Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження методів генерації опису зображення
(тема)

Виконала: студентка 2 курсу, групи ІПЗм-17-1
спеціальності 121- Інженерія програмного забезпечення
(код і повна назва спеціальності)
спеціалізації Інженерія програмного забезпечення

Кальметьєва М.К.
(прізвище, ініціали)
Керівник к.т.н., доц. Турута О.П.
(прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Дудар З.В.
(підпис) (прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121-Інженерія програмного забезпечення
(код і повна назва)
Спеціалізація Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:

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

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

Студентові Кальметьєвій Марині Костянтинівні
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту): Дослідження методів генерації опису зображення

затверджена наказом по університету від "18 " квітня 2019 р. №546СТ
2. Термін подання студентом роботи (проекту) 26 червня 2019 р.
3. Вихідні дані до роботи (проекту) скрипти для аналізу набору даних, для тренування моделі та отримання прогнозування моделі, пояснювальна записка. Використовувати ОС Linux, середовище інтерактивної розробки Jupyter Notebook.
4. Перелік питань, що потрібно опрацювати в роботі: мета роботи, аналіз проблемної галузі і постановка задачі, огляд існуючих методів з генерації опису зображення, застосування методів для поставленої задачі, аналіз якості моделей генерації опису, пошук способів підвищення якості та аналіз їх ефективності.

(Зворотній бік бланку завдання)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів)

Мета роботи, актуальність дослідження, наявні датасети та метрики, наявні сервіси, загальні методи генерації опису до зображення, модель «Show And Tell», задача оптимізації, розробка моделі генерації підпису до зображення, огляд MS COCO, основні гіперпараметри, вибір архітектури згорткової частини, режими розморозки моделі при переносі навчання, використання аугментацій, приклад роботи моделі.

6 Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	к.т.н., доц. Турута О.П.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка*
1.	Аналіз предметної галузі	19 квітня 2019 р.	
2.	Огляд існуючих методів	27 квітня 2019 р.	
3.	Методи швидкого детектування відрізків ліній	25 травня 2019 р.	
4.	Підготовка пояснювальної записки	26 травня 2019 р.	
5.	Спецчастина	31 травня 2019 р.	
6.	Підготовка презентації та доповіді	06 червня 2019 р.	
7.	Попередній захист	18 червня 2019 р.	
8.	Нормоконтроль, рецензування	25 червня 2019 р.	
9.	Занесення диплома в електронний архів	25 червня 2019 р.	
10.	Допуск до захисту у зав. кафедри	26 червня 2019 р.	
* заповнюється вручну після виконання чергового пункту			

Дата видачі завдання 19 квітня 2019 р.

Студент _____
(підпис)

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

РЕФЕРАТ / ABSTRACT

Пояснювальна записка атестаційної роботи: 101 с., 25 рис., 7 табл., 6 формул, 48 джерел.

ГЕНЕРАЦІЯ ОПИСУ ДО ЗОБРАЖЕНЬ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, РЕКУРЕНТНА НЕЙРОННА МЕРЕЖА, MS COCO, PYTORCH.

Метою роботи є дослідження існуючих методів генерації опису зображень та визначення методу, що надає можливість використовувати функціональність генерації підписів у власних комп'ютерних програмах та бути впевненим у якості цієї функціональності.

Методи розробки базуються на бібліотеках для аналізу даних, написаних на мові Python, та фреймворку глибокого навчання PyTorch. У результаті роботи була розроблена модель генерації підписів до зображення, проведене додаткове налаштування моделі та перевірена якість її роботи.

IMAGE CAPTION GENERATION, CONVOLUTIONAL NEURAL NETWORK, RECURRENT NEURAL NETWORK, MS COCO, PYTORCH.

The aim of this work is to study the existing methods of image caption generation and to determine the method that enables the functionality of captioning to be used in software applications with confidence of the quality.

Development methods are based on data analysis libraries written in Python and deep learning framework PyTorch. As a result of the work, an image caption generation model was developed, tuned and checked to have fine quality.

ЗМІСТ

ВСТУП

Упродовж останніх кількох років галузь машинного навчання набула надзвичайної популярності. Воно й не дивно – на даний момент ця область розвивається неймовірними темпами, щодня наближуючи нас до науково-фантастичного майбутнього, де комп’ютери «думають» та поводяться так само, як люди. Багато хто пов’язує початок «буму» штучного інтелекту зі створенням бази даних з проанотованими зображеннями під назвою ImageNet [1]. ImageNet був першим датасетом з такою кількістю якісних візуальних даних і відкрив двері для використання багатьох алгоритмів машинного навчання, зокрема, глибокого навчання [2]. Це, в свою чергу, дало поштовх для розвитку активності в напрямку задач, які раніше люди навіть не сподівалися вирішити на належному рівні.

Однією з таких задач є генерація підпису до зображень, тобто створення короткого тексту, що описує зміст картинки. Ця задача стоїть на перетині комп’ютерного зору та обробки природної мови, тож в теорії може як використовувати ідеї з обох галузей, так і впливати на обидві одразу.

Однак у порівнянні з традиційними анотаціями зображень на основі ключових слів (за допомогою розпізнавання об'єктів, виявлення атрибутів, маркування сцени тощо), автоматичні системи опису зображень виробляють більш людські пояснення візуального вмісту, забезпечуючи більш повне уявлення про сцену. Ці системи також можуть полягти в основу більш інтелектуальних систем штучного зору, що зуміють робити висновки про сцени через створені описи обґрунтованих зображень і, отже, взаємодіяти зі своїми середовищами більш природним чином.

Також, мабуть, найбільш важливий прямий вплив, що може здійснити система генерації підписів до зображення, – це допомога людям, в яких наявні проблеми із зором. Адже якщо приєднати до цієї системи аудіо-модуль, це буде якісний спосіб зробити візуальну інформацію такою, що сприймається на слух.

Незважаючи на помітне збільшення кількості систем опису зображень в останні роки, експериментальні результати показують, що продуктивність системи все ще дуже далека від людської продуктивності. Метрики та інструменти, що використовуються в даний час, все ще недостатньо корелюються з людськими судженнями, що вказує на необхідність прийняття заходів, які можуть адекватно вирішувати складність проблеми з описом зображення.

Метою даної роботи є дослідження існуючих методів генерації опису зображень та визначення методу, що надає можливість використовувати функціональність генерації підписів у власних комп'ютерних програмах та бути впевненим у якості цієї функціональності.

Для досягнення мети були поставлені наступні задачі:

– загальне порівняння методів генерації опису, наявних метрик та наборів даних;

– розробка моделі генерації підписів до зображення;

– додаткове налаштування моделі для підвищення її якості;

– оцінка якості результуючої моделі та аналіз результатів.

Об'єктом дослідження є створювання коротких підписів, що характеризують зображення.

Предмет дослідження – компроміс між доступністю функціональності генерації підписів та її якістю.

В якості методів дослідження використовується як якісний, так і кількісний аналіз: відбувається збір інформації по цільовій темі, формується детальний опис існуючих рішень та проводиться порівняльний аналіз, а потім в рамках розробки власної моделі відбуваються експерименти із замірюванням числових значень метрик.

Підходи до налаштування моделі, що застосовані у цій роботі, добре зарекомендували себе в інших задачах комп'ютерного зору, але знаходять мало освітлення для генерації підписів до зображень. Досліждення, проведені в рамках цієї роботи, допоможуть перевірити вплив цих підходів на рішення цільової задачі.

1 АНАЛІЗ ПРОБЛЕМНОЇ ГАЛУЗІ

1.1 Огляд задачі генерації опису до зображень

Побачивши якусь складну сцену в житті або намальовану на картині, люди можуть узагальнити її лише кількома словами, не думаючи двічі. Ця ж задача є набагато більш складною для комп'ютерів. Однак на даний момент є рішення, що дозволяють певною мірою робити це й на комп’ютерах – системи, які вміють автоматично створювати підпис до зображення щоб точно описати зображення вперше, коли вони їх «бачать».

Формально задачу генерації підпису до зображення можна представити, як на рисунку 1.1: вхідними даними є тільки зображення, без будь-яких інших метаданих як-то: ключові слова або теги; комп’ютерна система після обробки зображення генерує короткий текст, що описує сутність зображеного.

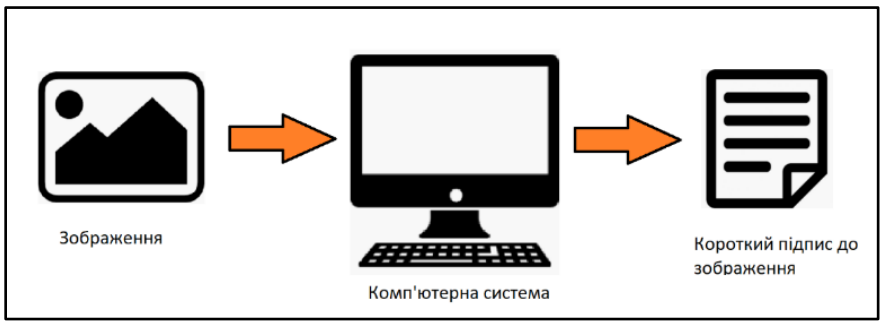

Рисунок 1.1 – Схема задачі генерації підпису

Цільова задача стоїть на перетині галузей комп’ютерного зору та обробки природної мови. Однак ця задача істотно складніше ніж, наприклад, класифікація зображень або завдання розпізнавання об'єктів. Генерація опису повинна охоплювати не тільки об'єкти, що містяться на зображенні, та їх категорії, але також повинна

виражати, який зв’язок між цими об’єктами, а також їхні атрибути, як-то колір чи розмір, та діяльність. Крім того, текст, що генерується, повинен бути зв’язним та відповідати синтаксичним та морфологічним нормам мови. Приклад зображень та відповідних підписів на англійській мові можна побачити на рисунку 1.2.

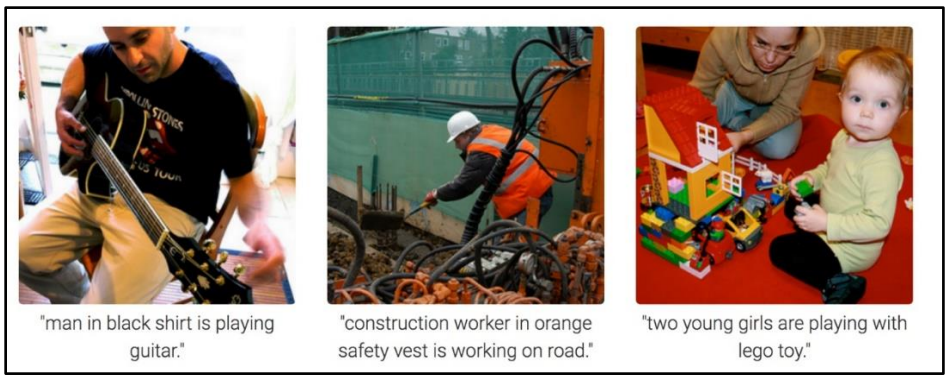

Рисунок 1.2 – Приклад зображень та згенерованих підписів

Системи генерації опису до зображення можуть зрештою привнести чимало користі у людське життя, наприклад:

– допомогти людям з вадами зору зрозуміти сутність зображення;

– надати альтернативний текст для зображень у деяких частинах світу, де мобільні з'єднання є повільними;

– зробити пошук зображень за змістом (content-based image retrieval) більш «розумним», щоб при пошуці зображень максимально враховувалися всі складові частини зображеної сцени та відносини між цими складовими;

– допомогти пояснити події, що відбуваються на відео, кадр за кадром;

– надати джерело додаткової інформації для соціальних медіа платформ, наприклад, щоб збирати більше даних про те, що саме люди роблять на певних заходах чи в певних місцях.

Таким чином, можна вважати, що тема генерації опису до зображень є актуальною та багатообіцяючою.

1.2 Наявні датасети

Для перевірки якості систем, що генерують опис зображень, було створено кілька наборів даних. Розглянемо ці набори від найменшого до найбільшого за розміром.

UIUC Pascal Sentences [3] – це один з найпростіших датасетів, що можна використовувати для вирішення цільової задачі. Цей датасет містить 1000 зображень, що були відібрані з PASCAL VOC 2008, та по 5 підписів, створених людьми, до кожного зображення. PASCAL VOC 2008 – це, загалом, змагання з класифікації та детекції об’єктів з більш ніж 10 тисячами зображень, однак частину зображень вдалося виділити, додатково розмітити, та перевикористати для генерації опису зображення.

При цьому, наявні підписи до зображень є дещо обмеженими з точки зору розмаїття використаної мови. Близько 25% підписів не містять дієслів; 15% містять лише статичні дієслова (інфінитиви), такі як сидіти, стояти, носити, дивитися. Приклад зображення з датасету та підписи можна побачити на рисунку 1.3.

Рисунок 1.3 – Зображення з набору даних UIUC Pascal Sentences та 5 відповідних підписів

FLICKR 8K [4] – другий за розміром датасет. Набір даних містить зображення, отримані з веб-сайту Flickr, та доступ до нього надається університетом Іллінойсу. Датасет містить 8108 зображень та 5 речень до кожного зображення.

Цей набір даних демонструє більшу різноманітність мови: 11% підписів не мають дієслів, 10% мають лише загальні статичні дієслова. Також характерною рисою цього датасету є наявність великої кількісті зображень людей і тварин (в основному собак).

FLICKR 30K [5] являє собою розширення датасету FLICKR 8K, налічує 31,783 зображень, також з 5 реченнями до зображення.

Автори датасету також надають до вільного доступу так званий граф візуальних позначень (visual denotation graph). Цей граф, по суті, пов’язує між собою велику кількість лінгвістичних виразів, що зустрічалися у підписах в датасеті, та зображення, що відповідають кожному лінгвістичному виразу (рис. 1.4).

Граф є досить корисним через наявність у ньому батьківських та дочірніх зв’язків, таким чином по ньому можно переходити від більш загальних до більш конкретних описів.

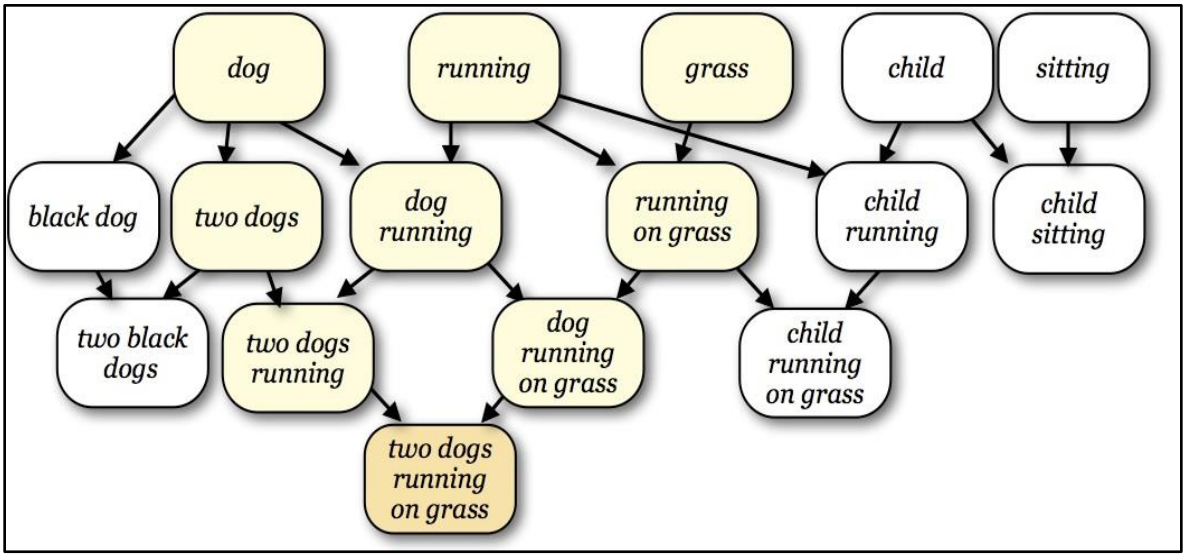

Рисунок 1.4 – Фрагмент графу візуальних позначень для FLICKR 30K

При використанні цього графу пошук зображень по тексту можна звести до асоціації тексту запиту з тегами зображення (листовим вузлом графу та його батьківських вузлів). Ідентифікація зображень таким чином стає задачею мультикласової класифікації зображень за окремими словами або тегами. Цю задачу можна вважати декілька спрощеною задачею генерації опису зображення, адже ми встановлюємо асоціацію не з цілком новим згенерованим підписом, а з тегами, наявними у нашій базі.

