамами – це кількість n-грамів у створеному машиною тексті, які також з'являються в посиланні.

Щоб обчислити бал BLEU, точність n-грамів об'єднується в середнє геометричне таким чином:

$$BLEU = brevity - penalty * \frac{exp(sum(log(precision\_i))}{N}, \quad (3.11)$$

де brevity - penalty = 1, якщо довжина машинного перекладу більша за довжину еталонного перекладу, і exp(1 - (довжина еталонного перекладу / довжина машинного перекладу)) в іншому випадку;

precision_i – модифікована n-грамова точність для n-грами;

N – кількість n-грам у машинному перекладі.

Наприклад, якщо хочемо обчислити оцінку BLEU-4 для машинного перекладу, обчислимо точність для кожного збігів 1-грам, 2-грамів, 3-грамів і 4-грамів, а потім об'єднаємо переведіть їх у середнє геометричне за наведеною вище формулою.

3.5.3 METEOR

METEOR – це ще один показник, спочатку розроблений для оцінки завдань МП. Він базується на 44 сучасному узагальненому понятті відповідності уніграми між машинним перекладом і створеним людиною довідковим перекладом. Уніграми можна зіставляти на основі їх

поверхневих форм, форм основи та значень. Спочатку він виконує узагальнені уніграмні відповідності між реченням-кандидатом і реченням, написаним людиною, а потім обчислює оцінку на основі результатів відповідності. Обчислення балів передбачає точність, запам’ятовування та вирівнювання зіставлених елементів, а у випадку кількох посилальних речень найкращий бал серед усіх можливих збігів береться як остаточний бал оцінки. METEOR забезпечує кращу кореляцію на рівнях речень або сегментів. Існує універсальна версія цього показника, яка підтримує багато мов, наприклад українську або гінді.

Оцінка METEOR коливається від 0 до 1, причому більша оцінка вказує на кращий переклад. Оцінка 1 означає, що створений машиною переклад ідентичний еталонному перекладу.

Формула оцінки METEOR така:

$$METEOR = (1 - \alpha) * P + \alpha * R * F\_mean, \qquad (3.12)$$

де α – це параметр, який контролює баланс між точністю та запам’ятовуванням. Зазвичай його встановлюють на 0,5;

P – це показник точності, який вимірює, скільки слів у машинно створеному перекладі збігається з еталонним перекладом;

R – це показник запам’ятовування, який вимірює, скільки слів у еталонному перекладі охоплено машинним перекладом;

F_mean – це гармонічне середнє значення точності та запам’ятовування, яке обчислюється як (2 * precision * recall) / (precision + recall), яка штрафується за помилки порядку слів шляхом застосування коефіцієнта штрафу, що називається оцінкою вирівнювання, яка базується на кількості відповідностей слово до слова між машинним перекладом і еталонним перекладом.

Оцінка вирівнювання розраховується таким чином:

$$align\_score = 1 - \frac{(num\_insertions + num\_deletions + num\_substitutions)}{num\_words}, \quad (3.13)$$

де num_insertions, num_deletions, and num_substitutions – це кількість вставок, видалень і замін, необхідних для перетворення машинно створеного перекладу в еталонний переклад;

num_words – це загальна кількість слів у еталонному перекладі.

3.5.4 ROUGE

ROUGE – це набір показників, представлений як альтернатива BLEU для роботи з більшими текстами, як-от текстові резюме. Цей показник використовує найдовшу загальну підпослідовність між реченням-кандидатом і набором еталонних речень, щоб виміряти їхню подібність на рівні речення. Найдовша спільна підпослідовність між двома реченнями вимагає лише збігів слів у послідовності, і зібрані слова не обов'язково є послідовними. Визначення найдовшої спільної підпослідовності досягається за допомогою динамічного програмування. Існують різні версії ROUGE для різних завдань: ROUGE_1 і ROUGE_W підходять для оцінки одного документа, тоді як ROUGE_2 і ROUGE_SU4 мають хорошу продуктивність для коротких резюме.

Оцінка ROUGE зазвичай виражається як оцінка F1, яка є середнім гармонійним значенням точності та запам'ятовування. Точність вимірює, скільки слів або фраз у згенерованому машиною резюме або ключових фразах відповідає тим, що містяться в довідкових резюме або ключових фразах, тоді як відкликання вимірює, скільки довідкових слів або фраз охоплено машинно згенерованим резюме або ключовими фразами.

Метрика ROUGE_N, яка вимірює збіг між n-грамами (суміжною послідовністю з n слів) у згенерованому машиною резюме або ключових

фразах і еталонних резюме або ключових фразах, є однією з найбільш широко використовуваних метрик ROUGE. Оцінка ROUGE_N F1 розраховується таким чином:

$$ROUGE\_N\ F1\ = \frac{2 * ROUGE\_N\ precision * ROUGE\_N\ recall)}{ROUGE\_N\ precision + ROUGE\_N\ recall}\ , \qquad (3.14)$$

де ROUGE_N precision – це кількість n-грамів, що перекриваються, між згенерованим машиною резюме або ключовими фразами та еталонними резюме або ключовими фразами, поділена на загальну кількість n-грамів у згенерованому машиною резюме або ключових фразах;

ROUGE_N recall – це кількість n-грамів, що перекриваються, між згенерованим машиною резюме або ключовими фразами та довідковими резюме або ключовими фразами, поділена на загальну кількість n-грамів у довідкових резюме або ключових фразах.

Метрика ROUGE-L, яка вимірює найдовшу загальну підпослідовність між згенерованим машиною резюме або ключовими фразами та еталонними підсумками або ключовими фразами, є ще одним часто використовуваним показником ROUGE. Оцінка ROUGE-L F1 розраховується таким чином:

$$ROUGE\_L\ F1\ = \frac{2 * ROUGE\_L\ precision * ROUGE\_L\ recall)}{ROUGE\_L\ precision + ROUGE\_L\ recall}, \qquad (3.15)$$

де ROUGE_L – це довжина найдовшої спільної підпослідовності між згенерованим машиною резюме або ключовими фразами та довідковими резюме;

ROUGE_L recall – це довжина найдовшої спільної підпослідовності між згенерованим машиною резюме або ключовими фразами та довідковими резюме або ключовими фразами, поділена на загальну кількість слів .

РОЗРОБКА МОДЕЛЕЙ ДЛЯ ПІДПИСУ ДО ЗОБРАЖЕННЯ

4.1 Набір даних

Набір даних Multi30k – це популярний еталонний набір даних для завдань машинного перекладу та створення підписів до зображень, що містить паралельні речення англійською, німецькою, французькою та чеською мовами в поєднанні з відповідними зображеннями. Набір даних Multi30k доступний для вільного завантаження з офіційного веб-сайту набору даних. Він поширюється за міжнародною ліцензією Creative Commons Attribution-ShareAlike 4.0. Набір даних містить 31789 пар зображення-речення, де кожне зображення супроводжується 5 підписами англійською, німецькою, французькою та чеською мовами.

Першим джерелом був набір Flickr30. Даний набор охоплював широкий спектр різного роду тем, а саме їжа, природа. люди, міські пейзажі, тварини, тощо. Для підпису зображень була використана англійська мова, а різні описи до зображення були різноманітні за довжиною, структурою та змістом, що робить його складним набором даних для завдань обробки природної мови.

Набір даних Multi30k було створено з метою надання більшого та більш різноманітного набору даних, ніж Flickr30k, який в основному був зосереджені на підписах до зображень. Набір даних використовувався для навчання та оцінки різних моделей для задач машинного перекладу та створення підписів до зображень і став стандартним набором даних у цій галузі.

Оскільки магістерська робота загострює увагу для україномовних мультимодальних даних, було прийняте рішення використовувати Multi30k який було перекладено на українську мову.

4.2 Згорткова нейронна мережа

Для даної роботи було використано CNN модель для рівня кодеру. Даний рівень забезпечує отримання вектора функцій для подальшої роботи з моделлю. Після обробки за допомогою кодеру набір векторів ознак можно записати наступним чином:

$$a = \{a_1, a_2, a_3, \ldots\ldots a_L\}, \quad (4.1)$$

де $a_i \in R^D$ – це масив векторів ознак;

D – кількість елементів, які використовуються для кодування кожного фрагмента (кількість одиниць повністю пов’язаного шару);

L – кількість фрагментів зображення.

Проаналізував існуючи марежі було брано три згорткові мережі VGG16, ResNet , InceptionV3. Дані мережі показали себе дуже добре у класифікації зображення. Тому для отримання вектору ознак було обрано самі ці згорткові мережі.