Приклад вузлу графу наведений на рисунку 1.5.

Рисунок 1.5 – Вузол графу, що пов’язує вираз «two dogs running on grass» з відповідними зображеннями

MS COCO[6] дуже довгий час був найбільшим датасетом з наявними зображеннями та підписами: близько 120 тисяч зображень з 5 підписами до кожного зображення. Відмінною рисою датасету є також те, що кожен об’єкт, використаний у будь-якому підписі, проанотований категорією (однією з наявних 80 категорій) та просегментований, тобто наявні координати обмежувальної рамки на зображенні, що задають точні координати об'єкта (рисунок 1.6), й таким чином можна встановити приналежність кожного пікселя до певного класу.

На датасеті MS COCO з березня 2015 року й по сьогоднішній день відкрите змагання з генерації підписів до зображень. Саме на сервері цього змагання перевіряються нові моделі та визначаються найбільш точні та «проривні» рішення.

Насправді на даний момент кількість зображень у MS COCO налічує близько 330 тисяч, однак із них промаркованими є лише близько 200 тисяч. Але для задачі генерації підписів до зображень зазвичай використовуються саме початкові 120 тисяч зображень, адже змагання проводиться і проводилося саме на них, і саме ці дані можна вважати найбільш «надійними» та перевіреними часом.

а) б)

a little dog sitting on a wooden bench.
a dog is sitting on a bench by corn and hay.
a small dog sits on wooden bench next to a pumpkin.
a dog sitting on a bench with harvest decorations.
a small dog on a bench next to squash, hay and variegated corn

в)

Рисунок 1.6 – Приклад з набору даних MS COCO: а) Початкове зображення; б) Зображення з просегментованими об’єктами; в) П’ять підписів до зображення.

Conceptual Captions [7] – найновіший та на даний момент найбільший датасет для генерації підписів, що був викладений до вільного доступу компанією Google у вересні 2018 року. Він налічує близько 3,3 мільйонів пар зображення/підпис (рис. 1.7), що є на цілий порядок більшим за обсягом ніж MS COCO. Набір даних Conceptual

Captions не був вручну промаркований людьми, а був створений шляхом автоматичного вилучення та надзвичайно ретельної фільтрації підписів до зображень, вказаних через відповідний HTML-тег, з мільярдів веб-сторінок.

За оцінками експертів з Google, точність підписів складає близько 90%. Крім того, оскільки зображення в Conceptual Captions витягуються з Інтернету, вони представляють більш широкий спектр тематик та стилів підписів, ніж попередні набори даних, що дозволяє краще навчатися моделям генерації підписів.

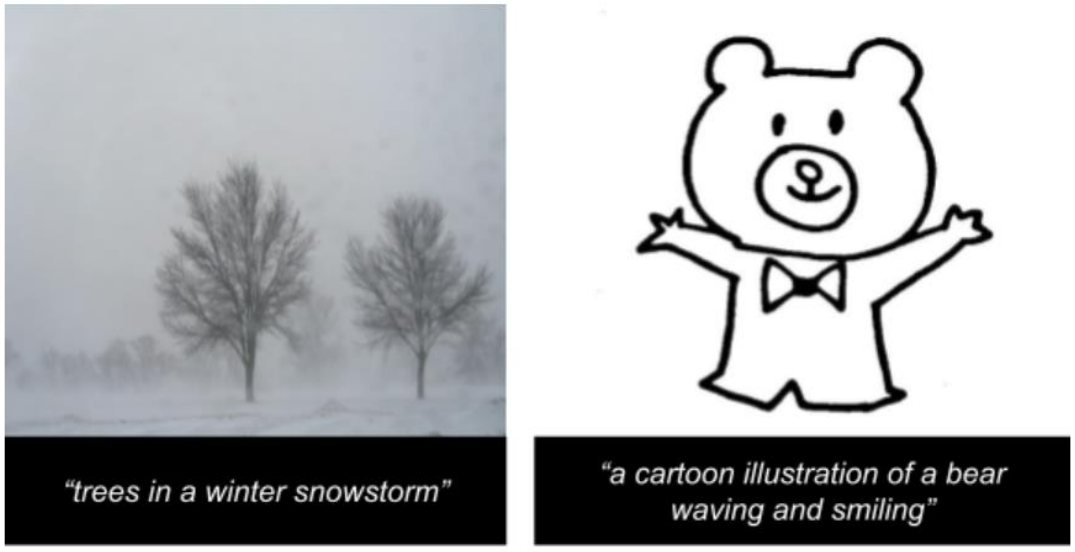

Рисунок 1.7 – Приклад зображень та підписів з набору даних Conceptual Captions

Щоб відслідковувати прогрес у вирішенні задачі генерації підписів, Google також відкрив своє змагання на датасеті Conceptual Captions. Хоча еталонним змаганням для перевірки систем генерації підписів усе ще є змагання MS COCO, у майбутньому саме Conceptual Captions може завоювати першість.

У Conceptual Captions тренувальна вибірка налічує 3 318 333 елементів з 51 201 унікальними токенами, валідаційна вибірка – 15 840 елементів з 10 900 токенами, і тестову (тобто приховану) вибірку з 12 559 зображеннями та 9 645 токенами.

1.3 Метрики

І при розробці, і при використанні системи генерації опису зображень необхідно мати уявлення, наскільки якісною є цільова система. У цьому допомагає використання спеціальних метрик, пристосованих до цієї задачі.

Основними метриками, що використовуються для оцінки задачі генерації підписів до зображень, є наступні (розташовані у порядку від вигаданих давно до відносно нових):

– BLEU (BiLingual Evaluation Understudy) [8] і його різновиди BLEU-1, BLEU-2, BLEU-3, BLEU-4;

– ROUGE (Recall Oriented Understudy of Gisting Evaluation) [9] та його модифікації;

– METEOR (Metric for Evaluation of Translation with Explicit ORdering) [10];

– CIDEr (Consensus-based Image Description Evaluation) та його варіація CIDEr-D [11];

– SPICE [12].

Розглянемо метрики детальніше.

BLEU вимірює точність (precision): скільки слів або N-грам в машинно-згенерованих підписах з'явилися в початкових, сформованих людьми, підписах. При цьому метрика BLEU являє собою дещо модифікований алгоритм підрахунку точності, а не стандартний. BLEU була першою метрикою, для якої була заявлена висока кореляція з людськими сужденнями щодо схожості підписів.

Розглянемо приклад вимірювання точності генерації підпису, що наведений у статті Papineni та інших [8]. Цей приклад використовувався для для оцінки алгоритмів машинного перекладу, однак для оцінки генерації підписів до зображення він теж відмінно підходить.

Таблиця 1.1 – Приклад порівняння слів у підписах, сформованих комп’ютером та людьми

Вид підпису	Слова підпису						
	1	2	3	4	5	6	7
Підпис, запропонований комп’ютером	the	the	the	the	the	the	the
Сформований людиною підпис №1	the	cat	is	on	the	mat	
Сформований людиною підпис №2	there	is	a	cat	on	the	mat

Як ми бачимо, комп’ютерна система дещо «схитрила» та згенерувала підпис, що складається з єдиного слова «the». Якби BLEU вимірювала стандартну точність відносно слів, то вона була би такою:

$$P = \frac{m}{w_t} = \frac{7}{7} = 1 \qquad (1.1)$$

де P – точність;

m – кількість слів у згенерованому комп’ютером підписі, що містяться у людських підписах;

w_t – загальна кількість слів у згенерованому комп’ютером підписі.

Таким чином, при підрахунку «класичної» точності ми отримали найвищу можливу оцінку, нібито підпис від комп’ютера був ідеальним. Авжеж, нас ця ситуація не враховує, і саме тому у BLEU алгоритм вирахування точності дещо модифікований. Для кожного слова w в запропонованому підписі алгоритм підраховує $m_{\max}$ – максимальну кількість разів, котру слово зустрічається у будь-якому з людських підписів.

У наведеному вище прикладі слово «the» з'являється двічі в людському підписі № 1, і один раз у підписі 2. Відтепер маємо $m_{\max}$ = 2. Для запропонованого

комп’ютером підпису підрахунок m_w кожного слова «підрізається» до значення $m_{\max}$ для цього слова. У нашому випадку «the» має m_w = 7 і $m_{\max}$ = 2, таким чином m_w підрізається до 2. Ці обрізані лічильники m_w потім підсумовуються по всіх окремих словах у запропонованому підписі, і сума ділиться на загальну кількість слів у цьому підписі. У наведеному вище прикладі BLEU буде дорівнювати:

$$BLEU = \frac{\sum_w cropped(m_w)}{w_t} = \frac{\sum_w \min(m_w, [m_{max}]_w)}{w_t} = \frac{2}{7} \qquad (1.2)$$

де $BLEU$ – цільова метрика;

m_w – кількість слів w у згенерованому підписі, що співпадають зі словами в людських підписах;

$[m_{max}]_w$ – максимальна кількість разів, котру слово w зустрічається у будь-якому з людських підписів;

w_t – загальна кількість слів у згенерованому комп’ютером підписі.

На практиці використання слів в якості одиниці порівняння не є оптимальним. Замість цього BLEU обчислює ту ж саму модифіковану метрику точності, використовуючи поєднання послідовних слів, тобто N-грами. Найчастіше використовують BLEU для N-грамів з N від 1 (маються на увазі уніграми – окремі слова) до 4, і позначається ця метрика відповідними назвами BLEU-1, BLEU-2, BLEU-3 та BLEU-4.

Одною з суттєвих проблем BLEU є те, що ця метрика часто надає перевагу коротким підписам, і це може призводити до того, що комп'ютерна система навмисно буде генерувати лише якісь ключові слова. Наприклад, короткий підпис «the cat» для проаналізованого вище прикладу привів би до результату BLEU-1 рівного 1, тобто найвищого можливого. Щоб «стимулювати» комп'ютерну систему генерувати підписи й більшої довжини, використовуються модифікації метрики BLEU,

наприклад метрика NIST, або ж для оцінки роботи системи використовується поєднання метрик BLEU та ROUGE.

Метрика ROUGE схожа на BLEU, однак вона вимірює повноту (recall): скільки слів або N-грам з наданих людьми описів зустрічаються в підписах, створених комп'ютерною системою. Існує декілька різновидів метрики ROUGE, запропонованих авторами [9]:

– ROUGE-N: враховує перекриття N-грам між запропонованим комп'ютером та людськими підписами. Приклади конкретних метрик: ROUGE-1 для окремих слів, ROUGE-2 для біграм;

– ROUGE-L [13]: використовує статистику на основі найдовшої загальної підпослідовності (LCS – Longest Common Subsequence). Ідея полягає в тому, що чим довше LCS двох підписів, тим більш схожими є ці підписи. Для оцінки подібності між двома реченнями використовується заснований на LCS показник F1-score;

– ROUGE-W: зважена статистика на основі LCS, що надає перевагу знайденим LCS, що розташовані послідовно;

– ROUGE-S: статистика на основі біграмів з пропусками;

– ROUGE-SU: статистика, у якій враховуються і біграми з пропусками, і уніграми.

Найбільш поширеною є метрика ROUGE-L – саме вона використовується у змаганнях з генерації опису до зображення MS COCO Challenge [14] та Conceptual Captions Challenge [7].

Метрика METEOR ґрунтується на середньому гармонійному від точності (precision) та повноти (recall) уніграмів, причому повнота враховується з більшим ваговим коефіцієнтом, аніж точність. METEOR також має декілька методів встановлення відповідності між підписами:

– слова вважаються однаковими, якщо їх написання повністю ідентичне;

– слова предоброблюються за допомогою стемінгу, тобто кінцівка кожного слова відсікається, залишаючи від слова лише головну, найбільш значущу частину; отримані терми перевіряються на ідентичність написання;

– слова вважаються однаковими, якщо вони є синонімами;

– фрази є однаковими, якщо вони є перифразами в рамках цільової мови.

Метрика була розроблена, щоб виправити деякі з проблем, що притаманні популярній метриці BLEU, а також забезпечити гарну кореляцію з людським судженням щодо схожості підписів на рівні словосполучень або речень. Для вирахування метрики можуть використовуватися різні алгоритми стемінгу. Огляд цих аглоритмів представлений у додатку В.

Усі метрики, що були розглянуті до цього моменту, спочатку розроблялися як метрики для оцінювання якості машинного перекладу, і можна вважати щасливим збігом те, що вони підходять і до задачі генерації опису до зображення. А ось CIDEr – це метрика, що була створена саме для оцінки генерації підписів до зображень.

При підрахуванні CIDEr до всіх слів застосовується стемінг, таким чином від слів залишаються лише кореневі форми. Після цього кожне речення представляється як множина N-грамів, що присутні у реченні. Автори статті використовують N-грами з N від 1 до 4. Також у CIDEr відіграє роль те, що N-грами, які часто зустрічаються для зображень з датасету, враховуються с меншим ваговим коефіцієнтом, адже через їхню поширеність вони несуть меншу інформаційну користь. Для реалізвації цієї ідеї використовується показник TF-IDF (term frequency–inverse document frequency) [15]. Після переводу множини N-грамів до вектору значень TF-IDF, метрика CIDEr для N-грамів довжини N вираховується як косинус подібності між згенерованим реченням та людськими реченнями, що враховує одразу і точність і повноту:

$$CIDEr_n(c_i, S_i) = \frac{1}{m}\sum_j \frac{g^n(c_i) \cdot g^n(s_{ij})}{\|g^n(c_i)\|\|g^n(s_{ij})\|} \quad (1.3)$$

де c_i – згенероване комп’ютером речення;

S_i – набір сформованих людьми речень;

m – кількість сформованих людьми речень;

$g^n(c_i)$ – вектор TF-IDF, сформований для усіх N-грамів довжини N для згенерованого речення c_i;

$\|g^n(c_i)\|$ – скаляр вектору TF-IDF величин для згенерованого речення c_i;

$g^n(s_{ij})$ – вектор TF-IDF, сформований для усіх N-грамів довжини N для людського підпису s_{ij};

$\left\|g^n(s_{ij})\right\|$ – скаляр вектор TF-IDF, сформований для людського підпису s_{ij}.

При цьому метрики для N-грамів з різною довжиною комбінуються у єдину метрику наступним чином:

$$CIDEr(c_i, S_i) = \sum_{n=1}^{N} w_n \cdot CIDEr_n(c_i, S_i) \qquad (1.4)$$

де c_i – згенероване комп'ютером речення;

S_i – набір сформованих людьми речень;

w_n – ваговий коефіцієнт для окремої метрики $CIDEr$ для N-грамів довжини n;

$CIDEr_n(c_i, S_i)$ - окрема метрика $CIDEr$ для N-грамів довжини n.

Емпіричним шляхом було встановлено, що однорідні ваги $w_n = \frac{1}{N}$ працюють найкраще. Автори статті доводять, що метрика CIDEr дає оцінку схожості двох підписів більш наближену до людської оцінки, ніж метрики BLEU, ROUGE та METEOR.