Дані мережі на вхід отримували зображення у тривимірному тензорі такий, як (299, 299, 3) для InceptionV3 та (224, 224, 3) для VGG16, ResNet.

Однак нас цікавить не отримання класифікації, а карта ознак. Її можно отримати на останьому шару згорткової мережі, що загалом складається із тривімірного тензора (fw, fh, fc), де fw і fh — ширина та висота карти функцій відповідно, а fc — кількість каналів. Для InceptionV3 це (8x8x2048), для VGG16 (1x1x4096) та ResNet (1x1x2048).

Зазвичай працюють із цією карткою у двовимірному тензорі стискаючи карту до форми (fw × fh, fc). Після передається до рівня кодеру.

4.3 Рекурентна нейронна модель

Recurrent Neural Networks (RNNs) – це тип нейронних мереж, які призначені для обробки послідовних даних, таких як дані часових рядів або тексту в природній мові. Ці мережі мають петлі в мережі, які дозволяють інформації зберігатися з часом та передаватись від одного кроку послідовності до наступного.

Архітектура RNN зображена на рисунку 4.1. Модель обробки послідовних даних збирає вхідні дані Xi та прихований стан hi-1 з попереднього кроку на кожному кроці часу. Після обробки вона генерує новий прихований стан та цільове значення.

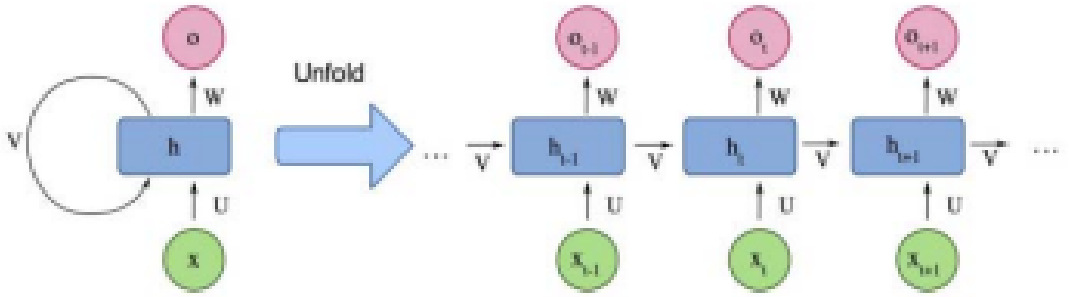

Рисунок 4.1 – Архітектура RNNs моделі

Оскільки у моделі RNN є великий недолік, який пов’язан із проблемою короткочасної пам’яті. Цей недолік пов’язують із проблемою зникнення градієнта. Тому для вирішення проблеми було розроблено дві архітектури LSTM та GRU.

Архітектура LSTM (Long Short-Term Memory) складається зі входу, вихідних даних та 3 воріт (рис. 4.2)- ворота забування, ворота оновлення та ворота виведення.

Ворота забування вирішують, яка інформація повинна забуватись, ворота оновлення додають нову інформацію до попереднього стану, а ворота виведення вирішують, яку інформацію повернути. LSTM може допомогти вирішити проблему зниклих градієнтів і дозволяє моделі зберігати та використовувати інформацію з довгих послідовностей.

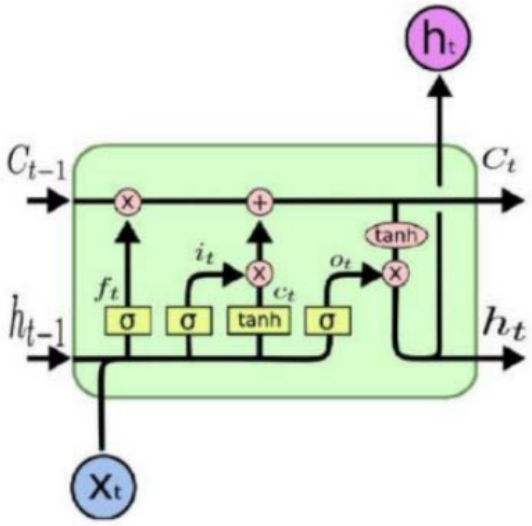

Рисунок 4.2 – Архітектура LSTMs моделі

Зв’язок між воротами , станом та входом визначається за формулами:

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}), \quad (4.2)$$

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}), \quad (4.3)$$

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}), \quad (4.4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W^{(c)} x_t + U^{(c)} h_{t-1} + b^{(c)}), \quad (4.5)$$

$$h_t = o_t tanh(c_t), \quad (4.6)$$

де $x_t \in R^d$ – вхідний вектор для блоку LSTM кроці часу t;

d – розмірність ознаки для кожного слова;

σ – сигмоїдна функції (для значень у межах [0, 1]);

$\odot$ – елементний добуток;

c_t – вектор клітини пам'яті, які призначені для вирішення проблеми вибуху градієнта або зникнення;

f_t – вектор активації воріт для забуття, які призначені для скидання комірки пам'яті;

i_t – вхідні ворота, котрі керують входом комірки пам'яті;

o_t – вихідні ворота котрі керують виходом комірки пам'яті.

Архітектура GRU (Gated Recurrent Unit) має меншу кількість параметрів, аніж LSTM і містить дві ворота (рис. 4.3) – ворота оновлення та ворота виведення. Ворота оновлення вирішують, яка інформація повинна бути збережена, а ворота виведення вирішують, яку інформацію повернути. GRU також допомагає уникнути проблеми зниклих градієнтів, але вона менш складна за LSTM, що робить її більш ефективною при обробці невеликих послідовностей.

Зв’язок між воротами , станом та входом визначається за формулами:

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z), \quad (4.7)$$

$$i_t = \sigma_g(W_r\, x_t + U_r h_{t-1} + b_r), \tag{4.8}$$

$$\hat{h}_t = \phi_g(W_h\, x_t + U_h(r_t \odot h_{t-1}) + b_h), \tag{4.9}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t), \tag{4.10}$$

де x_t – вхідний вектор для блоку GRU;

h_t – вхідний вектор;

$\hat{h}_t$ – вектор, який забезпечує активацію кандидата;

z_t – оновлення вектору воріт;

r_t – скинути вектор воріт;

W, U, b – матриці параметрів і вектор.

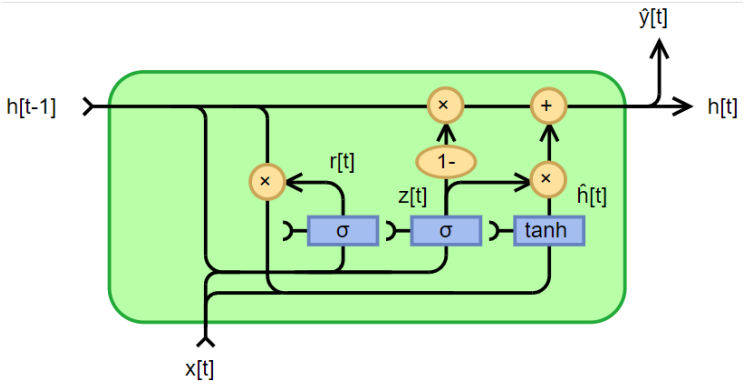

Рисунок 4.3 – Архітектура GRUs моделі

Моделі LSTM та GRU дуже добре справляється із своїми задачами. Тому для того щоб виявити наскільки ефективно вони можуть навчати модель при невеликому наборі даних, та зробити порівняльний аналіз для подальшої роботи було обрано саме ці моделі.

4.4 Кодер-декодер

Створення субтитрів складна задача. Для цього було розроблено багато методів та реалізацій. Однією із цих рішень стала архітектура кодеру декодеру. Даний підхід поєднав у себе комп'ютерний зір та обробку природної мови. Також архітектура поєднує у собі передавальне навчання, яке створює вектор функцій для кодеру моделі.

Стандартна архітектура має два елементи кодер та декодер (рис. 4.4), які представляють собою архітектури нейронних мереж. Кодер приймає на вхід зображення та перетворює вектор фіксованої довжини. Для цієї задачі підходить CNNs або передавальне навчання. Для даної роботи було використано передавальне навчання, яке було реалізовано за допомогою CNNs рішень таких, як VGG16, ResNet , InceptionV3.

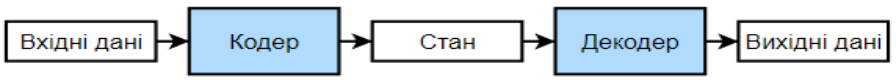

Рисунок 4.4 – Архітектура кодеру-декодеру

Стандартна архітектура була розроблена для перекладів тексту. Потім розробники взяли за основу архітектуру та створили модель для пидпису зображення. Я декілька прикладів реалізації архітектури кодеру декодеру – це Inject та Merge. За даними досліджень Merge модель отримувала більш

якісні результати, через що було обрано цю модель для ефективного порівняння роботи LSTM та GRU.

Merge модель представляє собою не складну модель які представлено на рисунку 4.5. Вона складається із вектора функції розміром 4098 або 2044. який передається до зв’язного шару із функцією активації ReLu, який на виході створює вектор ознак 256.

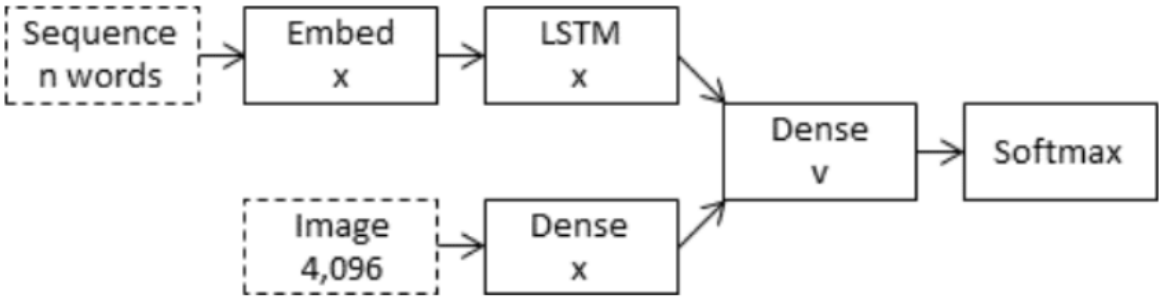

Рисунок 4.5 – Демонстрація Merge моделі

Обробка тексту тех перед початком необхідно зробити. Очистити текст від зайвих знаків та перевести текст у нижній регістр. Додати лексеми «start» «end». Результат зображений на рисунку 4.6.

Далі створити словник слово та його індекс. Оскільки машина краще розуміє мову чисел. Далі вектор, що містить послідовність цілих чисел, що відповідає текстовому опису зображення передається до Embedding шару. Вектор проходить через Embedding який перетворює кожне ціле число в вектор з плаваючою крапкою фіксованої довжини, що можна навчити під час навчання моделі. Далі все передається до LSTM, який обробляє послідовність векторів, що містять текстовий опис зображення. LSTM має можливість запам'ятовувати попередній контекст та використовувати його для генерації наступного слова. На виході ми отримуємо на виході вектор, такий як і у зображення 564. Декодер приймає на вхід ці дані та об'єднує

два вектори за допомогою операції додавання. Далі передаються до зв'язного шару із функцією активації ReLu, який на виході створює вектор ознак 256. Наступний рівень використовує функцію активації softmax для передбачення найбільш ймовірного наступного слова в словнику.

```
['<start>  двоє молодих білих чоловіків біля багатьох кущів <end>',
 '<start>  кілька чоловіків у касках керують системою гігантських блоків <end>',
 '<start>  маленька дівчинка піднімається в деревяний будиночок <end>',
 '<start>  чоловік у синій сорочці стоїть на драбині й миє вікно <end>',
 '<start>  двоє чоловіків біля плити готують їжу <end>',
 '<start>  чоловік у зеленому тримає гітару а інший дивиться на його сорочку <end>',
 '<start>  чоловік посміхається опудалу лева <end>',
 '<start>  модна дівчина розмовляє на мобільний телефон повільно ковзаючи вулицею <end>',
 '<start>  жінка з великою сумкою йде біля воріт <end>',
 '<start>  хлопчики танцюють на стовпах посеред ночі <end>']
```

Рисунок 4.6 – Демонстрація обробленого тексту

Для збільшення точності генерації підписів було реалізовано Word2Vec Embeddings для україномовного тексту. Його використовуються для перетворення текстового опису зображення в векторну форму, що може бути використана в якості вхідних даних для нейронної мережі. Собою він представляє метод векторизації слів, який використовує контекстні залежності між словами у корпусі текстів для створення векторів з фіксованою довжиною для кожного слова.

Для того щоб оцінити різницю між сгенерованою послідовністю слів та правильною послідовністю слів, що описують зображення. було використано категоріальна крос ентропія та оптимізатор Adam, який показав гарну ефективність у моїй бакалаврській роботі.

4.5 Модель кодер-декодер з механізмом уваги.

Механізм уваги теж можно використовувати для задач для підпису зображення. За основу частіше беруть архітектуру кодеру декодеру, та

трішки модифікують. Основна ідея цього методу це приблизити розуміння моделі до того, як робить людина. Людина спроможна фільтрувати інформацію підсвідома, залишаючи тільки важливі деталі. За таким принципом також працює і дана модель. Механізм уваги дозволяє нейронній мережі мати можливість зосереджуватися на своїй підмножині вхідних даних для вибору конкретних функцій.

За останніми дослідженнями механізм уваги довів, що він може покращити якість субтитрів до зображення. Під увагою мають на увазі зважену сумму кодера. Архітектура дуже проста. Приклад архітектури зображено на рисунку 4.7.

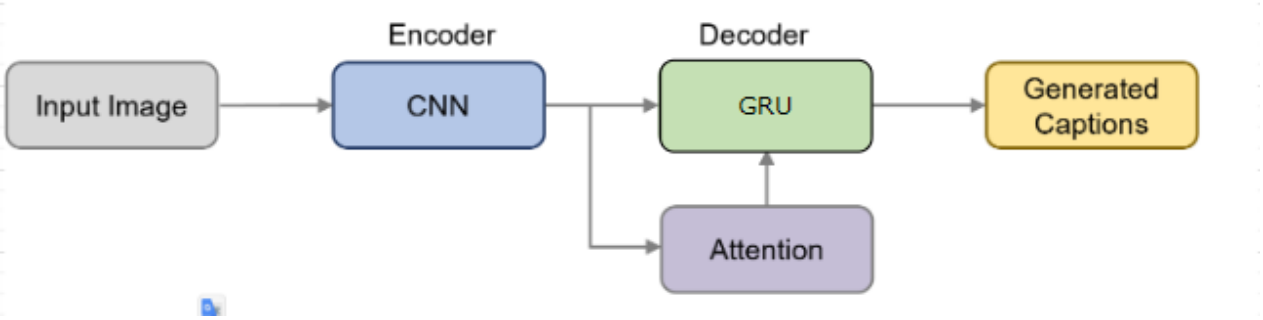

Рисунок 4.7 – Архітектура кодеру-декодеру з механізмом уваги

На початку ми обробляємо зображення за допомогою моделі CNN. На виході ми отримаємо карти функцій. Їх передаємо до модулю уваги із прихованим станом і призначає вагу кожному пікселю зображення. Далі пропускаем через декодер, де генерується вихідна послідовність слів. В кінці отримуємо опис зображення.

Існує багато реалізацій механізмів уваги. Для даної роботи було обрано модель описану у роботі Show, Attend and Tell, by Xu et al. 2015 року[], який у свою чергу був надихнутий м’яким механізмом уваги Богдана. Його називають м’яким через те, що він дивиться на зображення в цілому. Це сильно відрізняється від інших. Оскільки більшість механізмів

дивляться локально, або на певні місця зображення в кожен момент часу вхідного простору зображення. Сам же м’який механізм, уваги залежно від того які є показники релевантності, з урахуванням того, що передбачає декодер, намагається отримати певні характеристики зображення.

Даний вид уваги являється видом самоуваги, оскільки оскільки декодер звертається до кодера, щоб створити його вихід, але він також керується своїм власним внутрішнім станом.

Рівень кодеру не нічим не відрізняється як і в моделі кодер декодер реалізованою раніше.

Цікавіше стає коли декодер отримує вихідні дані кодера та генерує послідовність слів для опису вхідного зображення. На кожному кроці часу t декодер отримує попередній прихований стан h_{t-1} і генерує новий прихований стан h_t і слово y_t.

У той самий час механізм уваги обчислює контекстний вектор c_t, який є зваженою сумою областей карти функцій. для обчислення ваги обчислюють показник подібності між попереднім прихованим станом h_{t-1} і кожною областю карти ознак і за допомогою параметризованої функції f:

$$e_{t,i} = f(h_{t-1}, a_i), \tag{4.11}$$

де $e_{t,i}$ – це оцінка подібності між h_{t-1} і областю карти характеристик і.

Далі необхідно нормалізувати показники подібності за допомогою функції softmax, щоб отримати вагові коефіцієнти уваги

$$a_{t,i} = \frac{exp(e_{t,i})}{\sum_{k=1}^{L} exp(e_{t,k})}, \tag{4.12}$$

де $a_{t,i}$ – вага уваги для області та карти ознак на кроці часу t.