Після того, як метрика CIDEr була уведена на змаганні MS COCO, були виявлені деякі її недоліки, і була запропонована метрика CIDEr-D. CIDEr-D перевикористовує ідею метрики CIDEr з такими змінами:

– у CIDEr-D не використовується стемінг, аби упевнитися в правильній формі згенерованих слів;

– уводиться штраф, що являє собою Гаусову функцію від різниці довжин підписів, адже було помічено, що метрика CIDEr у деяких випадках дає згенерованому реченню більшу оцінку просто якщо в ньому декілька разів повторити слово з великим TF-IDF. Для Гаусової функції автори використовують σ = 6;

– у чисельнику формули кількість N-грамів запропонованого речення урізається до кількості N-грамів з речень, сформованими людьми, також щоб не можна було повторювати слова з високим TF-IDF багато разів для досягнення бажаної довжини підпису;

– до усієї формули доданий множник 10 аби результат метрики чисельно був схожий на інші метрики.

Метрика CIDEr-D для N-грамів певної довжини розраховується наступним чином:

$$CIDEr\text{-}D_n(c_i, S_i) = \frac{10}{m}\sum_j e^{\frac{-\left(l(c_i)-l(s_{ij})\right)^2}{2\sigma^2}} * \frac{\min(g^n(c_i),\ g^n(s_{ij})) \cdot g^n(s_{ij})}{\|g^n(c_i)\|\left\|g^n(s_{ij})\right\|} \qquad (1.5)$$

де c_i, S_i, m, $g^n(c_i)$, $\|g^n(c_i)\|$, $g^n(s_{ij})$, $\left\|g^n(s_{ij})\right\|$ мають ті ж самі значення, що і у формулі 1.4.

$l(c_i)$ – довжина згенерованого підпису;

$l(s_{ij})$ – довжина людського підпису.

Метрика CIDEr-D, узагальнена для усіх N-грамів, вираховується аналогічно до метрики CIDEr.

SPICE – це відносно нова метрика, що була представлена у статті 2016 року. Вона також, як і метрика CIDEr була запропонована саме для задачі генерації підписів до зображень. Головною ідеєю є врахування семантичної структури підписів при

оцінці якості їх генерації. Метрика SPICE вираховується дещо складніше, аніж інші метрики, та її якість дуже залежить від якості семантичного графа, що використовується.

Однак автори сттатті доказали, що ця метрика гарно корелює з людським представленням про схожість описів, тож ця метрика використовується у змаганнях з генерації підписів по зображеннях та в цілому є досить багатообіцяючою.

У MS COCO Challenge використовується більшість розглянутих нами метрик, а саме: BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, METEOR, CIDEr-D, SPICE. Основною метрикою вважається CIDEr.

У Conceptual Captions challenge використовуються CIDEr (також як основна метрика), ROUGE-L та SPICE.

1.4 Доступні сервіси, що генерують підписи до зображень

Сервіси, що надають можливість генерувати підписи до зображень онлайн, присутні в досить обмеженій кількості. Список майже повністю обмежується наступними:

- Microsoft Cognitive Services [16];
- Microsoft Caption Bot [17];
- IBM Model Asset eXchange (MAX) [18].

Розглянемо ці сервіси детальніше.

Перш за все, можливість генерації опису до зображення доступна в рамках пакету Computer Vision сервісів Microsoft Cognitive Services, що надають алгоритми машинного навчання розробникам для зручного впровадження AI у їх власні системи. Підпис до зображення можна отримати разом з переліком об’єктів виявлених на рисунку, а також спеціальними тегами, що також відображають зміст зображеного.

Computer Vision сервіс створює лише один підпис та виводить його вірогідність. Приклад роботи приведений на рисунку 1.8.

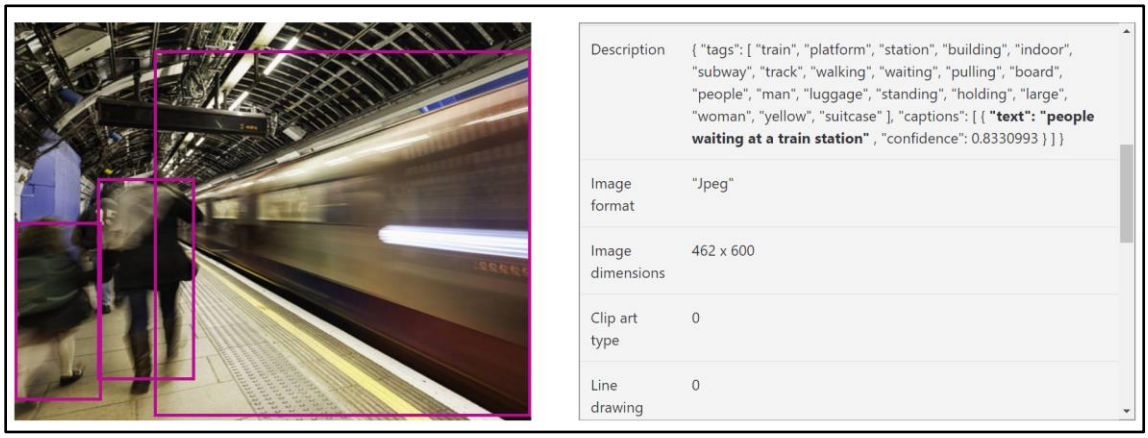

Рисунок 1.8 – Генерація підпису до зображення через Microsoft Cognitive Services API

Використання цієї функціональності регулюється стандартними тарифами Microsoft Cognitive Services та наразі входить у пакет, що коштує 2.5 USD за 1000 операцій [19]. До речі, на даний момент цей пакет – найдорожчий серед пакетів, тарифів у категорії комп'ютерного зору Cognitive Services. На даний момент Microsoft надають не дуже багато інформації щодо внутрішньої реалізації функціональності. Перевагою сервісу є те, що генерація опису доступна не тільки на англійській мові, однак про інші мови компанія зазначає, що генерація відбувається все ще під «людським наглядом», тобто є у певній мірі тестовою функцією.

Сервіс Microsoft Caption Bot є дещо «іграшковим» та дозволяє швидко оцінити, наскільки взагалі якісним може бути генерація опису до зображення, та зокрема генерація через сервіси Microsoft. У Caption Bot можна завантажувати по одному зображенню на сайт та отримувати відповідний опис. Ні зображення, ні підпис ніде не зберігаються для повторного використання. Вірогідність згенерованого підпису користувачам не надається, однак, судячи з усього, підписи з дуже низькою

вірогідністю не виводяться, адже іноді сервіс видає інформацію, що не може створити опис. Користувач додатково має можливість оцінити якість роботи сервісу по шкалі від одного до п’яти, таким чином надаючи зворотній зв’язок Microsoft (рис. 1.9).

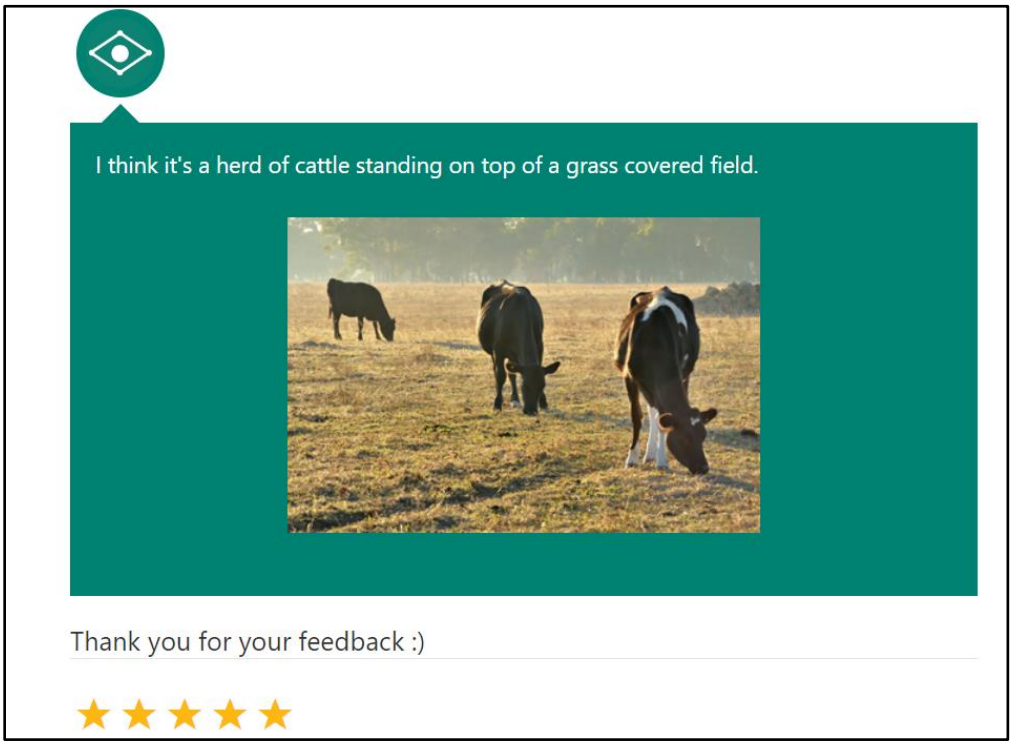

Рисунок 1.9 – Генерація підпису до зображення через Microsoft Caption Bot

В основі Caption Bot лежать ідеї, описані в 2016 році співробітниками Microsoft у статті «Rich Image Captioning in the Wild» [20]. Однак на даний момент тут використовуються ті ж моделі, що і в Microsoft Cognitive Services, що, безумовно, відрізняються від початкової реалізації.

IBM надає сервіс генерації опису до зображень в рамках MAX (Model Asset eXchange) – програми надання повністю відкритого доступу розробникам до моделей глибокого навчання, які можна тренувати та розгортати. MAX модель генерації підписів наявна у вигляді вихідного коду, Docker образу, API, що підключається через Node-RED, та демо-додатку. Загалом ця MAX модель являє собою базову модель,

описану у 2014 році у статті «Show and Tell: A Neural Image Caption Generator» [21], що буде розглянута далі у підрозділі 2.4. Приклад роботи API представлений на рисунку 1.10.

Рисунок 1.10 – Генерація підпису до зображення через IBM MAX API

Таким чином MAX модель IBM – це відома, навіть «класична» модель генерації опису до зображення, поверх котрої зробили обгортку для її зручного розгортання на власному обладнанні або через IBM Bluemix.

Як ми бачимо, при виникненні необхідності використання генерації опису до зображення у власній програмній системі, найпростіші можливі варіанти – це використання API у Microsoft Cognitive Services чи розгортання IBM MAX моделі. Перевагами Microsoft Cognitive Services є те, що з ними можна розраховувати на високу якість згенерованого тексту (хоча точна якість ніде компанією не зазначається), однак за це треба буде поплатитися надзвичайно високою ціною за

використання. Якість моделі IBM, у свою чергу, можна дізнатися з наукової статті [21], однак з часу написання статті у 2014 році сфера глибокого навчання дуже просунулася вперед, і на даний момент можна сподіватися отримати помітно кращі результати генерації.

Таким чином, якщо брати в розрахунок проаналізовані переваги та недоліки існуючих сервісів, розумною альтернативою сервісам можна вважати розробку власної моделі генерації опису. Ця модель, скоріш за все, буде фінансово вигіднішою, ніж Microsoft Cognitive Services, і при цьому її використання може бути більш гнучке, наприклад, з врахуванням не одного, найбільш вірогідного підпису, а певної кількості найвірогідніших. При розробці своєї моделі є можливість власноруч проводити оцінку її ефективності та обширний аналіз роботи саме на своїх цільових даних та для своїх задач. Це також дозволяє виконувати додаткове налаштування чи навіть зміни в моделі для досягнення кращих результатів, аніж, наприклад, при використанні IBM MAX моделі.

Тому задача дослідження в даній науковій роботі складається в розробці власної моделі генерації опису до зображення та налаштування моделі для покращення якості її роботи.

2 ПОСТАНОВКА ЗАДАЧІ

2.1 Загальні методи генерації опису до зображення

Усі методи генерації опису зображень можна умовно поділити три обширні категорії [22]. Ці категорії включають в себе:

– створення підпису на основі шаблонів;

– опис на основі пошуку;

– використання підходів глибокого навчання.

Підходи на основі шаблонів використовують фіксовані шаблони з певною кількістю пустих слотів для створення підписів. У цих підходах спочатку виявляються різні об'єкти, атрибути, дії, а потім пробіли в шаблонах заповнюються. Таким чином цей метод є досить обмеженим, адже шаблони попередньо визначені. Прикладом такого підходу є работа Li та інших [23].

У підходах на основі пошуку підписи вибираються з попередньо заданого набору підписів. Спочатку відшукуються візуально подібні зображення з існуючого набору даних, та виявляються відповідні підписи. Ці підписи можна вважати кандидатами на роль шуканого підпису.

Результуючий підпис обираються з пулу підписів-кандидатів. Тож цей метод дає загальні та синтаксично правильні підписи, однак він не може генерувати специфічні речення, детально описуючі конкретне зображення. Цей метод, наприклад, використовується у роботі M. Hodosh [4], в якій оформлення опису зображення розглядається як задача ранжування.

Загальний хід дій при використанні підходів глибокого навчання полягає в початковому аналізі візуального вмісту зображення і в подальшій генерації підпису з візуального вмісту за допомогою мовної моделі. Цей метод може генерувати істотно нові підписи для кожного зображення. При цьому наукові дослідження показують, що згенеровані підписи є також семантично більш точними, ніж підписи, сформовані

двома попередньо розглянутими методами. Приклад роботи системи глибокого навчання, що створює цілком новий підпис, специфічний для тестового зображення, приведений на рисунку 2.1.

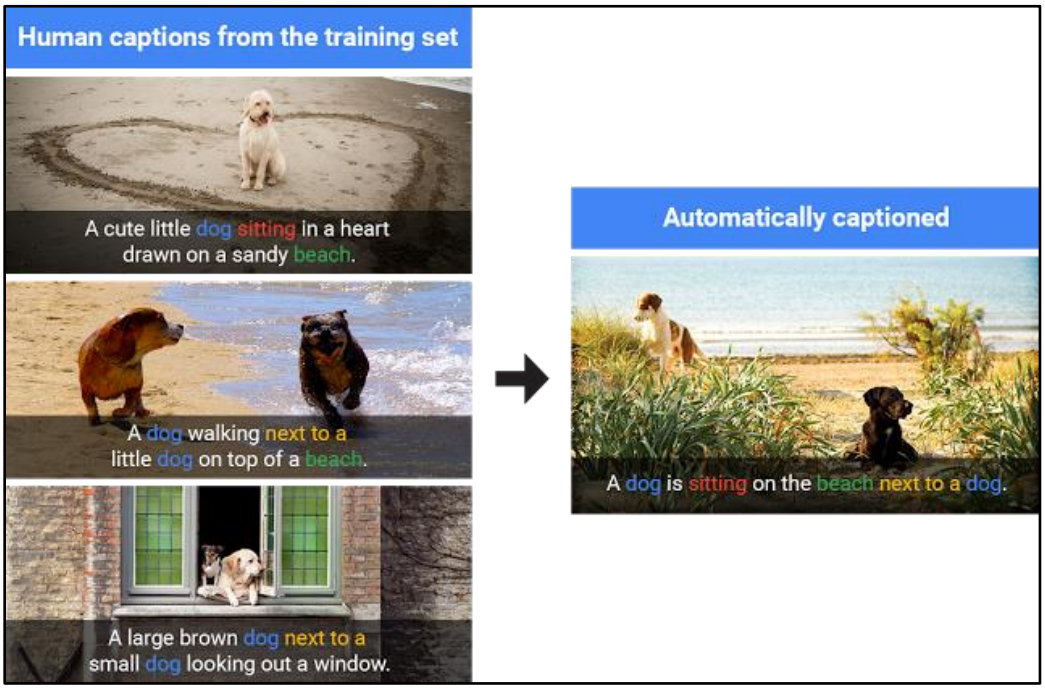

Рисунок 2.1 – Приклад роботи системи глибокого машинного навчання, що генерує підписи до зображення

Орієнтація на генерацію нових підписів, специфічних для зображення, як наслідок призводить до кращої узагальнюючої здібності моделі.