Далі необхідно обчислити контекстний вектор c_t, як зважену суму областей карти функцій:

$$a_{t,i} = \sum_{k=1}^{L} a_{t,k}, a_{k'} \quad (4.13)$$

де $a_{t,i}$ – вага уваги для області та карти ознак на кроці часу t.

Вектор контексту c_t об'єднується з поточним прихованим станом h_t декодера та проходить через лінійний рівень з наступною функцією активації tanh для отримання нового прихованого стану h'_t:

$$h'_t = tanh\,(W_c[\,c_t;\, h_t\,] + b_c), \quad (4.14)$$

де W_c – вивчена вагова матриця;

b_c – вивчений термін зміщення;

$[\,c_t;\, h_t]$ – конкатенація вектора контексту та поточного стану декодера;

$tanh$ – функція активації гіперболічного тангенса.

Новий прихований стан h'_t передається через лінійний рівень, а потім функція активації softmax для генерації розподілу ймовірностей у словнику:

$$P(y_t \mid y_1, \ldots, y_{t-1}, x) = softmax(W_s * h'_t + b_i), \quad (4.15)$$

де W_c – вивчена вагова матриця;

b_c – вивчений термін зміщення;

y_t – прогнозований розподіл ймовірностей у словнику для наступного слова в підписі.

4.6 Трансформер модель

Трансформер модель це нейронна мережа, яка вивчає контекст та значення, враховуючи зв'язки між послідовними даними, такими як слова в реченні. У даній роботі реалізован трансформер із Multi-head Attention.

Стандартна архітектура трансформера складається із шарів кодеру та декодеру. Кожен із шарів має відповідні рівні вбудування для відповідних вхідних даних. Останнім шаром є вихідний рівень, який створює остаточне передбачення тексту. Стандартна архітектура зображена на рисунку 4.8.

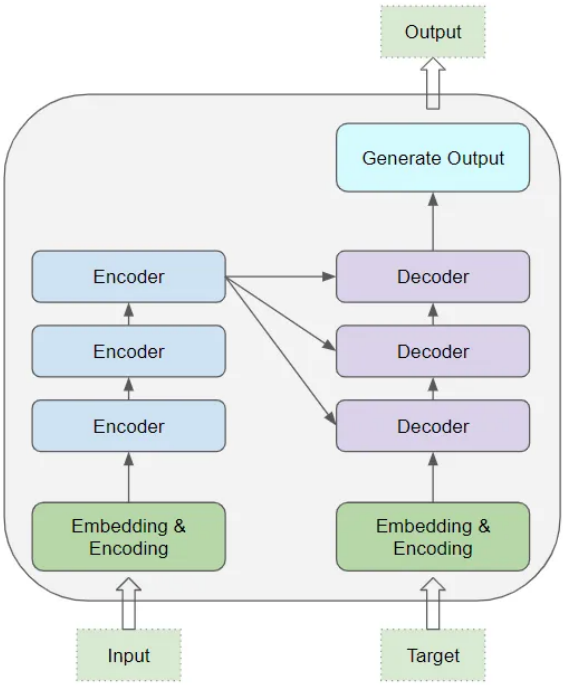

Рисунок 4.8 – Базова архітектура моделі трансформер

Архітектура кодер шару має рівень Self-attention, який у нашому випадку є Multi-head Attention рівнем. Цей рівень обчислює зв’язок між різними словами в послідовності. та рівень Feed-forward. Окрім двох вищезазначених шарів, він також має залишкові з’єднання пропуску навколо обох шарів разом із двома шарами LayerNorm. Декодер має рівень Self-attention, який у нашому випадку є Masked Multi-head Attention рівнем, энкодер декодер уваги, який у нашому випадку є Multi-head Attention рівнем та рівень Feed-forward. Як і в випадку з кодувальником, декодер має поміж рівнів є шар LayerNorm. Демонстрація моделі на рисунку 4.9

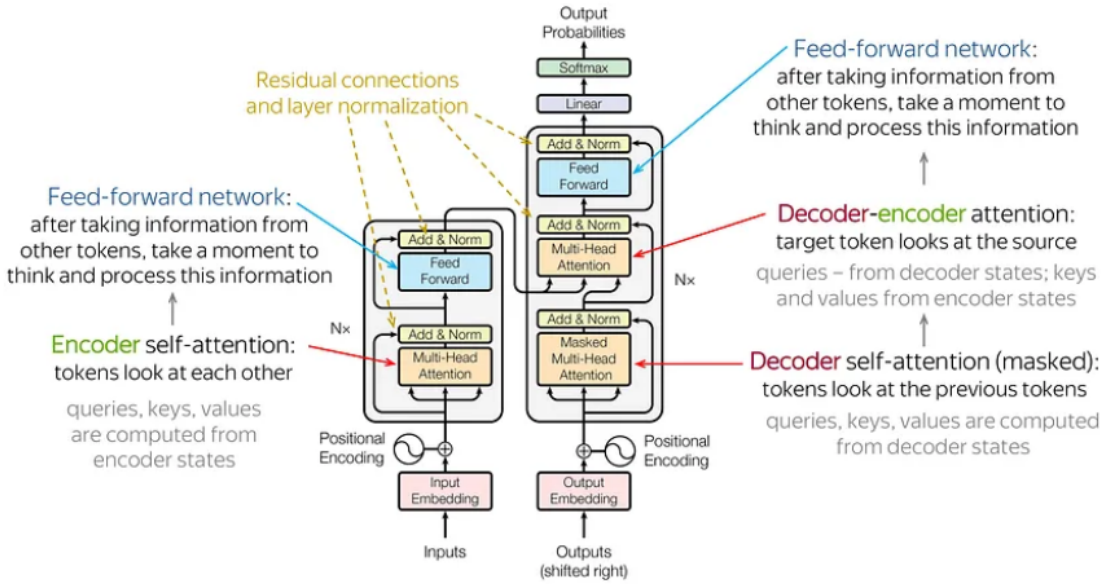

Рисунок 4.9 – Демонстрація моделі трансформеру з механізмом уваги

Механізм уваги для підписів до зображення у кодері ваги уваги призначаються кожному фрагменту зображення. На кожну частину, на яку було звернено увагу відповідають ваги, під час виконання створення підпису до зображення.

Архітектура механізму уваги складається із запиту Q , ключу K та значення V. Кожен ключовий вектор пов’язаний із вектором значень і

запитується вектором запиту. Кожен патч у зображенні представлений ключем K, і кожен ключ K відповідає певному значенню V. Під час обчислення уваги кількох головок у декодері, запити до патчів здійснюються шляхом використання токенів Q.

$$Attention(Q, K, V) = (\frac{QK^T}{\sqrt{d_k}})V, \quad (4.16)$$

де Q, K, V – являються тензорами.

Кожен тензор утворюється за допомогою множення матриці між навчальним на ваговим коефіцієнтом W_q , W_k, W_v

Multi-head Attention це модуль уваги, який повторює свої обчислення кілька разів паралельно Кожне з цих обчислень називається Attention Head. Кожне з цих обчислень пізніше об’єднується для отримання остаточного оцінки уваги.

Запит, ключ і значення проходять через індивідуальні лінійні шари з власними вагами, створюючи три окремих результати – Q, K і V. Після цього ці результати комбінуються за допомогою формули уваги, яка використовується для обчислення показника уваги. Математична архітектура представлена на рисунку 4.10

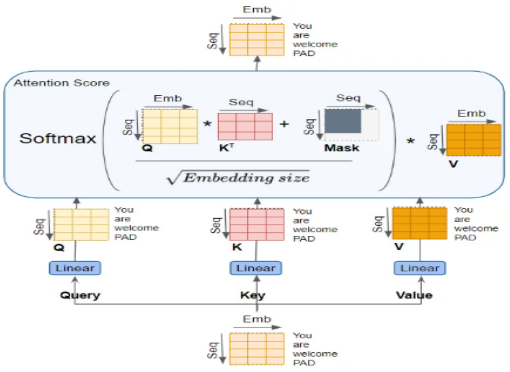

Рисунок 4.10 – Демонстрація роботи Multi-head Attention

Архітектура Multi-head Attention передбачає наявність методу маски. Для рівнів Self-attention та кодер-декодер.маска дозволяє обнулити увагу на виходи, що відповідають відступом у вхідних реченнях. Це забезпечує, що відступи не мають впливу на самоувагу (рис. 4.11).

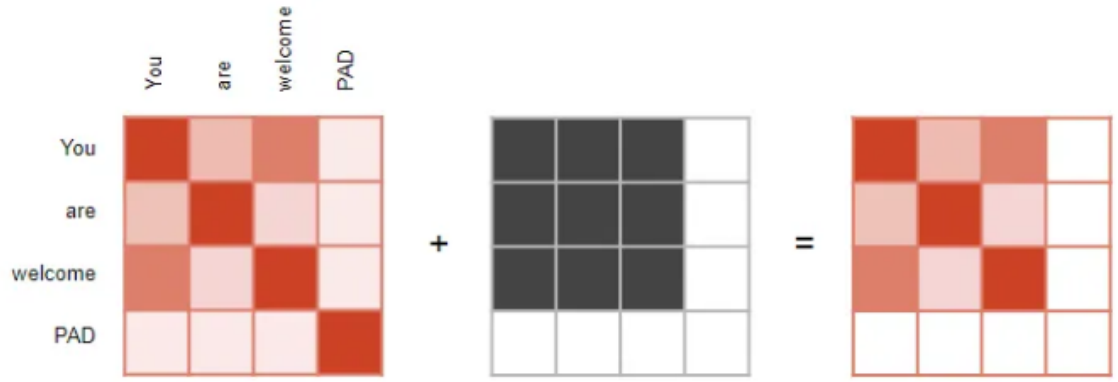

Рисунок 4.11 – Демонстрація роботи маски

Маска у шарі декодеру допомагає моделі не шахрувати, іншими словами підглядати у наступне слово під час прогнозування.Під час обчислення показника уваги, маскування застосовується до чисельника безпосередньо перед застосуванням функції Softmax. Замасковані елементи встановлені на величину від'ємно нескінченності, що дозволяє функції Softmax перетворити ці значення в нуль.

АНАЛІЗ РЕЗУЛЬТАТІВ

5.1 Аналіз результатів

Багато було створено наборів даних? які підходять для вирішення задачі підпису зображень. У даній роботі було обрано набір даних Multi30k який було переведено на українську мову. Набір даних має пару зображення та один опис. Загальна кількість зображень 30000, підписів 30000. Максимальна довжина речення у наборі даних дорівнює 36 слів. Словниковий запас отриманий після токенізації дорівнює 20000 слів.

Для роботи з даними даний набір даних був поділений як 60% на тренувальний, 20 % валідаційний та 20 тестувальний. Тому кінечні дані для тренування містили 18000 даних, далідаційних та тестувальні 6000 прикладів.

Всі експерименти були сплановані та мали наступні плани експерименту.

Модель 1 – Кодер-декодер:

– Тип моделі комп'ютерного зору: «VGG16», «ResNet50», «InceptionV2»;

– Тип рекурентної мережі: «LSTM» «GRU» ;

– Кількість об’єктів зображення 256;

– Кількість текстових одиниць: 256;

– Розмір партії: 32, 64;

– Оптимізатор: Адам.

Модель 2 – Кодер-декодер з механізмом уваги:

– Тип моделі комп'ютерного зору: «VGG16», «ResNet50», «InceptionV2»;

– Тип рекурентної мережі: «LSTM» «GRU» ;

– Кількість об’єктів зображення 256;

– Кількість текстових одиниць: 256;

– Розмір партії: 32, 64;

– Оптимізатор: Адам.

Модель 3 – Трансформер модель з механізмом уваги Multi-head Attention:

– Тип моделі комп'ютерного зору: «VGG16», «ResNet50», «InceptionV2»;

– Кількість об’єктів зображення 256;

– Кількість текстових одиниць: 256;

– Розмір партії: 32, 64;

– Оптимізатор: Адам.

Тренування для кожної моделі виконувались 30 епох. Також дослідження про проводились для 50, 100, 150 епох.

Після тривалого часу тренувань було виявлено, що модель стає схильною до перенавчаня. тому оптимальним варіантом для тренування було обрано 30 епох.

Головним обмеженням при тренування моделі було системні та тривалий час тренування. Оскільки деякі дані були передчасно підготовлені. Та архітектура концепту моделі була розроблена, таким чином, що оброблені дані вже подавались на вхід моделі, це скорочувати час. Але все ще займало багато часу. Тому не було достатньо часу випробувати усі варіанти. Рішення стосовно який гіперпараметр використовувати було прийнято на основі попередніх результатів моделей.

Для того щоб оцінити наскільки ефективність моделі залежить від гіперпараметрів було проведено експерименти із різними комбінаціями параметрів.

Отримані результати з першої моделі представлені у таблиці 5.1.

Таблиця 5.1 – Результати експерименту з гіперпараметрами моделі кодер-декодер

CNN	RNN	Emb	Feat	Batch	Optim	Precision	Recall	F1 score
VGG16	LSTM	256	256	32	Адам	0.6201	0.6859	0.6512
VGG16	LSTM	256	256	64	Адам	0.6925	0.6934	0.6928
VGG16	GRU	256	256	32	Адам	0.7284	0.7644	0.7459
VGG16	GRU	256	256	64	Адам	0.7621	0.7954	0.7784
Inception V2	LSTM	256	256	32	Адам	0.7436	0.7872	0.7647
Inception V2	LSTM	256	256	64	Адам	0.6957	0.7354	0.7149
Inception V2	GRU	256	256	32	Адам	0.6340	0.6789	0.6557
Inception V2	GRU	256	256	64	Адам	0.7121	0.7263	0.7191
ResNet50	LSTM	256	256	32	Адам	0.7688	0.7782	0.7734
ResNet50	LSTM	256	256	64	Адам	0.6834	0.7686	0.7235
ResNet50	GRU	256	256	32	Адам	0.7394	0.8645	0.7971
ResNet50	GRU	256	256	64	Адам	0.7324	0.8827	0.8006

На рисунку 5.1 можно побачити, результати оцінки зображення методом оцінки берта, які були середнім числом отриманих усіх результатів з тестового набору даних. Перша оцінка є результатом точності опису зображення, друга оцінкою того скільки фактичних позитивних результатів охоплює наша модель. Та третя оцінка є оцінкою точності тексту.

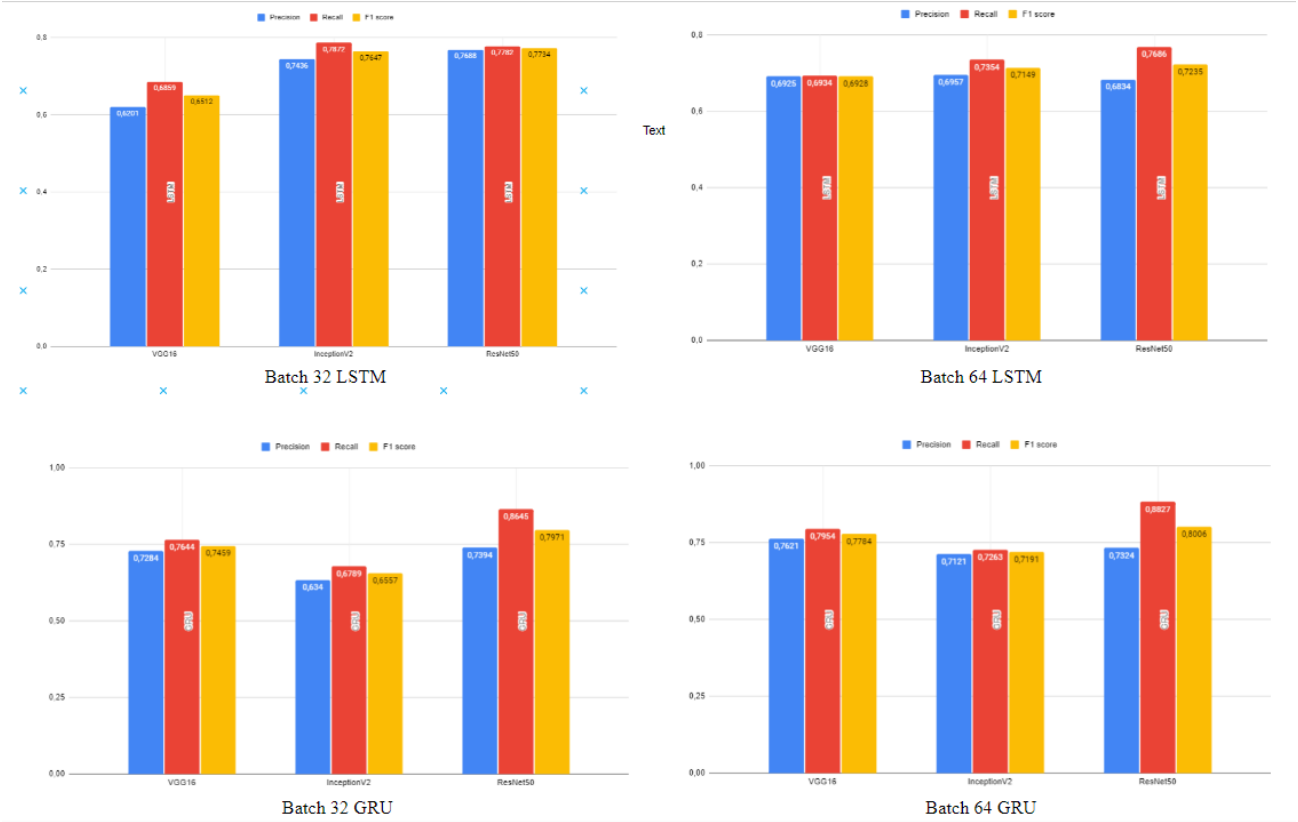

Рисунок 5.1 – Результат порівняння оцінок моделі та його партій

Дані результати були дуже гарними, оскільки більшість створених описів до зображення були більше 0.7 процентів. Що є дуже близьким до еталонного опису з яким порівнювали згенерований текст.