Переважна кількість наукових робіт на тему генерації опису зображення, що була опублікована в останні роки, використовує саме глибинне навчання через значно кращі результати генерації [24]. Тому у цій роботі також було прийнято рішення зосередитися більше на цьому методі. Для кращого розуміння принципів його роботи необхідно зробити огляд згорткових нейронних мереж (CNN) та рекурентних нейронних мереж (RNN).

2.2 Огляд роботи CNN

Згорткові нейронні мережі (CNN) – це особлива архітектура штучних нейронних мереж, запропонована в 1989 році відомим науковцем в галузі комп’ютерних наук Яном ЛеКуном (Yann LeCun) [25]. CNN використовує деякі ідеї, що були почерпнуті зі спостерігання за біологічними процесами та будовою зорової кори головного мозку тварин. На сьогодні використання CNN є надзвичайно поширеним в системах розпізнавання зображень та аналізу відео [26].

CNN добре підходять для роботи з візуальними даними, адже, у порівнянні з традиційними нейронними мережами (наприклад, багатошаровим перцептроном), вони використовують значно меншу кількість вагових коефіцієнтів, а також ефективно вилучають ознаки (features) зображення. Останнє досягається шляхом розумного використання інформації про просторові зв’язки пікселей: спочатку шляхом згортки ми отримуємо ознаки, що враховують сусідні пікселі, потім будуємо ознаки більш високого рівня на їх основі, беручи в розрахунок пікселі, вже більше віддалені один від одного. Ця процедура виконується далі й далі, і тільки у самому кінці відбувається прогнозування.

Таким чином, якщо на зображенні є, скажімо, пес, згорткова нейронна мережа зможе вилучити ознаки, характерні для морди пса, і ці ж ознаки будуть використовуватися незалежно від того, чи розташований пес знизу зображення, чи зверху, головне – що морда пса являє собою сукупність пікселів, і мережа змогла розпізнати певну особливість розташування сукупності пікселів. Повнозв’язна нейронна мережа, зі свого боку, була би чутлива до місця розташування собаки, та коректувала би свої вагові коефіцієнти, щоб підлаштуватися під нове розташування.

Розглянемо будову CNN. Вона складається з вхідних та вихідних шарів, а також із декількох прихованих шарів. Приховані шари CNN зазвичай включають в себе згорткові (convolutional), агрегувальні (pooling), повноз'єднані (fully connected) шари.

Першим завжди виступає шар згортки. У згортку вводиться зображення (матриця з піксельними значеннями). Уявіть, що читання вхідної матриці зображення починається у верхній лівій частині. Далі система бере матрицю меншого розміру, яка називається фільтром (або нейроном, або ядром), й виробляє згортку, тобто рухається меншою матрицею вздовж вхідного зображення. Завдання фільтра – помножити його значення на початкові значення пікселів, а результат підсумувати й отримати в кінці одне число (рис. 2.2). Оскільки фільтр зчитує зображення лише у верхньому лівому куті, він зсувається далі і далі праворуч по одному пікселю, виконуючи подібну операцію. Після проходження фільтра по всіх позиціях отримують нову матрицю, меншу за розміром, ніж вхідна матриця.

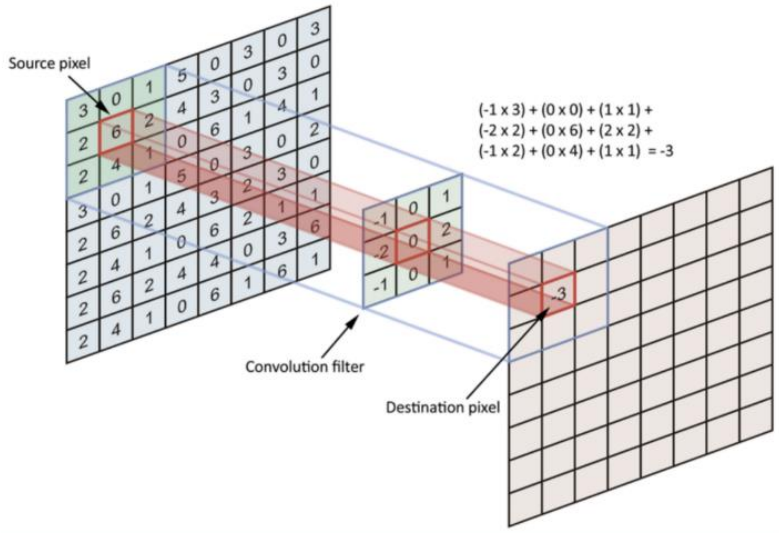

Рисунок 2.2 – Робота згорткового шару у CNN

Після кожної операції згортки в мережі присутній нелінійний шар. Цей шар являє собою функцію активації (часто ReLU), яка приносить до мережі нелінійну властивість. Якби нелінійної складової не було, то всі лінійні шари можна було б об’єднати в один, і нейронна мережа мала би поведінку аналогічну до моделі лінійної регресії, була б дуже обмеженою й показувала гірші результати.

За нелінійним шаром йде агрегувальний. Агрегувальні шари зменшують розміри даних завдяки об'єднанню виходів кластерів нейронів на певному шарі в один нейрон в наступному шарі. Це частково запобігає перенавчанню, надаючи абстраговану форму представлення. Крім того, це знижує обчислювальну вартість, зменшуючи кількість параметрів для навчання.

Повнозв’язні шари розташовуються наприкінці мережі, перед вихідним шаром. Повнозв’язний шар з'єднує кожний нейрон в одному шарі з кожним нейроном в іншому шарі. В принципі це те ж саме, що і традиційна нейронна мережа (багатошаровий перцептрон). «Витягнута» матриця проходить через повнозв’язний шар для прогнозування, наприклад, класифікації зображень.

Схема архітектури CNN предсвлена на рисунку 2.3.

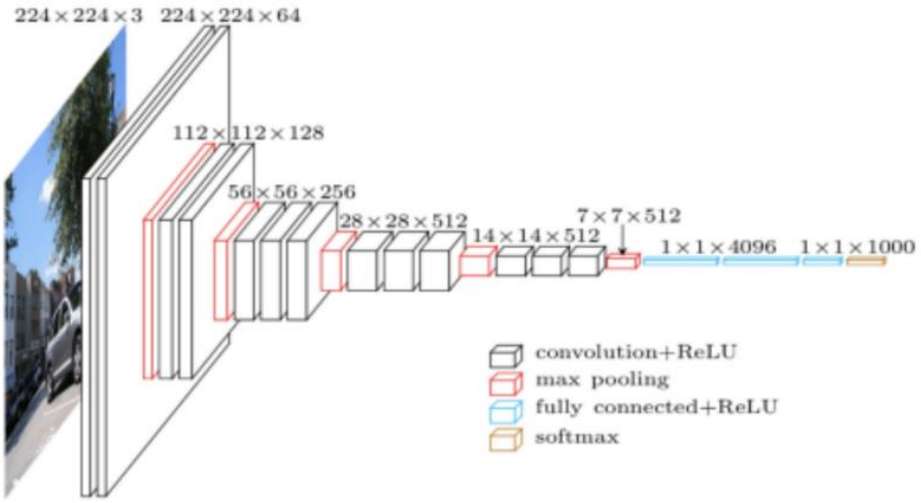

Рисунок 2.3 – Приклад архітектури стандартної CNN

Перші шари мережі вивчають примітивні ознаки: ребра, кути тощо. Середні шари вивчають ознаки, які виявляють частини об'єктів. Останні шари високорівневі: вони розпізнають повні об'єкти, в різних формах і положеннях.

2.3 Огляд роботи RNN та LSTM

Рекурентні нейронні мережі (RNN) – це окремий клас штучних нейронних мереж, які приймають на вхід не тільки поточний елемент даних, а й те, що мережа «вивчила» нещодавно. Рекурентні нейронні мережі містять зворотні зв’язки та дозволяють якби зберігати інформацію, тому часто говорять, що у RNN «є пам’ять». RNN добре підходить для роботи з послідовними даними, наприклад часовими рядами чи зв’язним текстом. Рекурентну нейронну мережу можна представити як декілька копій однієї й тієї ж мережі, причому кожна копія передає інформацію наступній (рис. 2.4).

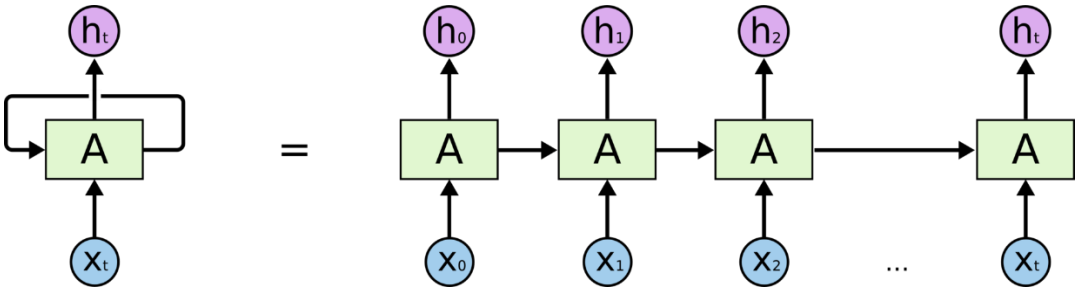

Рисунок 2.4 – Розгорнута рекурентна нейронна мережа

Стандартна RNN дуже схильна до проблеми зникання градієнту (vanishing gradient problem). У двох словах, проблема полягає в тому, що на кожному кроці часу під час тренування ми використовуємо однакові ваги для обчислення h_t. Ці ж ваги враховуються під час зворотного поширення помилки (backpropagation). Чим далі ми рухаємося назад, тим більшим або меншим стає наш сигнал помилки, тобто відбувається певне накопичення. Це призводить до того, що мережа має труднощі в запам'ятовуванні віддаленої інформації і робить прогнози на основі лише нещодавньої. Аби якось впоратися з цією проблемою була розроблена LSTM.

LSTM-мережа (англ. long short-term memory – довга короткочасна пам’ять) – особлива архітектура RNN, що здатна вивчати довгочасні залежності. LSTM добре працює при наявності між важливими подіями часових затримок невідомої, навіть дуже великої, тривалості [27]. Саме мала чутливість LSTM до розміру проміжків між використаною інформацією робить її потужним інструментом для обробки та генерації послідовних даних. LSTM за будовою також можна представити в формі ланцюга, однак кожен вузол цього ланцюга, на відміну від стандартної RNN, містить чотири шари замість одного (рис. 2.5).

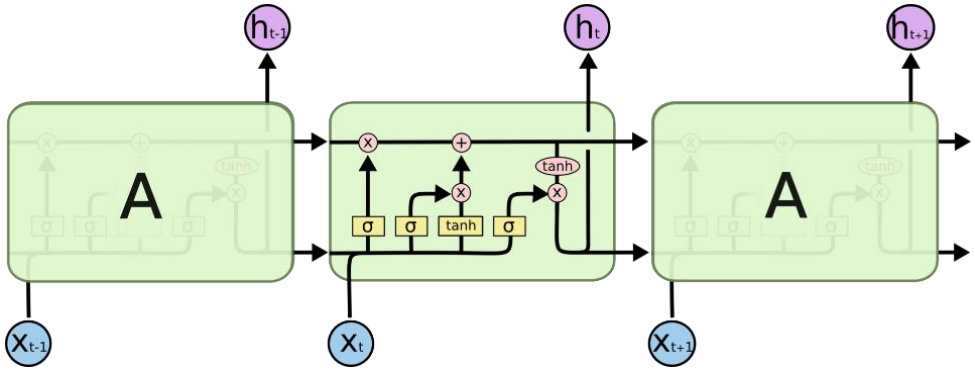

Рисунок 2.5 – Розгорнута схема LSTM

Кожен блок RNN складається із комірки (cell), що як раз і являє собою «пам’ять» блоку, та три «регулятори»: фільтр входу (input gate), фільтр забування (forget gate) та фільтр виходу (output gate).

Якщо коротко, комірка відповідає за виявлення залежностей між частинами вхідної послідовності. Фільтр входу визначає міру надходження нових значень до комірки, фільтр забування визначає міру збереження інформації в комірці, а фільтр виходу характеризує те, наскільки сильно значення в комірці впливає на вирахування виходу блоку LSTM. В якості функції активації у LSTM-фільтрах часто призначають логістичну (сигмоїдну) функцію.

2.4 Використання глибокого навчання для генерації підписів

У 2014 році дослідники з Google опублікували визначну статтю «Show And Tell: A Neural Image Caption Generator» [21], що вперше запропонувала використовувати моделі глибинного навчання для задачі генерації опису зображень. У той час ця архітектура давала найкращі результати на наборі даних MSCOCO. У статті використовували зв’язку CNN та LSTM, щоб взяти зображення в якості вхідних даних і видати опис зображення на вихід. Що цікаво, приблизно в той же час була опублікована стаття A. Karpathy та інших, що використовує дуже схожі ідеї поєднання двох нейронних мереж «Deep Visual-Semantic Alignments for Generating Image Descriptions» [28], однак з декілька іншою реалізацією.

Значна кількість наукових робіт, спрямованих на побудову комп'ютером природних описів зображень та написаних до 2014 року, пропонувала окремо обрати сучасні підходи як в комп'ютерному зорі, так і в обробці природної мови, та послідовно розташувати рішення цих підзадач для формування повного рішення до генерації опису зображень. Наприклад, такий підхід наявний у статті Farhadi та інших [29].

Альтернативний підхід являє собою об’єднання підходів комп’ютерного зору та обробки природної мови в єдину, так звану наскрізну (end-to-end) систему, яка на вхід отримує зображення і на вихід одразу видає безпосередньо зрозумілу людям послідовність слів, що описує зображення.

Ця ідея насправді випливає з нещодавніх досягнень в сфері машинного перекладу між мовами. Коротко сучасний процес машинного перекладу можна описати так: рекурентна нейронна мережа (RNN) перетворює, скажімо, французьке речення у векторне представлення, а друга RNN використовує це векторне представлення для створення цільового речення німецькою мовою.

Розглянемо, що буде, якщо ми замінимо першу RNN та її вхідні слова глибокою згортковою нейронною мережею, навченою класифікувати об'єкти на зображеннях. Зазвичай останній шар CNN використовується з застосуванням функції Softmax серед відомих класів об'єктів, що дозволяє визначити ймовірність того, що кожен об'єкт може бути на зображенні.

Але якщо ми видалимо цей останній шар, ми можемо подати «закодовану» за допомогою CNN інформацію про зображення прямо до RNN, розроблену для генерації текстових фраз. При такому налаштуванні ми маємо можливість тренувати всю систему безпосередньо на зображеннях та їх підписах так, що це максимізує ймовірність того, що описи, які система видає на вихід, максимально узгоджені з тренувальними (проанотованими людьми) підписами для кожного зображення. Описана система схематично виглядає так, як наведено на рисунку 2.6.