За результатами можно побачити що GRU показувала кращі результати аніж LSTM. Це можно побачити на рисунках 5.2 та 5.3. Та результати, які були згенеровані моделлю, коли партія дорівнював 64 були вище ніж при 32. Це пов’язано з тим що партія 64 забезпечує вищу точність при тренуванні. Але є великий недолік, він потребує меньше часу на тренування, та більше пам’яті, а це було головним обмежувачем для тренування. Оскільки оцінка точності опису зображення поміж партіями 64, 32 є не дуже великою то можно пожертвувати надання моделі точності та використовувати партію 32, щоб було більше пам'яті для розрахунків.

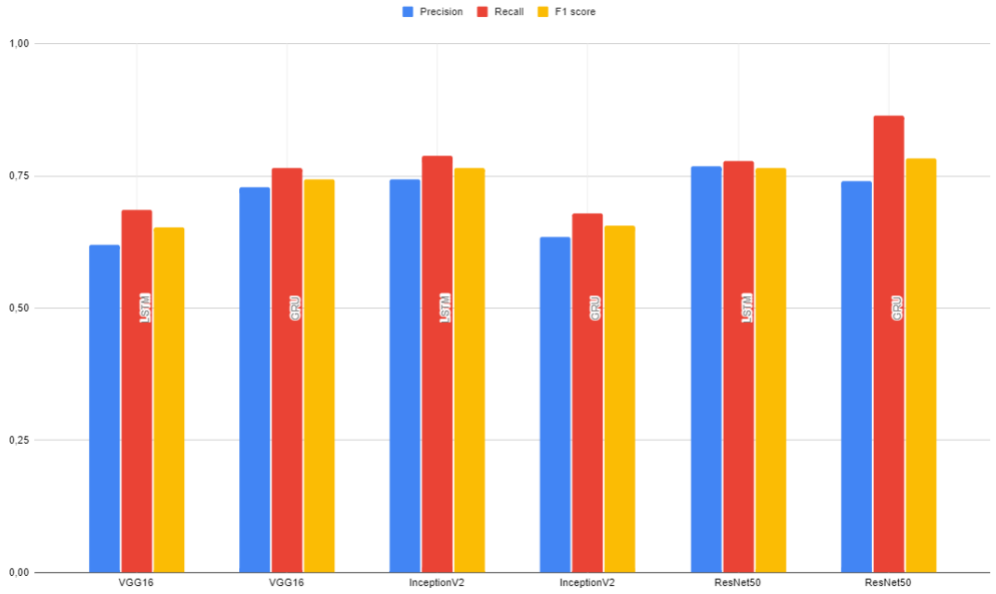

Рисунок 5.2 – Демонстрація результату при партії 32

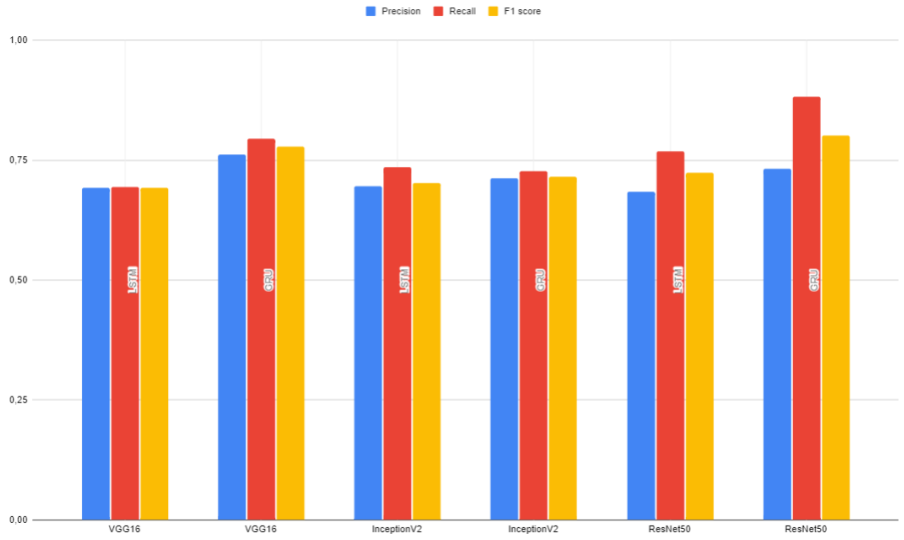

Рисунок 5.3 – Демонстрація результату при партії 64

На основі отриманих результатів було прийнято рішення що у подальших моделях будемо використовувати патрія 32 та GRU. Також не було зроблено експериментів із збільшенням гіперпараметрів, оскільки тренування забирало багато часу. Не тому не було можливості провести

експеримент. Але можно припустити що збільшення гіперпараметрів ускладньє модель тим самим може створити ситуацію з перенавчанням. Але це є тільки припущенням тому у наступній роботі буде необхідно провести ряд експериментів для підтвердження теорії.

Проблема перенавчання є розповсюдженою проблемою. Виявити її модно за допомогою порівняння продуктивності тренувально набору даних та валідаційного. Це можно виявити, оскільки в обох випадках використовуеться одна і та сама функція втрат.

Завданні генерації підписів до зображення є дуже складною, тому є велика кількість технік щоб уникнути цього. Наприклад метод виключення та нормалізатор партії. У даній роботі було використано метод виключення.

Тепер розглянемо результати другої моделі у таблиці 5.2.

Таблиця 5.2 – Результати експерименту з гіперпараметрами моделі кодер-декодер з механізмом уваги

CNN	RNN	Emb	Feat	Batch	Optim	Precision	Recall	F1 score
VGG16	GRU	256	256	32	Адам	0.7598	0.7854	0.7723
Inception V2	GRU	256	256	32	Адам	0.774	0.7974	0.7855
ResNet50	GRU	256	256	32	Адам	0.8025	0.8437	0.8225

За результати оцінки моделі, можно зробити висновок, що моделі із механізмом уваги покращила результати, якщо порівнювати з першою моделю. Ми можемо бачити, що за результатами оцінки точність опису зображення було набагато вище ніж у першої моделі. Вони були від 0.7 до 0.85. На рисунку 5.4 можно побачити результати.