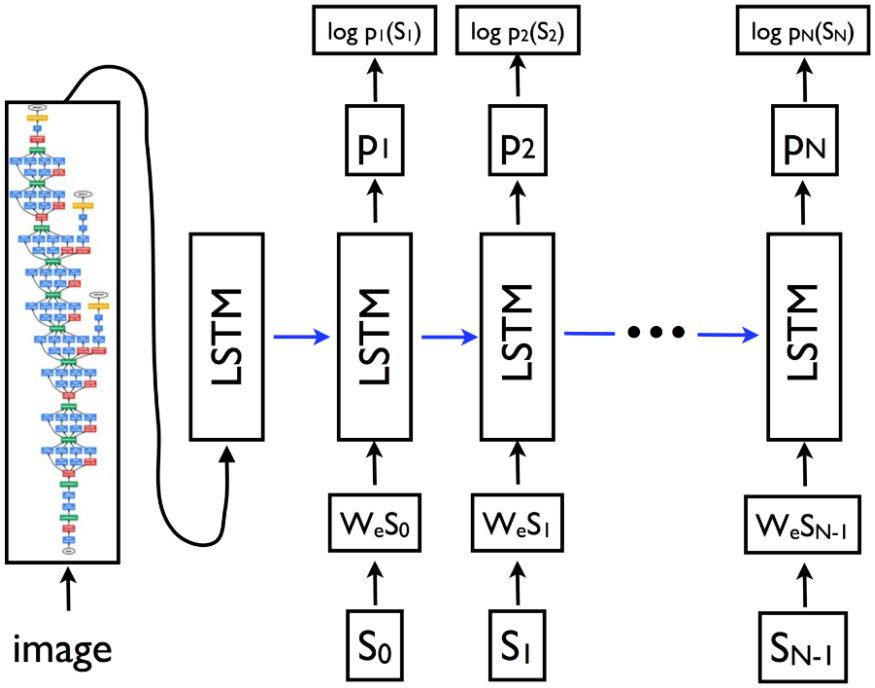

Рисунок 2.6 – CNN-LSTM Архітектура генерації опису зображення

Ця модель продемонструвала дійсно проривні результати: досягнена метрика BLEU-1 на датасеті FLICKR 30K була підвищена на 10, а саме з 56 до 66, у порівнянні з найпотужнішою (state-of-the-art) моделлю до того моменту. На інших наборах даних покращення метрики було не менш вражаючим. Ще більше уваги до моделі привернуло те, що саме вона перемогла на змаганні з генераціїї опісів до зображень на MS COCO у 2015 році. Фінальна версія моделі показала CIDEr, рівний 0.943, та BLEU-4, рівний 0.309.

Підходи, що фігурують у «Show And Tell: A Neural Image Caption Generator», відкрили двері для використання глибинного навчання для вирішення задачі генерації опису зображення й вплинули на велику кількість запропонованих у майбутньому рішень. На поточний момент «Show And Tell» можна вважати базовим рішенням цієї задачі.

Саме тому дана модель глибокого навчання буде використана як базова і в цій роботі.

Детальніше задача оптимізації, що лежить в основі описаної моделі, розглядається в наступному підрозділі.

2.5 Опис задачі оптимізації

Формалізуємо нашу задачу генерації опису зображення у математичному вигляді так, як того вимагає використання моделі глибокого навчання «Show And Tell».

В рамках вирішення задачі необхідно оптимізувати, а саме максимізувати функцію вірогідності формування правильного підпису з використанням наступної формули:

$$\theta^* = \arg max_\theta \sum_{(I,S)} \log p(S|I;\theta) \tag{1.1}$$

де θ – параметри моделі;

I – зображення;

S – підпис.

Вірогідність відносно підпису можна розкласти відносно послідовності слів, використовуючи ланцюгове правило з теорії ймовірності:

$$\log p(S|I;\theta) = \sum_{t=0}^{N} \log p(S_t|I, S_0, \dots, S_{t-1}) \tag{1.2}$$

де I – зображення;

S – цілий підпис;

S_t – останнє слово у підписі;

S_0 – перше слово;

S_{t-1} – передостаннє слово.

Під час навчання моделі $(S|I)$ – це зразок з навчальних даних навчання, і ми оптимізуємо суму логарифмічних ймовірностей, як описано в (2), протягом всього навчального набору, використовуючи стохастичний градієнтний спуск.

Слід зазначити, що для тренування кожне слово представляється закодованим унітарним кодом (one-hot encoding), при цьому довжина вектору для кодування дорівнює розміру нашого словника, тобто кількості усіх слів, що зустрічаються в підписах. При цьому вводиться два додаткових слова: S_0 – спеціальне стартове слово, що позначає початок речення, та S_N – спеціальне кінцеве слово, що позначає завершення речення. Таким чином, при генерації слова S_N нейронна мережа сигналізує про кінець речення.

Загальну процедуру навчання моделі можна представити таким чином:

$$x_{-1} = \mathrm{CNN}(I) \tag{1.3}$$

$$x_t = W_e S_t, \qquad t \in \{0 \dots N-1\} \tag{1.4}$$

$$p_{t+1} = \mathrm{LSTM}(x_t), \qquad t \in \{0 \dots N-1\} \tag{1.5}$$

де I – зображення;

N – кількість слів;

t – номер кроку запуску моделі;

S_t – слово з тренувального підпису або спеціальне слово для t-ого кроку;

x_{-1} – попередні вхідні дані для моделі RNN;

x_t – вхідні дані для RNN на t-ому кроці;

p_{t+1} – кількісні показники ймовірності для обрання наступного, $t+1$ слова.

Зображення I подається моделі тільки один раз, на найпершому кроці. І зображення, і слова, перед подачею до моделі відображаються в один і той же простір, зображення – шляхом використання ботлнек-ознак (передостаннього шару) з CNN, а слова – шляхом векторного представлення (word embedding) W_e.

В якості функції втрат (loss function) використовується сума негативних логарифмічних ймовірностей правильного слова на кожному кроці побудови підпису наступним чином:

$$L(I, S) = -\sum_{t=1}^{N} \log p_t(S_t) \tag{1.6}$$

де I – зображення;

S – цілий підпис;

S_t – правильне слово на кроці t.

На стадії прямого поширення (inference) моделі побудова речення зі слів може відбуватися як з використанням простої техніки, а саме вибору на кожному кроці слова з найбільшою вірогідністю, так і більш складних підходів, наприклад, променевого пошуку (beam search), при якому на кожному кроці t підтримується певна обмежена кількість k найвірогідніших речень.

2.6 Налаштування моделі глибокого навчання

Після завершення конкурсу MS COCO, у 2016 році була опублікована стаття «Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge» [30]. Ця стаття здебільшого повторює контент статті [21], однак з декількома доповненнями щодо додаткових методів для підвищення точності роботи моделі.

Навіть не змінюючи загальної структури моделі, а лише вносячи невеличкі корективи у передобробку вхідних даних, навчання CNN чи RNN, можна отримати помітно кращі підсумкові результати. Пошук та перевірку таких коректив можна вважати додатковим налаштуванням системи.

У статті були описані такі підходи додаткового налаштування, що виявилися ефективними для задачі генерації підпису до зображень на конкурсі MS COCO:

– пакетна нормалізація (batch normalization);
– тонке налаштування системи (fine tuning) та перенос навчання (transfer learning);
– запланована вибірка (scheduled sampling);
– використання ансамблів (ensembling);
– підбір ширини пучка для променевого пошуку (beam search).

Найбільший внесок у роботу моделі на змаганні, як стверджують автори статті, мали впровадження пакетної нормалізації та зміна розміру пучка для променевого пошуку. Кожен з цих підходів допоміг підняти метрику BLEU-4 на 2 одиниці.

Цікавими є результати з beam search. У початковій статті 2014 року [21] було перевірено лише дві ширини пучка: 20, при якому одночасно на кожному кроці підтримуються 20 найкращих речень, та 1, що насправді перетворює променевий пошук на жадібний алгоритм. При цьому ширина 1 давала кращі результати, аніж ширина 20. Для змагання ж було випробувано ще кілька варіантів налаштування, та був обраний той, що був найліпшим відносно метрики CIDER, а саме з пучком розміром 3. Виявилося, що при більшому розмірі пучка відбувалося перенавчання (overfitting) моделі на тренувальних даних. Підтримання малого розміру, в свою чергу, збільшувало новизну генерованих підписів, тобто замість точного повторювання 80% тренувальних підписів, модель повторювала лише 60%. Таким чином, зменшення ширини пучка у даній ситуації можна вважати технікою регуляризації, що призводить до кращої узагальнюючої здібності моделі.

2.7 Нові дослідження

Розглянемо нещодавно написані наукові роботи, у яких запропоновані цікаві ідеї щодо задачі генерації опису до зображень.

Однією з робіт є стаття «Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models» [31]. Вона націлена на вирішення наявної проблеми «Show And Tell» моделі: при використанні променевого пошуку з шириною пучка, що перевищує 1, згенеровані підписи здебільшого дуже схожі один на одний, відмінність часто присутня лише в якомусь єдиному слові.

Автори статті використовують beam search, об’єднуючи схожі підписи в групи та підтримуючи певну різноманітність серед груп, уводячи додаткову метрику відстані між підписами. Цей підхід дозволяє використовувати променевий пошук з більшою шириною пучка без страху, що модель перенавчиться. У статті продемонстровано, як diverse beam search з розміром пучка 20 дозволив досягти дещо кращої роботи моделі. Приклад використання цього підходу наведений на рисунку 2.7.

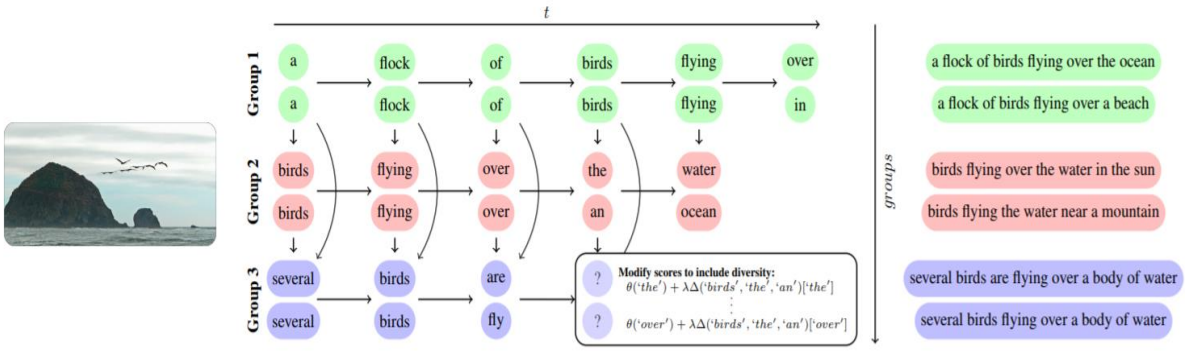

Рисунок 2.7 – Приклад об’єднання підписів у групи при використанні diverse beam search

Робота «Show, Attend and Tell: Neural Image Caption Generation with Visual Attention» [32] розглядає вирішення задачі з використанням механізму «увага» (attention mechanism). Сам по собі механізм «увага» був запропонований наприкінці 2014 році для галузі машинного перекладу [33], однак він чудово зарекомендував себе і для генерації опису до зображень. Великою перевагою механізму «увага» є те, що ми маємо можливість зробити відображення вхідних даних на вихідні. Це дозволяє зрозуміти, на що саме «дивиться» модель для генерації результату, зокрема, які фрагменти зображення грають найбільшу роль для генерації кожного слова при створенні описів до зображення (рис. 2.8).

Важливим недоліком є значне підвищення часових затрат на тренування моделі та складність у паралелізації. Тим не менш, «Show, Attend and Tell» модель дозволила підвищити метрику BLEU-1 (без врахування штрафу за стисліть) з 66.6 до 71.8 та метрику BLEU-4 з 24.6 до 25.0 на датасеті MS COCO у порівнянні з «Show And Tell».

Рисунок 2.8 – Приклади використання механізму «увага» з відображенням відповідності між словом та фрагментом зображення

Для вирішення задачі генерації опису до зображення також було запропоноване використання підходів навчання з підкріпленням (reinforcement learning). Ці підходи оглядаються в статтях «Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning» [34] та « Self-critical Sequence Training for Image Captioning» [35]. Модель з першої статті продемонструвала на MS COCO результати BLEU-1, рівні 74.2, та BLEU-4, рівні 33.2. Підхід з другої статті дозволив отримати єдину модель з показником BLEU-4, рівним 34.2, та ансамбль моделей з BLEU-4, рівним 35.4.

Наукові роботи про генерацію підписів до зображення продовжують публікуватися, і на MS COCO Challenge команди продовжують змагатися у якості своїх моделей. На даний момент перше місце в змаганні посідає модель з кінця березня 2019 року, на якій був досягнений BLEU-1, рівний 81.7, BLEU-4, рівний 40.4, та CIDEr-D, рівний 1.3.

2.8 Формулювання задачі

Метою цієї роботи є створення моделі генерації опису до зображення та додаткове налаштування моделі для покращення якості її роботи.

Серед загальних методів, що можна використовувати для генерації опису, найбільш доцільним буде обрати метод з використанням підходів глибокого навчання. Цей метод упродовж останніх п’яти років демонстрував результати значно кращі за ті, що були досягнуті при методах створення підпису на основі шаблонів чи опису на основі пошуку. Переважна кількість наукових робіт в цій галузі за останній час тим чи іншим чином використовує глибоке навчання.

При виборі моделі глибокого навчання було вирішено узяти за основу «Show And Tell». Ця модель є класичною для розв’язання задачі генерації підписів, і хоча її початкову реалізацію можна вважати дещо слабкою, дивлячись на числові оцінки метрик у порівнянні з сьогоднішніми найкращими результатами, при додатковому налаштуванні модель можна спробувати покращити.

Скоріш за все, після проведених оптимізацій модель все одно не зможе наздогнати результати, отримані більш прогресивними підходами, як, наприклад, навчання з підкріпленням. Однак важливим у даному випадку недоліком reinforcement learning моделей чи attention моделей у порівнянні з класичною моделлю є значно більша обчислювальна складність.

Для навчання та перевірки якості моделі як тренувальний набір даних найкраще буде використовувати MS COCO. MS COCO помітно більший за переважну кількість інших датасетів, містить підписи з різноманітною граматичною структурою та великим словником слів. Це дозволить моделі вивчити багато різних зв'язків та бути більш пристосованою для використання на зображеннях «поза лабораторними умовами», наприклад, в якомусь реальному програмному продукті. При цьому розмір MS COCO не настільки великий, як Conceptual Captions, і це робить його придатним для використання на обладнанні, наявному для даної роботи.

В якості метрики основною обрана CIDEr-D, адже на даний момент вважається, що саме вона може найкраще оцінити якість моделей генерації опису, тому вона і є головною на змаганнях MS COCO Challenge та Google Conceptual Captions Challenge. Однак корисним буде й обчислення BLEU метрик, бо вони є досить інтуїтивно зрозумілими людині й відносно швидко розраховуються.

Для підвищення якості моделі будуть проведені експерименти з додатковим налаштуванням. В рамках досліджень будуть розглянуті підходи, що добре зарекомендували себе у використанні глибинного навчання для інших задач, зокрема на платформі онлайн-змагань з машинного навчання Kaggle [36]. І хоча застосування цих підходів не має значного освітлення в наукових роботах по генерації описів до зображень, можна сподіватися, що вони дадуть гарні результати.

Підсумовуючи цей підрозділ, задачу дослідження можна сформулювати таким чином: розробка моделі генерації опису для зображення на основі моделі глибинного навчання «Show And Tell» з додатковим налаштуванням моделі для підвищення якості її роботи та перевірка якості на датасеті MS COCO з основною метрикою CIDEr-D.

3 ФОРМУВАННЯ МЕТОДИКИ ДОСЛІДЖЕНЬ

3.1 Експерименти з Transfer learning

Один з найрозповсюдженіших методів тренування глибоких нейромереж у галузях комп'ютерного зору та обробки природної мови в наші часи – це так званий transfer learning або перенос навчання.

Перенос навчання – це методика машинного навчання, де модель, що навчається на одному завданні, після цього перенавчається на інше відповідне завдання [37]. Хоча перенос навчання можна застосовувати для великої кількості задач, розглянемо детально його використання для сфери комп'ютерного зору, а саме задачі класифікації.

Припустимо, що перед нами стоїть задача класифікації квітів по зображеннях. Для цієї задачі доцільно використовувати згорткову нейронну мережу. Також припустимо, що в нас є доступ до CNN, вже натренованої на великому наборі даних. Кожна згорткова нейронна мережа складається з двох частин: частини вилучення ознак (зі згортковими шарами й pooling шарами) та власне класифікаційної частини (з повнозгортковим та softmax шарами). При переносі даних ми вилучаємо з раніше претренованої мережі частину вилучення ознак та перевикористовуємо її, «підключаючи» до нашої мережі. Після цього в нашій мережі ми можемо здебільшого навчати тільки класифікаційну частину, щільно пов'язану з цільовим доменом, у даному випадку зображень квітів. Це дозволяє значно зекономити обчислювальні ресурси та час.

В задачах комп'ютерного зору часто для переносу даних використовуються ваги з мережі, претренованої на наборі даних ImageNet [38].

Схема використання переносу навчання для описаної задачі класифікації представлена на рисунку 3.1.

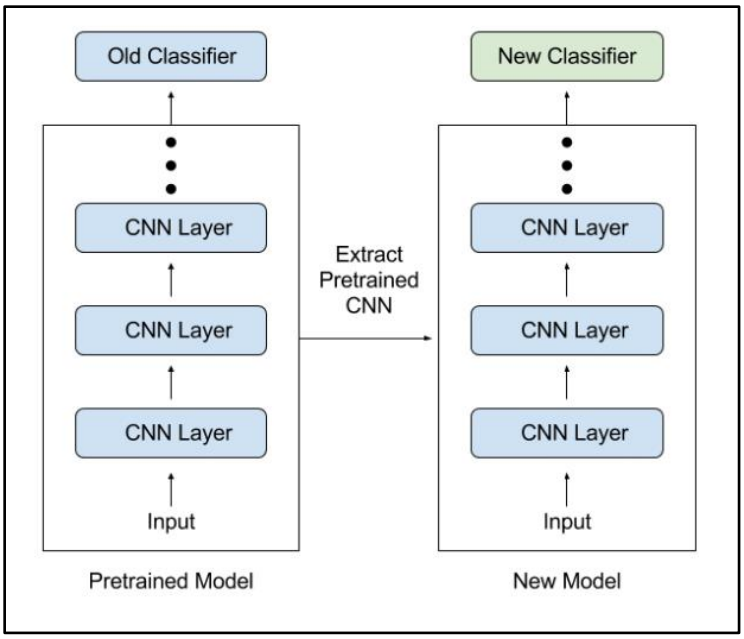

Рисунок 3.1. – Схема використання transfer learning для задачі класифікації

Інший цікавий приклад використання transfer learning – галузь обробки природної мови (NLP). Часто в задачах NLP є необхідність замість слів використовувати їх векторне представлення (word embedding). При цьому підході кожному слову ставиться у відповідність певний багатовимірний вектор дійсних чисел, при чому слова, пов’язані між собою синтаксично чи семантично, у векторному просторі виявляються розташованими поряд. Цей векторний простір також доцільно створювати шляхом тренування на великому корпусі тексту, наприклад Вікіпедії [39], та в подальшому шляхом переносу навчання перевикористовувати для своєї моделі.

В цілому, з появленням відкритих репозиторієв претренованих мереж transfer learning став невід’ємною частиною тренування нейронних мереж у сфері комп’ютерного зору. Один з таких репозиторієв для Python – це pretrained-models.pytorch, і він цілком підходить для проведення експериментів в рамках цієї роботи.

У базовій моделі, що взято для експериментів, вже використовується претренована згорткова нейромережа, тож transfer learning вже проводиться.

У цій роботі пропонується порівняти простий transfer learning, який відбувається шляхом поєднання претренованої згорткової мережі з рекурентною та сумісним навчанням всієї мережі з іншим варіантом: після з’єднання двох частин мережі згорткова частина заморожується на кілька епох, щоб не тренована рекурентна мережа вивчила якісь патерни, а тільки після цього тренується вся конструкція.

3.2 Дослідження застосування різних згорткових архітектур

Давно відомо, що вибір архітектури згорткової частини нейромережі дуже вагомо впливає на фінальну точність моделі у будь-яких задачах комп’ютерного зору.

Історично архітектури згорткових мереж спочатку створюються и тестуються для задачі класифікації зображень, де результати вимірюються на датасеті ImageNet (рис. 3.2).

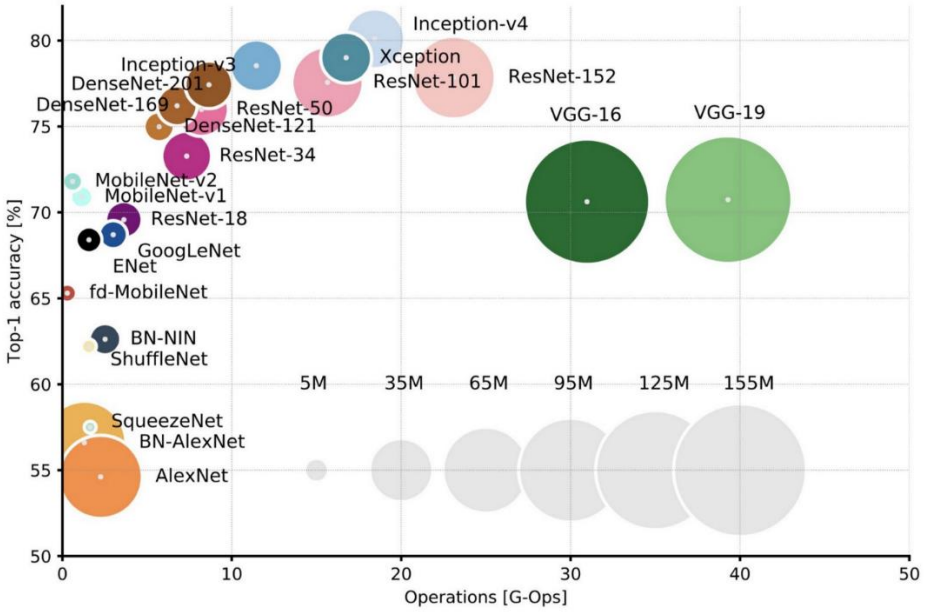

Рисунок 3.2 – Точність різних архітектур на ImageNet

Як можна побачити з діаграми, більш великі та повільні архітектури нейромереж, що мають більше параметрів, часто мають більшу точність класифікації на ImageNet через те, що більша кількість параметрів дозволяє мережам вивчати більш складні патерни, а отже й більш репрезентативний простір ознак.

Така ж сама тенденція може бути продемонстрована на інших задачах, де згорткові архітектури використовуються як складові частини більш складної мережі. Наприклад, для детекції об’єктів більш важкі й точніші архітектури, зображені на діаграмі класифікації зображень, також демонструють більш високі показники точності (рис. 3.3).

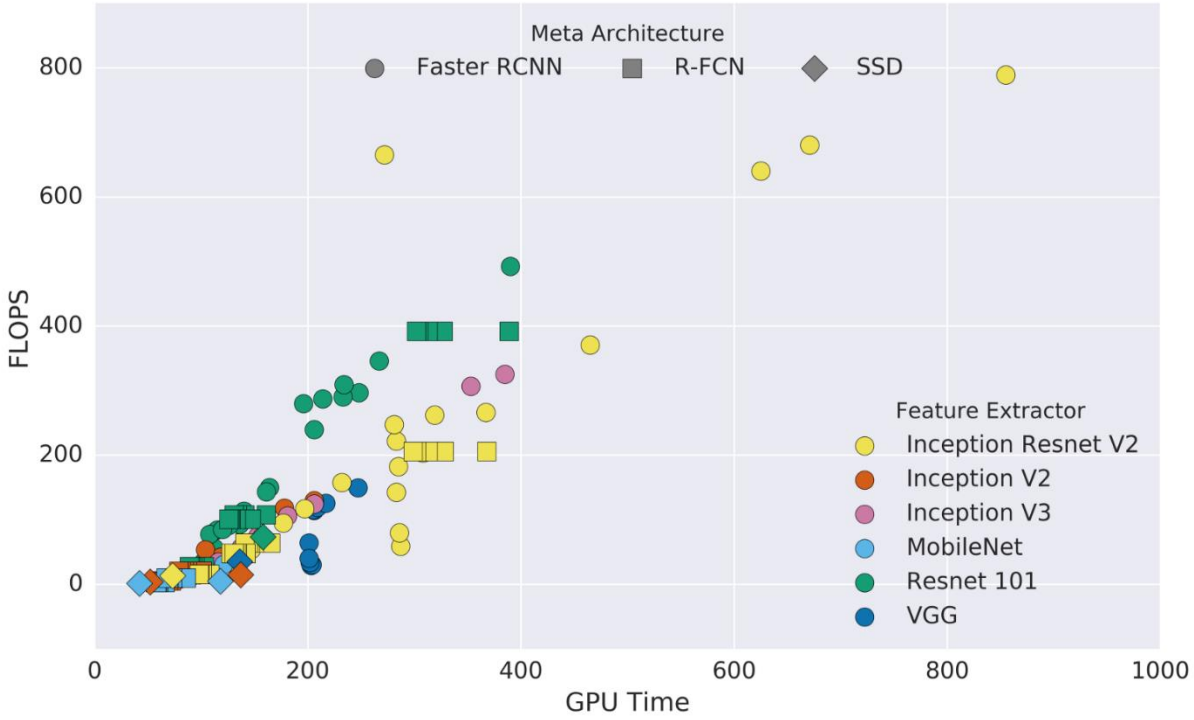

Рисунок 3.3 – Точність різних згорткових архітектур для задач детекції

Щоб перевірити аналогічну гіпотезу для задачі генерації заголовків для зображень, в якості згорткової частини комплексної нейромережі було застовано наступні архітектури:

– InceptionV4 [40] (23.8 мільйони параметрів, 94.9% точності на ImageNet);

– Xception [41] (22.9 мільйони параметрів, 94.5% точності на ImageNet);

– NASNet Large [42] (88.9 мільйонів параметрів, 96.0% точності на ImageNet).

Для кожної архітектури буде протестовано, як саме її використання відображається на результатах навчання повного пайплайну генерації підписів до зображень.

3.3 Дослідження використання аугментацій

Аугментація (augmentation) – це збільшення розмірів набору даних за рахунок застосування природних перетворень до картинок при навчанні мережі. Ідея спирається на те, що людське око вміє досить непогано розрізняти зображення. Якщо ми дивимося на зображення кота, віддзеркалене по вертикальній осі та затемнене, або ж повернене на кілька градусів, ми все ще без проблем впізнаємо кота та зможемо його описати. Так само і наша модель повинна бути в змозі розпізнати зображення після подібного перетворення, а отже, таку аугментацію можна додавати при навчанні мережі.

Аугментувати можна по-різному: застосовувати різні види перетворень, а також регулювати їх рівень жорсткості. Аугментація в деякому сенсі працює схоже на регуляризацію, тому якщо зробити її надто жорсткою, то мережа може почати недонавчатися. Але зате якщо мережа перенавчається, аугментація повинна давати дуже помітний ефект.

Приклад використання відображень та поворотів в якості аугментацій представлений на рисунку 3.4. Як можна побачити, вміле використання аугментації допомагає гарно урізноманітнити наявні дані.

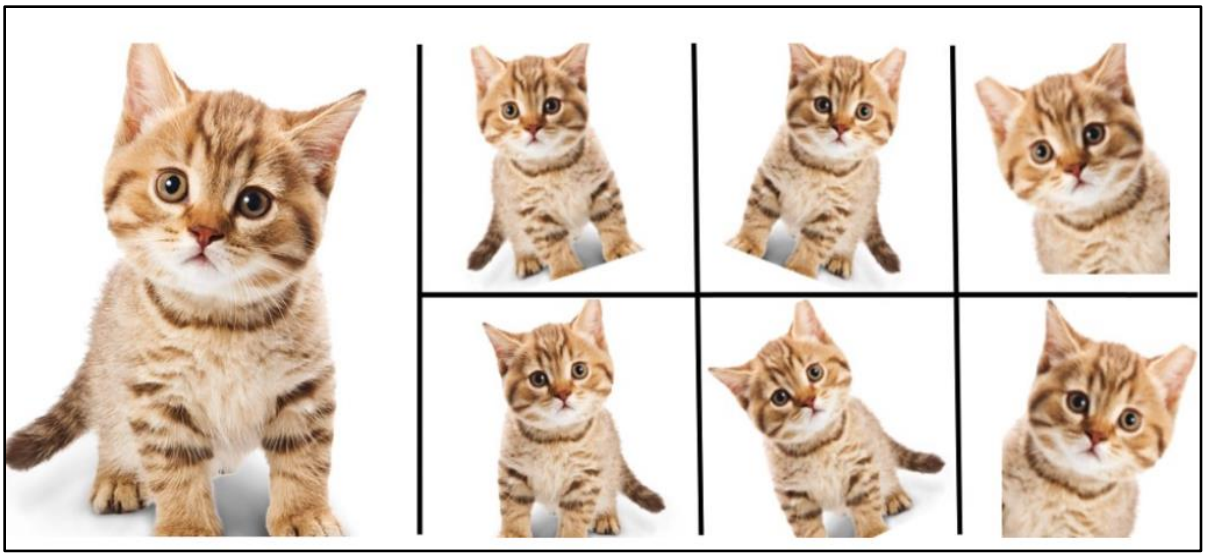

Рисунок 3.4 – Використання аугментацій для зображення з котом

Подібно до того, як аугментації використовуються при тренуванні на навчальному наборі, ми також можемо виконувати випадкові модифікації тестових зображень при валідації та саме тестуванні. Цей підхід носить назву аугментації під час тестування (TTA – test time augmentation). Таким чином, замість того, щоб передавати «чисте» зображення з датасету на треновану модель тільки один раз, ми можемо передавати їй кожне зображення, змінене по кілька разів. Потім же ми можемо усереднити спрогнозовані результати для цих зображень та отримати підсумковий результат для елементу з датасету.

Звичайні аугментації під час тренування моделі відмінно підходять і для задачі генерації опису до зображень. Однак буде доцільним використання не дуже «жорстких» аугментацій, адже, наприклад, у деяких підписах фігурує слово «яскравий» («bright») і при аугментації з затемненням ми можемо втратити цю рису. Також у зв’язку зі специфікою набору даних можуть не підійти аугментації сильних поворотів та горизонтального віддзеркалення, бо слова «зліва» (left) та «справа» (right) також використовуються для певної кількості описів.

Через це пропонується провести три окремих експерименти. У першому експерименті використані наступні аугментації:

– випадкові зміни контрасту (15% ймовірність використання, максимальний рівень зміни – 20%);

– випадкове розмиття зображень (10% ймовірність використання, максимальний рівень розмиття – 20%);

– випадкові зсуви (25% ймовірність використання, максимальний зсув - 15% від розмірів зображення);

– випадкові зміни розміру (25% ймовірність використання, максимальне зменшення або збільшення зображення – 15%);

– випадкові повороти зображення (25% ймовірність використання, максимальний кут повороту – 15 градусів).

Другий експеримент збільшує зміни яркості, кольору та контрасту, перевіряючи, чи впливають кольорові деформації на відповідні прикметники у заголовках, що генеруються:

– випадкові зміни контрасту підвищено з 15% ймовірності до 25%;

– додано випадкові зсуви кольору в просторі HSV (ймовірність використання – 25%, максимальний зсув hue – 15%, інші координати не зсуваються);

– додано випадкові зміни яркості (ймовірність використання 15%, аугментація взаємовиключна зі змінами контрасту, максимальна зміна яркості – 25%).

Третій експеримент перевіряє, чи можуть негативно впливати просторові зміни та віддзеркалення на заголовки, що генеруються:

– випадкові повороти зображення підсилено – максимальні кути повороту підвищено до 25 градусів;

– додано горизонтальне віддзеркалення зображення з ймовірністю 50%.

Для впровадження аугментацій під час тестування, на жаль, знадобилося би більше зусиль, адже текстові підписи не можна просто так усереднити, як, наприклад, числові значення. Тому TTA було вирішено в роботі не використовувати.