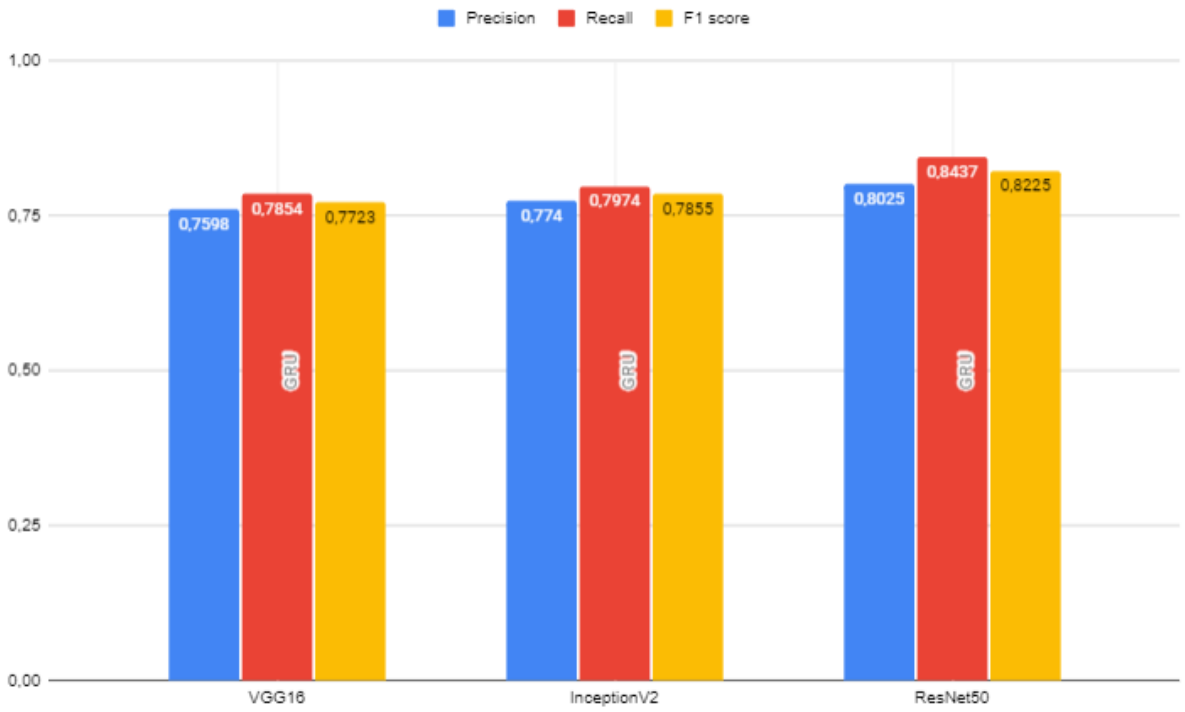

Рисунок 5.4 – Демонстрація результатів моделі з механізмом уваги

Тепер розглянемо результати третьої моделі у таблиці 5.3.

Таблиця 5.3 – Результати експерименту з гіперпараметрами моделі трансформер

CNN	RNN	Emb	Feat	Batch	Optim	Precision	Recall	F1 score
VGG16	GRU	256	256	32	Адам	0.7372	0.7634	0.7501
Inception V2	GRU	256	256	32	Адам	0.7554	0.7712	0.7632
ResNet50	GRU	256	256	32	Адам	0.7934	0.8237	0.8082

За результати оцінки моделі, можно зробити висновок, що модель трансформер не гірше ніж модель уваги. Оскільки обидві моделі використовують механізм уваги. Ми можемо бачити, що за результатами

оцінки точність опису зображення були від 0.7 до 0.85. На рисунку 5.5 можно побачити результати.

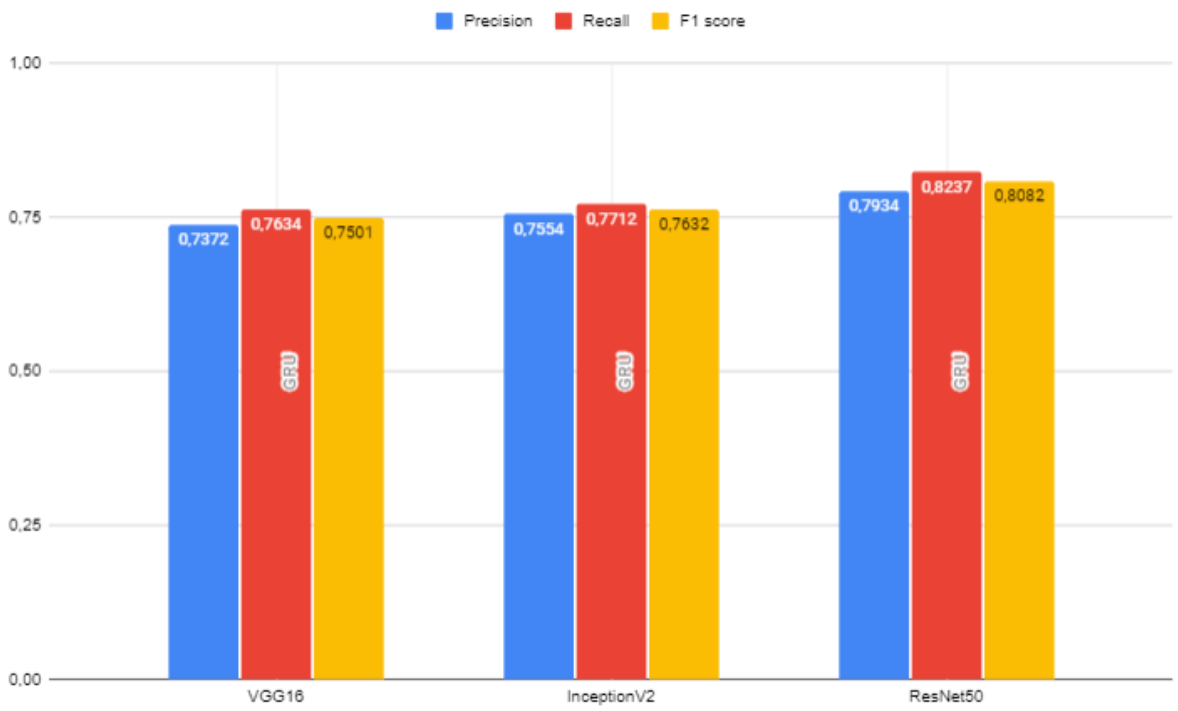

Рисунок 5.5 – Демонстрація результатів моделі трансформеру з механізмом уваги

Тепер перейдмо до оцінювання якості підписів до зображення. За результатами у кожної моделі можно було спостерігати наступні результати. У кожної системи можно було спостерігати як вини були спроможні у виявлення основних елементів

У моделі кодеру-декодеру можно було спостерігати не різноманітність підпису у деталях, оскільки більшість створених підписів були зосереджені на объектах та взаємодії з оточеними предметами. Приклад можно побачити на рисунку 5.6.

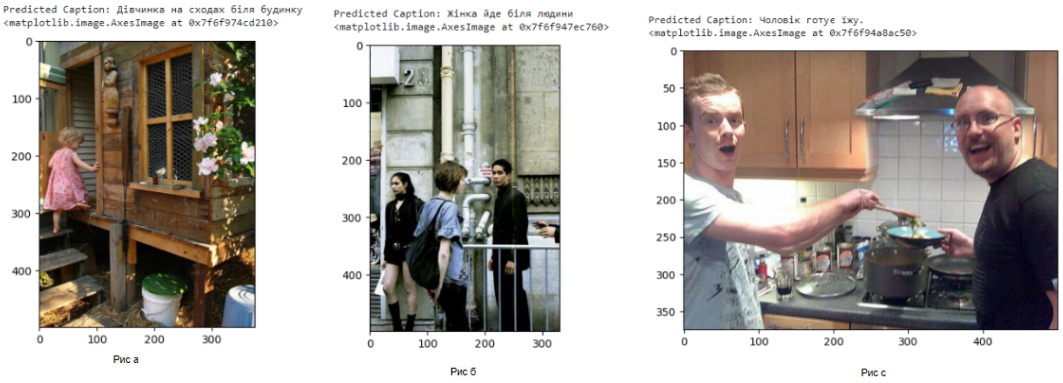

Рисунок 5.6 – Приклад результатів кодер-декодер

У моделі кодеру-декодеру з увагою, можно було спостерігати що моделі були більш лояними до опису об'єкта, але зустрічаються помилки у деяких деталях, або помилкова кількість виявлених людей.

Рисунок 5.7 – Приклад результатів кодер-декодер с механізмом уваги

У моделі трансформеру можно було спостерігати теж саме у прикладах інколи було присутнє опис одягу або взаємодію із предметами.

Самими розповсюдженою проблемою було те що модел пропускала деталі, якщо їх було дуже багато. Та не описувала іх. Це не є критично, бо модель навчалась на прикладах, які теж могли не мати великого різноманітного опису для зображення.

Рисунок 5.8 – Приклад результатів трансформер модель с механізмом уваги

ВИСНОВКИ

В ході виконання даної дипломної роботи було розроблено концептуальну модель для створення підписів до зображення. На основі проаналізованої моделі було розроблено та реалізовано моделі для вирішення даної задачі.

Спираючись на постановку задачі було обрано підходящий набір мультимодальних даних, а саме Multi30k, який було переведено на українську мову. Даний набір було перекладено на українську та містить 31789 пар зображення-речення.

Були проаналізовані та досліджені існуючі архітектури та методи для обробки мультимодальних даних. Оскільки розроблена модель містить дві задачі це задачі комп'ютерного зору та обробки природної мови було досліджено методи та підходи для вирішення поставлених задач.

Було досліджено методи комп'ютерного зору, а саме Xception, VGG16, VGG19, ResNet50, InceptionV2, InceptResNetV2, MobileNet.

Було досліджено та проаналізовано методи природної обробки мови, такі як рекурентні нейронні мережі LSTM та GRU, мультимодальне навчання, методи передавального навчання, методи генеративних змагальних мереж, метод кодер-декодер, метод кодер-декодер з механізмом ураги, трансформер модель, мета-навчання. Серед яких було обрано метод кодер-декодер, метод кодер-декодер з механізмом ураги та трансформер модель, які було досліджено та розроблено.