3.4 Вибір засобів та технологій

Серед мов програмування, що використовуються для роботи у галузі науки про дані (data science), найбільш популярними є Python та R. Це підтверджує проведене у 2016 році компанією O'Reilly опитування Data Science Salary Survey [43]. Обидві мови надають доступ до величезної кількості інструментів та бібліотек для обробки, аналізу та візуалізації даних. Відрізняє їх те, що R розроблявся саме як мова для статистичного аналізу, а Python являє собою загальну мову програмування, що можна використовувати також і для розробки веб-додатків, настільних додатків, написання невеликих скриптів тощо. Це робить Python більш придатним для роботи у data science, якщо є необхідність не тільки розробляти й перевіряти моделі, а і впроваджувати їх у повноцінні програмні системи.

За результатами останнього опитування сайту StackOverflow мова Python є четвертою за популярністю серед розробників – 41,7% респондентів відповіли, що в тій чи іншій мірі використовують Python, тоді як R з кількістю 5.8% респондентів посідає аж сімнадцяте місце [44]. Саме тому для проведення досліджень в рамках цієї роботи була обрана мова Python.

Також буде більш ніж доцільно використовувати Python-бібліотеку NumPy [45]. Ця бібліотека дозволяє значно спростити реалізацію розповсюджених математичних обчислень за рахунок надання зручної реалізації багатовимірних матриць та масивів, а також підтримує вражаючу за розміром колекцію високорівневих математичних функцій.

NumPy також є надзвичайно швидким у порівнянні з простим Python завдяки інтенсивному використанню розширень, написаних на мові C. Часто використання масивів NumPy замість стандартних колекцій мови є більш вигідним і з точки зору зайнятої оперативної пам’яті. Велика кількість бібліотек Python, пов’язаних зі

статистичним обчисленням, аналізом даних чи машинним навчанням, широко використовує бібліотеку NumPy.

Робота з нейронними мережами теж може бути спрощена шляхом підключення спеціального фреймворку. Популярні фреймворки на Python включають наступні:

– TensorFlow;
– Keras;
– Caffe;
– Theano;
– PyTorch.

У цій роботі в якості фреймворку глибинного навчання обрано PyTorch [46]. PyTorch - це фреймворк, реалізований на C++ з інтерфейсом на мові програмування Python, який надає дві основні високорівневі функції:

а) тензорні обчислення, що нагадують аналогічні обчислення на NumPy, але виконуються з використанням GPU-прискорення;

б) глибокі нейронні мережі, побудовані на основі алгоритму автоматичного диференціювання.

Є кілька причин, чому було обрано саме цей фреймворк:

– його новий гібридний фронтенд дозволяє легко переходити між режимом поточного виконання і графовим режимом, щоб забезпечити гнучкість і швидкість;
– масштабоване розподілене навчання і оптимізації, вбудовані в PyTorch, роблять дослідження з його використанням більш швидкими з точки зору часу розробки;
– глибока інтеграція в Python дозволяє використовувати популярні бібліотеки та пакети для легкого написання нових шарів нейронних мереж в Python, а не в C++, як це потребує, наприклад, Tensorflow;
– велика екосистема інструментів і бібліотек доповнює PyTorch і підтримує його використання у комп'ютерному зорі, обробці природної мови і багатьох інших задачах.

Окрім самого PyTorch також використано дві бібліотеки з його екосистеми – torchvision (включає в себе низку функцій та класів для роботи з візуальними даними, а також нейромережеві архітектури для задач комп'ютерного зору) та pretrained-models.pytorch (репозиторій, що включає в себе групу глибоких згорткових нейромереж, претренованих для задачі класифікації зображень на датасеті ImageNet).

Для реалізації аугментацій використано бібліотеку albumentations, що розробляється відкритою спільнотою розробників (зокрема з України). Ця бібліотека обрана через те, що вона включає в себе реалізовані аугментації всіх типів, що розглядаються в експериментальній частині, а також за результатами публічних бенчмарків працює швидше за конкурентів, що дозволяє розвантажити центральний процесор під час обчислень.

Для обробки мовних даних у роботі використовується фреймворк NLTK – Natural Language Toolkit [47].

NLTK є провідною платформою для створення програм на Python для роботи з задачами обробки природної мови. Вона надає прості у використанні інтерфейси для більш ніж п'ятдесяти корпусів і лексичних ресурсів, таких як WordNet, а також набір бібліотек для обробки тексту, класифікації, маркування, розбору та семантичного аналізу, обгортки для промислових бібліотек і розвинену документацію.

Для візуалізації даних та результатів експериментів будуть використовуватися бібліотеки Matplotlib та Seaborn.

4 ПРОВЕДЕННЯ ДОСЛІДЖЕНЬ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Базова модель

Як вже було зазначено раніше, в якості базової моделі було обрано використовувати модель, описану в статті «Show and tell: A neural image caption generator» [21].

Для тренування моделі використовувалися 82783 зображень, а для валідації – 40504 зображень. Це відповідає розподілу даних MS COCO у змаганні MS COCO Challenge. До кожного зображення присутні 5 підписів. Дані були отримані завдяки MS COCO Python API.

Спочатку був проведений аналіз даних. Для графічної частини даних було встановлено, що наявні зображення мають різні ширину та довжину, однак їх розмір не перевищує 640x640 пікселів. Формат зображень – JPEG.

Для аналізу підписів вони були предоброблені за допомогою NLTK модуля Punct: він «розрізає» текст за пробілами (whitespace characters), формуючи список токенів зі слів та пунктуаційних знаків. Довжина підписів помітно варіюється: найкоротший складається з 6 токенів, а найдовший – з 57 токенів. Середня (mean) кількість токенів на один опис дорівнює 11, а загалом у підписах присутні 25122 різних токенів. Розподіл усіх підписів за кількістю токенів та 20 найбільш популярних токенів з їх частотою зустрічі, відсортовані у порядку спадання, приведені на рисунку 4.1.

З рисунку 4.1 видно, що більшість підписів мають довжину від восьми до близько двадцяти токенів. Також видно, що найрозповсюдженішими токенами є артиклі, крапка з комою та різноманітні прийменники, що є досить очікуваним. Також у топ-20 популярних токенів увійшли слова «man» та «people», що вказує на значну кількість зображень, у яких фігурують люди.

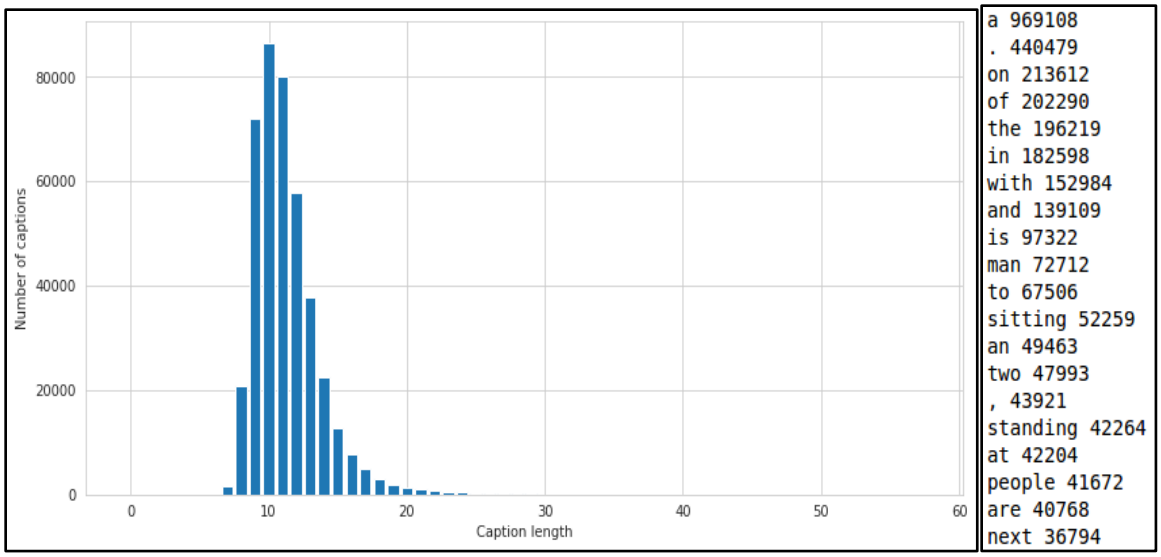

Рисунок 4.1 – Розподіл кількості підписів за їх довжиною та топ-20 найбільш розповсюджених токенів з їх частотою

Є доцільним не використовувати для навчання моделі слова, що зустрічаються зовсім нечасто. Рідкі слова можна вважати менш важливими, до того ж, скоріш за все, модель не зможе вивчити необхідні зв’язки для слів, які вона «зустрічає» лише кілька разів. Зменшення словника слів, у свою чергу, також прискорює навчання моделі.

Було обрано використовувати у словнику лише токени з кількістю входжень 5 або більше разів. Усі слова з меншою частотою були замінені на єдиний токен <unknown>. Усічення слів за їх частотою призвело до зменшення розміру словника з з 25122 до унікальних 9956 токенів.

Для більш ефективних та швидких розрахунків і робота CNN, і робота LSTM була перенесена на GPU з використанням CUDA. CUDA – це платформа для паралельних обчислень на GPU Nvidia зі зручним API. На щастя, PyTorch реалізує корисні обгортки для виклику CUDA функцій, що перетворює роботу з GPU на ще більш просту та гнучку. Обчислення за допомогою CUDA на GPU прискорило навчання моделі в порівнянні з обчисленням на CPU щонайменше у п’ять разів.

В якості функції втрат (loss function) використовувалася крос-ентропія. Також згідно з [32], найкращим оптимізатором для налаштування стохастичного градієнтного спуску мережі для датасету MS COCO виступає запропонований у 2014 році оптимізатор Адам (Adam optimizer) [48], тож він і був обраний для даної моделі. У таблиці 4.1 представлені гіперпараметри, що використовувалися для налаштування моделі:

Таблиця 4.1 – Перелік гіперпараметрів для налаштування моделі

Назва гіперпараметру	Значення гіперпараметру
Темп навчання (learning rate)	0.0005
Розмір батча (batch size)	32
Розмір прихованого шару RNN	512
Розмірність векторного простору (embedding) слів та зображень	256
Максимальна допустима довжина згенерованого підпису	20
Ширина пучка променевого пошуку	3
Максимальна кількість епох	50

При тренуванні була встановлена верхня межа для кількості епох, рівна п’ятдесяти, однак додатково використовувалася техніка раннього припинення (early stopping). Ця техніка полягає в тому, що при відсутності покращення значення функції втрат упродовж певної кількості епох поспіль (у даному випадку це було п’ять епох) модель припиняла своє навчання. Цільовою метрикою була CIDEr-D.

Для вибору згорткової частини моделі було вирішено одразу перевірити декілька архітектур CNN. Перед використанням у моделі слова були закодовані як послідовні числа (кожному унікальному слову відповідає унікальне число), а зображення змінені за розміром до 299x299 пікселів та переформатовані у RGB, адже згорткові нейронні мережі InceptionV4 і Xception приймають на вхід тензор розміром 299x299x3, а NASNet Large – 331x331x3. Зміна розміру відбувалася простим ущільненням зображення до потрібного розміру, тобто пропорції зображених об’єктів могли змінюватися. Результати роботи моделей та кількість епох тренування представлені у таблиці 4.2.

Таблиця 4.2 – Результати роботи моделей з різними згортковими архітектурами та кількість епох тренування

Архітектура CNN	CIDEr-D	BLEU-1	BLEU-4	Кількість епох
InceptionV4	0.883	70.5	28.9	44
Xception	0.861	69.9	27.8	41
NASNet Large	0.885	70.7	29.2	30 (тренування припинене)

Як і було теоретично описано у попередньому розділі, згорткові архітектури з великою кількістю шарів та параметрів демонструють на великому наборі даних більш високі значення метрик. Однак, NASNet Large займає майже в три рази більше часу на тренування у порівнянні з InceptionV4, тому тренування NASNet Large було припинене на 20 епохі. Зваживши отримані результати, для подальших експериментів у якості базової моделі взято архітектуру, де як згорткова частина виступатиме саме InceptionV4.

Валідація моделей відбувалася після закінчення кожної епохи для можливості побудови графіку кривої навчання (learning curve). Крива навчання дозволяє отримати базову інформацію про узагальнюючу здібність моделі, про зміну якості моделі упродовж епох та про те, чи наявне у моделі перенавчання (overfitting) чи недонавчання (underfitting). Крива навчання для моделі зі згортковою архітектурою InceptionV4 відносно функції втрат продемонстрована на рисунку 4.2. Для побудови цього графіку одразу після закінчення кожної епохи тренування запускалася стадія валідації. На діаграмі також позначена точка, в якій досягається найменше значення функції втрат під час валідації.

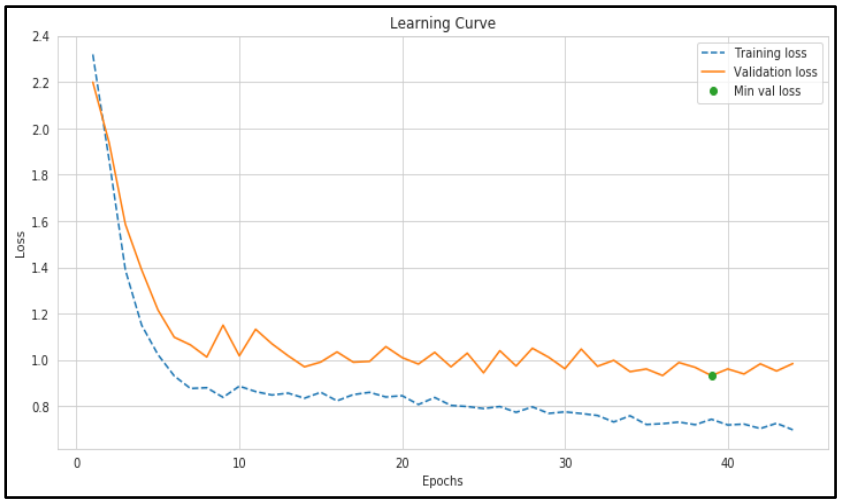

Рисунок 4.2 – Крива навчання для моделі з InceptionV4

З кривої навчання видно, що функції втрат для тренування та валідації мають між собою проміжок, однак не дуже великий. Loss під час валідації осцилює в більшій мірі й з певного моменту він припиняє покращуватися, а росте вгору. Це свідчить про те, що модель починає перенавчатися. Аби запобігти перенавчанню, можна загалом додавати ще даних, впроваджувати регуляризацію чи припиняти навчання моделі. У даному випадку якраз допомагає early stopping, і ми перериваємо роботу моделі після відсутності покращення метрики протягом п’яти епох.

Приклад роботи базової моделі з використанням InceptionV4 на даних, що являють собою зображення пам’ятника в Харкові, наведений на рисунку 4.3.

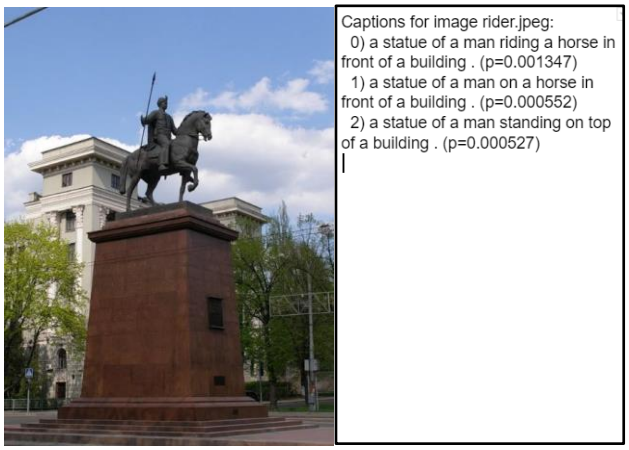

Рисунок 4.3 – Результат роботи базової моделі на фотографії пам’ятника у Харкові

Таким чином, для базової моделі був отриманий CIDEr-D, рівний 0.883, та BLEU-4, рівний 28.9. Це вже перевищує результати, опубліковані авторами у оригінальній статті [21]: там вони досягли на локальній валідації BLEU-4, рівний 27.7, та CIDEr-D, рівний 0.819. На жаль, про значення метрики BLEU-1 інформація від авторів статті відсутня.

4.2 Основні експерименти

Тепер, використовуючи обрану базову модель, ми можемо провести основні дослідження. Першим йде експеримент з режимом розморозки згорткової частини при використанні transfer learning. Для варіанту з відстроченою розморозкою перші 10

епох було обрано тренувати лише рекурентну частину, а після цього тренувати вже всю модель цілком. Результати представлені у таблиці 4.3.

Таблиця 4.3 – Порівняння роботи моделей з використанням різних режимів розморозки мережі після переносу навчання

Назва моделі	Cider-D	BLEU-1	BLEU-4	Кількість епох
InceptionV4 з класичним переносом навчання	0.883	70.5	28.9	44
InceptionV4 з відстроченою розморозкою зготкової частини при переносі навчання	0.887	70.9	29.2	43

Як ми бачимо, використання InceptionV4 з відстроченою розморозкою при переносі навчання дало дещо кращі результати, аніж InceptionV4 з класичним переносом навчання. Однак з відстроченою розморозкою зготкової частини продемонструвала важливу перевагу: час навчання скоротився майже в півтора рази. Ця економія часу досягається як раз на перших епохах. В подальших експериментах саме ця модель буде порівнюватися з модифікованими моделями.

Наступним експериментом було порівняно вплив аугментацій на якість навчання моделей (табл. 4.4). Як можна побачити, загалом використання аугментацій вплинуло на якість моделі позитивно. Це можна пояснити тим, що наша базова модель була схильна до перенавчання, а аугментації дозволили створити видимість більшої кількості наявних даних. На жаль, аугментації з жорсткими змінами відносно кольору, а саме сильними змінами контрасту та яркості, а також зсувами кольору в просторі HSV, призвели до невеликої деградації якості моделі. Це можна пояснити тим, що у

підписах присутня певна кількість прикметників, пов’язаних з кольором, наприклад: white (зустрічається 35874 рази), red (17185 разів), blue (13618 разів). Скоріш за все, при сильних змінах кольорової складової на зображеннях, модель з аугментаціями робить помилки саме у цих словах.

Таблиця 4.4 – Результат роботи моделей з різними режимами аугментації

Назва моделі	Cider-D	BLEU-1	BLEU-4	Кількість епох
InceptionV4	0.887	70.9	29.2	43
InceptionV4 з м’якими аугментаціями	0.904	71.6	30.6	46
InceptionV4 зі збільшеними аугментаціями кольору	0.883	70.7	29.1	46
InceptionV4 зі збільшеними просторовими аугментаціями	0.917	71.9	30.8	47

Тим не менш, використання моделі з м’якими аугментацями, а також моделі з м’якими аугментаціями щодо кольору та жорсткими щодо просторової складової, призвело до дуже помітного поліпшення якості.

Таким чином, після проведених експериментів ми можемо виділити модель, що краще за все проявила себе під час валідації. Ця модель має в якості згорткової частини InceptionV4, при переносі навчання перші 10 епох її згорткова частина заморожена та не навчається, а також під час навчання вона отримує зображення з м’якою аугментацією по кольору та більш жорсткою аугментацією по просторових характеристиках.

4.3 Перевірка на власних даних та аналіз результатів

Для додаткової перевірки якості моделі був відібраний невеликий датасет із зображеннями, тема яких охоплює Україну, Харків чи ХНУРЕ. Набір даних складається з 20 елементів. Цей датасет був створений, щоб мануально протестувати роботу моделі на зображеннях, яких точно не буде ні в тренувальній, ні у валідаційній, ні у тестовій виборці MS COCO, при цьому зображення пов’язані з українськими реаліями. Зображення мають різний розмір та деталізацію.

Для кожного зображення через модель було згенеровано три підписи. Після цього підписи були переглянуті на відповідність моделі та оцінені за шкалою, приведеною у таблиці 4.5.

Таблиця 4.5 – Шкала для мануальної перевірки підписів

Оцінка	Критерій
5	Перший з підписів точно описує зображення.
4	Серед підписів присутній той, що описує зображення, однак він розташований не першим.
3	Серед підписів присутній той, що добре описує зображення, однак у підписі присутні помилки чи неточності.
2	У підписах згадуються певні об’єкти чи риси, характерні для зображення, однак у цілому підпис мало пов’язаний із зображенням.
1	Підписи не відповідають зображенню

У додатку А наведено дані про використані зображення, підписи, що були згенеровані моделлю, а також виставлені оцінки. Середня оцінка становить 3.45, що можна вважати гарним результатом, адже це означає, що в більшості випадків серед

підписів або присутній правильний підпис, або присутній підпис, що містить лише деякі неточності. Модель добре справляється з зображеннями різного розміру, у тому числі і якщо розмір менший за тензор, що в результаті піде на вхід до згорткової частини (тобто менший за 299x299 пікселів).

При аналізі роботи були зроблені наступні спостереження:

– словник термів, що використовувався для моделі та складається з близько 10 тисяч елементів, дозволяє покривати досить обширну кількість тематик;

– модель не завжди генерує в кінці речення точку. Іноді два підписи вважаються різними тільки через те, що в одному з них присутня точка, а в іншому – ні. Однак це ніяк не впливає на семантичний зміст підпису, і якщо потрібно, ці випадки легко відстежуються постобробкою описів;

– іноді модель генерує «дивні» підписи, у яких псується зміст через наявність зайвих слів. Прикладом може бути зображення крупного плану кішки, для якого модель буде генерувати підпис, схожий на такий: «Тут присутня біла кішка та біла кішка»; тоді як правильним був би простий підпис «Тут присутня біла кішка». Це пояснюється тим, що значна кількість підписів з MS COCO має довжину не менше десяти термів, тому модель певним чином вивчила розподіл довжин та обирає ту довжину, вірогідність якої більша. До того ж в рамках метрик часто враховується «штраф» за генерацію коротких підписів;

– модель добре розпізнає на зображеннях людей, що обумовлено великою кількістю даних, пов’язаних з людьми у тренувальній виборці. Однак предмети, з якими взаємодіють люди, не завжди розпізнаються правильно: наприклад, гітара може були переплутана з тенісною ракеткою;

– зображення з багатьма деталями, наприклад, знімки міста з висоти, розпізнаються не добре, скоріш за все через присутність занадто дрібних об’єктів.

Отримана модель вже може використовуватися як компонент прикладної програми, сумісної з Python, або ж як еталонна модель для проведення подальших наукових експериментів.

ВИСНОВКИ

В рамках цієї магістерської роботи було проведено аналіз існуючих методів генерації підписів до зображень. Були виявлені основні тенденції в даній області та переваги нових підходів та ідей щодо генерації над старими.

Було оглянуто відповідну наукову літературу за темою, проаналізовано ступінь вирішення задачі генерації опису зображень та оглянуті досягнені числові показники якості рішень. Було вивчено, як саме задача генерації опису зображень пов’язана з іншими задачами комп’ютерного зору та обробки природної мови. При розробці власної моделі в якості основи була обрана модель глибокого навчання «Show and tell», адже вона демонструє значно кращі результати, аніж рішення, запропоновані до 2014 року, і при цьому є не надто масивною, що робить її придатною для додаткового налаштування.

В рамках додаткового налаштування були порівняні між собою різні архітектури згорткових нейронних мереж, зокрема InceptionV4, Xception та NASNet Large. Виявилося, що найкращі результати показала NASNet Large, однак час її тренування майже в три рази перевищував час для InceptionV4 та Xception. Тому для подальших експериментів була обрана CNN InceptionV4, що показала другий результат на наборі даних MS COCO після NASNet Large. Допомогли підвищити якість моделі також «заморозка» згорткової частини на кілька епох після переносу навчання та використання аугментацій. Аугментації зробили дійсно значущий внесок у покращення моделі, однак дуже жорстке використання змін кольору, навпаки, призводило до погіршення точності.

Фінальна модель показала результат метрики BLEU-4, вищий за метрику оригінальної моделі «Show and tell» на 3.1 (30.8 проти 27.7), та метрику CIDEr-D, вищу на 0.098 (0.917 проти 0.819). Досягнена у роботі метрика BLEU-1 складає 71.9, і означає приблизно те, що 71.9 % слів зі згенерованих підписів присутні у описах,

сформованих людьми, що є досить непоганим результатом. Також було проаналізовано, в яких випадках модель генерує підписи краще, а в яких – гірше, та розглянуті потенційні причини цієї поведінки.

Результуюча модель може використовуватися і для проведення подальших експериментів та налаштування, і для надання бажаної функціональності в рамках прикладної програми, сумісної з Python.

Для подальшого підвищення якості моделі, безумовно, слід спробувати різні підходи до управління темпом навчання (learning rate). Гарними практиками є використання темпу навчання з постійним згасанням (decay) чи зменшення темпу, коли крива навчання виходить на плато. Обидва підходи здатні помітно покращити результат, адже вони поліпшують сходимість стохастичного градієнтного спуску та допомагають знайти оптимум.

Ще одним шляхом для потенційного розвитку наукової роботи є використання ідей з нещодавніх досліджень, наприклад механізму «увага». Цей механізм є надзвичайно цікавим, адже з ним модель перестає бути повністю «чорним ящиком». Однак при впровадженні механізму «увага» треба бути готовим до більших накладних витрат на навчання мережі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ImageNet: the data that spawned the current AI boom — Quartz [Електронний ресурс]. URL: https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/ (дата звернення: 17.06.2019).

2. Goodfellow I., Bengio Y., Courville A. Deep Learning. : The MIT Press, 2016. 800 с.

3. Rashtchian C. та ін. Collecting Image Annotations Using Amazon's Mechanical Turk. 2010. С. 139–147.

4. Hodosh M., Young P., Hockenmaier J. Framing Image Description as a Ranking Task Data, Models and Evaluation Metrics Extended Abstract С. 5.

5. Young P. та ін. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions // Transactions of the Association for Computational Linguistics. 2014. Т. 2. С. 67–78.

6. Lin T.-Y. та ін. Microsoft COCO: Common Objects in Context // Computer Vision – ECCV 2014 Lecture Notes in Computer Science. / под ред. D. Fleet та ін. : Springer International Publishing, 2014. С. 740–755.

7. Sharma P. та ін. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. 2018. С. 2556–2565.

8. Papineni K. та ін. BLEU: a Method for Automatic Evaluation of Machine Translation // 2002.

9. Lin C.-Y. ROUGE: A Package for Automatic Evaluation of Summaries. 2004. С. 74–81.

10. Denkowski M., Lavie A. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. 2014. С. 376–380.

11. Vedantam R., Lawrence Zitnick C., Parikh D. CIDEr: Consensus-based Image Description Evaluation // 2014.

12. Anderson P. та ін. SPICE: Semantic Propositional Image Caption Evaluation. 2016. С. 382–398.

13. Lin C.-Y., Och F.J. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-bigram Statistics // Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics ACL '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004.

14. Chen X. та ін. Microsoft COCO Captions: Data Collection and Evaluation Server // 2015.

15. Robertson S. Understanding inverse document frequency: on theoretical arguments for IDF // Journal of Documentation. 2004. Т. 60. № 5. С. 503–520.

16. Image Processing with the Computer Vision API | Microsoft Azure [Електронний ресурс]. URL: https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/ (дата звернення: 21.05.2019).

17. CaptionBot - For pictures worth the thousand words [Електронний ресурс]. URL: https://www.captionbot.ai/ (дата звернення: 21.05.2019).

18. Image Caption Generator – IBM Developer [Електронний ресурс]. URL: https://developer.ibm.com/exchanges/models/all/max-image-caption-generator/ (дата звернення: 21.05.2019).

19. Pricing - Cognitive Services All-In-One | Microsoft Azure [Електронний ресурс]. URL: https://azure.microsoft.com/en-in/pricing/details/cognitive-services/ (дата звернення: 21.05.2019).

20. Tran K. та ін. Rich Image Captioning in the Wild // arXiv:1603.09016 [cs]. 2016.

21. Vinyals O. та ін. Show and tell: A neural image caption generator // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. С. 3156–3164.

22. Hossain M.Z. та ін. A Comprehensive Survey of Deep Learning for Image Captioning // arXiv:1810.04020 [cs, stat]. 2018.

23. Li S. та ін. Composing Simple Image Descriptions using Web-scale N-grams C. 9.

24. Bai S., An S. A Survey on Automatic Image Caption Generation // Neurocomputing. 2018. T. 311.

25. LeCun Y. та ін. Backpropagation Applied to Handwritten Zip Code Recognition // Neural Comput. 1989. T. 1. № 4. C. 541–551.

26. LeCun Y., Bengio Y., Hinton G. Deep learning // Nature. 2015. T. 521. № 7553. C. 436–444.

27. Hochreiter S., Schmidhuber J. Long Short-term Memory // Neural computation. 1997. T. 9. C. 1735–80.

28. Karpathy A., Fei-Fei L. Deep Visual-Semantic Alignments for Generating Image Descriptions C. 17.

29. Farhadi A. та ін. Every Picture Tells a Story: Generating Sentences from Images // Computer Vision – ECCV 2010. : Springer, Berlin, Heidelberg, 2010. C. 15–29.

30. Vinyals O. та ін. Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge // IEEE Trans. Pattern Anal. Mach. Intell. 2017. T. 39. № 4. C. 652–663.

31. Vijayakumar A.K. та ін. Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models // arXiv:1610.02424 [cs]. 2016.

32. Xu K. та ін. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention // arXiv:1502.03044 [cs]. 2015.

33. Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate // arXiv:1409.0473 [cs, stat]. 2014.

34. Lu J. та ін. Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning // arXiv:1612.01887 [cs]. 2016.

35. Rennie S.J. та ін. Self-critical Sequence Training for Image Captioning // arXiv:1612.00563 [cs]. 2016.

36. Kaggle: Your Home for Data Science [Електронний ресурс]. URL: https://www.kaggle.com/ (дата звернення: 21.05.2019).

37. Torrey L., Shavlik J. Transfer learning // Handbook of Research on Machine Learning Applications. 2009.

38. Deng J. та ін. ImageNet: A large-scale hierarchical image database // 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009. С. 248–255.

39. Pretrained Embeddings - Wikipedia2Vec [Електронний ресурс]. URL: https://wikipedia2vec.github.io/wikipedia2vec/pretrained/ (дата звернення: 21.05.2019).

40. Szegedy C. та ін. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning // arXiv:1602.07261 [cs]. 2016.

41. Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions // arXiv:1610.02357 [cs]. 2016.

42. Zoph B. та ін. Learning Transferable Architectures for Scalable Image Recognition // arXiv:1707.07012 [cs, stat]. 2017.

43. 2016 Data Science Salary Survey [Book] [Електронний ресурс]. URL: https://www.oreilly.com/library/view/2016-data-science/9781492049029/ (дата звернення: 20.06.2019).

44. Stack Overflow Developer Survey 2019 [Електронний ресурс]. URL: https://insights.stackoverflow.com/survey/2019 (дата звернення: 20.06.2019).

45. Travis E. Oliphant. Guide to Numpy. 2006. 371 с.

46. Paszke A. та ін. Automatic differentiation in PyTorch С. 4.

47. Loper E., Bird S. NLTK: The Natural Language Toolkit // Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1 ETMTNLP ’02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. С. 63–70.

48. 1. Kingma D.P., Ba J. Adam: A Method for Stochastic Optimization // arXiv:1412.6980 [cs]. 2014.