Було розроблена модель кодер-декодер на основі архітектури Merge модель. було реалізовано модель кодер-декодер з механізмом самоуваги, та трансформер модель з механізмом уваги Multi-head Attention.

Перед початком роботи було розроблено план експерименту та проведен експеримент з розробленими моделями.

Розроблені моделі було натренерован на 30 епохах, та на основі досліджених метрик оцінки моделі було проведено оцінку ефективності моделі.

На базі отриманих результатів був зроблений висновок, що у архітектурі кодер-декодер кращі результати давала рекурентна нейронна мережа GRU. Інші дві моделі у яких були реалізовані різні архітектури механізму самоуваги давали кращі результати ніж кодер-декодер.

Оскільки даний напрямок тільки розвивається є багато методів для дослідження, які можно реалізувати, та дослідити. Тому у майбутньому можно провести додаткові дослідження, та зробити удосконалення для розроблених моделей які були реалізовані у даній роботі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Vinyals O., Toshev A., Bengio S., Erhan D. Show and Tell: A Neural Image Caption Generator. *Computer Vision and Pattern Recognition*. 2015. P. 3156–3164.

2. Vinyals O., Toshev A., Bengio S., Erhan D. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *International Conference on Machine Learning*. 2015. P. 3156–3164.

3. Vinyals O., Toshev A., Bengio S., Erhan D. Neural Image Caption Generation with Visual-Semantic Embeddings. *Computer Vision and Pattern Recognition.* 2015.

4. Karpathy A., Fei-Fei. L. Deep Visual-Semantic Alignments for Generating Image Descriptions. *Computer Vision and Pattern Recognition.* 2015.

5. Anderson P., He X., Buehler C., Teney D., Johnson M., Gould S., Zhang L. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. *Computer Vision and Pattern Recognition.* 2015.

6. You Q., Jin H., Wang Z., Fang C., Luo J. Image Captioning with Semantic Attention. *Association for Computational Linguistics*. 2015.

7. Kiros R., Salakhutdinov R., Zemel R. S. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. *In Advances in Neural Information Processing Systems Deep Learning Workshop,* 2014.

8. Mnih A., Hinton, G. Three new graphical models for statistical language modelling. *In Proceedings of the 24th international conference on Machine learning*. 2007. P. 641–648.

9. Mao J., Xu W., Yang Y., Wang J., Yuille A. L. Explain Images with Multimodal Recurrent Neural Networks. *In NIPS 2014 Deep Learning Workshop*. 2014.

10. Mao J., Yuille A. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). *International Conference on Learning Representations*. 2015.

11. Karpathy A., Fei-Fei L. Deep visual-semantic alignments for generating image descriptions. *In 2015 IEEE Conference on Computer Vision and Pattern Recognition* . 2015. Vol. 39. P. 3128–3137.

12. Chen X., Zitnick C. L. A recurrent visual representation for image caption generation. *In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. P. 2422–2431.

13. Kalchbrenner N., Blunsom, P. Recurrent Continuous Translation Models. *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 2013. P. 1700–1709.

14. Kiros R., Zemel R., Salakhutdinov, R. Multimodal Neural Language Models. *In Proceedings of the 31st International Conference on Machine Learning*. 2014. Vol. 32. P. 595–603.

15. Vinyals O., Toshev A., Bengio S., Erhan, D. Show and tell: A neural image caption generator. *In 2015 IEEE Conference on Computer Vision and Pattern Recognition.* 2015. June 07-12. P. 3156–3164.

16. Wu Q., Shen C., Liu L., Dick A. What Value Do Explicit High Level Concepts Have in Vision to Language Problems?. *In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. P. 203–212.

17. Shetty R., Rohrbach M., Hendricks L. A., Fritz, M., Schiele, B. Speaking the Same Language: Matching Machine to Human Captions by Adversarial Training. *In 2017 IEEE International Conference on Computer Vision*. 2017. P. 4155–4164.

18. Li D., Huang Q., He X., Zhang, L. Diverse and Accurate Visual Captions by Comparative Adversarial Learning. *In 32nd Conference on Neural Information Processing Systems*. 2018.

19. Bernardi R., Çakici R., Elliott D., Erdem A., Erdem E., Ikizler-Cinbis N., Keller F., Muscat A., Plank B. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *IJCAI International Joint Conference on Artificial Intelligence.* 2017. P. 4970–4974.

20. Bai S., An, S. A survey on automatic image caption generation. Neurocomputing. 2018. P 291–304.

21. Hossain M. Z., Sohel F., Shiratuddin M. F., Laga H., Hossain Z., Sohel F., Shiratuddin M. F., Laga, H. A Comprehensive Survey of Deep Learning for Image Captioning. ACM Computing Surveys. 2019. P. 1–36.

22. Rensink R. A. The Dynamic Representation of Scenes. Visual Cognition. 2000. P. 17–42.

23. Spratling M. W., Johnson, M. H. A Feedback Model of Visual Attention. Journal of Cognitive Neuroscience. 2004. P. 219–237.

24. Mnih V., Heess N., Graves A., Kavukcuoglu K., Deepmind, G. Recurrent Models of Visual Attention. *In Advances in Neural Information Processing Systems*. 2014.

25. Donahue J., Hendricks L. A., Guadarrama, S., Rohrbach M., Venugopalan S., Darrell T., Saenko, K. Long-term recurrent convolutional networks for visual recognition and description. *In 2015 IEEE Conference on Computer Vision and Pattern Recognition*. 2015. P. 2625–2634.

26. Chen X., Zitnick, C. L. Mind's eye: A recurrent visual representation for image caption generation. *In 2015 IEEE Conference on Computer Vision and Pattern Recognition.* 2015. P. 2422–2431.

27. Chen X., Zitnick, C. L. Mind's eye: A recurrent visual representation for image caption generation. *In 2015 IEEE Conference on Computer Vision and Pattern Recognition.* 2015. P. 2422–2431.

28. Mao J., Yuille A. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). *International Conference on Learning Representations.* 2015. P. 1–17.

29. Farhadi A., Hejrati M., Sadeghi M. A., Young P., Rashtchian C., Hockenmaier J., Forsyth D. Every Picture Tells a Story: Generating Sentences from Images. *In Proceedings of the 11th European Conference on Computer Vision.* 2010. Vol 6314. P. 15–29.

30. Kulkarni G., Premraj V., Dhar S., Li S., Choi Y., Berg A. C., Berg T. L. Baby talk: Understanding and generating simple image descriptions. *In 24th IEEE Conference on Computer Vision and Pattern Recognition.* 2011. Vol. 18. P. 1601–1608.

31. Li S., Kulkarni G., Berg T., Berg A., Choi, Y. Composing simple image descriptions using web-scale n-grams. *In Proceedings of the 15th Conference on Computational Natural Language Learning.* 2011. P 220–228.

32. Kuznetsova P., Ordonez V., Berg A. Collective generation of natural image descriptions. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics.* 2012. July 1. P. 359–368.

33. Elliott D., Keller F. Image Description using Visual Dependency Representations. *In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.* 2013. P. 1292–1302.

34. Lebret R., Pinheiro P. O., Collobert R. Phrase-based Image Captioning. *In Proceedings of the 32nd International Conference on Machine Learning.*

35. Yang Z., Yuan Y., Wu Y., Salakhutdinov R., Cohen W. W. Review Networks for Caption Generation. *In 30th Annual Conference on Neural Information Processing Systems.* 2016. P. 2016–2023.

36. Pedersoli M., Lucas T., Schmid C., Verbeek, J. Areas of Attention for Image Captioning. *In 2017 IEEE International Conference on Computer Vision*. 2017. P. 1251–1259.

37. Lu J., Xiong C., Parikh D., Socher, R. Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning. *In 2017 IEEE Conference on Computer Vision and Pattern Recognition.* 2017. P. 3242–3250.

ДОДАТОК А

Відомість кваліфікаційної роботи

Позначення	Найменування	Дод. відомості
	Текстові документи	
1.	Пояснювальна записка	_92_ с.
	Інші документи	
2.	Презентаційні матеріали	_34_ плакатів

.	.	Прізвище та ініціали.	Підп.	Дата	
Розробив		Стахевич А.В.			Розробка та дослідження методів одноразового навчання для мультимодальних даних
Перевірив		Турута О. П.			
Н.контр.		Малєєва І.А.			
Затв.		Філатов В.О.			

Шифр групи	Код напр./спец.
СШІм-21-1	122
ХНУРЕ кафедра ШІ	