Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

«Анотація зображень з використанням згорткових та рекурентних нейронних мереж»

Текстова частина до курсової роботи за спеціальністю «Комп'ютерні науки та інформаційні технології» 122

Керівник курсової роботи
старший викладач факультету інформатики
кафедри інформатики Бучко О. А.

_____

(підпис)

"____" _____ 2020 р.

Виконала студентка 4-го курсу
факультету інформатики,
спеціальність комп'ютерні науки
та інформаційні технології
Завертайло М. О.

"____" _____ 2020 р.

Київ – 2020

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

зав.кафедри інформатики, кадидат фізико-математичних наук, доцент

_____С. С. Гороховський

(підпис)

«____»_____ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Завертайло М. О. факультету інформатики 4-го курсу

ТЕМА Анотація зображень з використанням згорткових та рекурентних нейронних мереж

Вихідні дані: алгоритм для анотації зображень українською мовою з можливістю озвучування

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Календарний план

Вступ

1 Основні відомості про рекурентні нейронні мережі та обробку природної мови

2 Анотація зображень. Проблема та особливості вирішення

Дата видачі «__» _____ 2020 р.

Керівник _____ (підпис)

Завдання отримав _____ (підпис)

Тема: Анотація зображень з використанням згорткових та рекурентних нейронних мереж

Календарний план виконання роботи:

№ п/п	Назва етапу курсового проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу	15.10.2019	
2.	Огляд технічної літератури за темою роботи	15.12.2019	
3.	Аналіз сучасних методів пошуку та розпізнавання об’єктів на зображенні	20.12.2019	
4.	Аналіз сучасних алгоритмів обробки природної мови (генерація розумного тексту)	31.12.2019	
5.	Вибір середовища розробки, ознайомлення з необхідними бібліотеками	07.12.2020	
6.	Написання практичної частини курсової роботи	01.04.2020	
7.	Написання текстової частини курсової роботи	15.04.2020	
8.	Написання висновків курсової роботи	16.04.2020	
9.	Надання роботи керівнику для перевірки	17.04.2020	
10.	Корегування роботи за результатами перевірки	19.04.2020	
11.	Оформлення слайдів презентації курсової роботи, підготовка доповіді	19.04.2020	
12.	Подання роботи на кафедру для перевірки на плагіат	19.04.2020	
13.	Захист курсової роботи	кінець квітня	

Студент Завертайло М. О.

Керівник Бучко О. А.

“_____”_____

ВСТУП

Щодня нас оточують мільйони зображень в Інтернеті і попри те, що вони не мають словесного опису, людський мозок не задумуючись розпізнає різні об’єкти на фото, класифікує їх та формує загальне розуміння того, що знаходиться на фото. На жаль, комп’ютеру це зробити не так просто і для цього потрібно використовувати нестандартні підходи машинного навчання.

Актуальність. Використання та розвиток алгоритмів анотації зображень вирішить багато проблем, з якими люди зустрічаються в повсякденному житті, а також дозволить зробити значний прогрес у багатьох сферах: оптимізація пошуку зображень, створення охоронних систем, безпілотних автомобілів, а також мобільних застосунків для людей з вадами зору.

Мета роботи – структурувати знання про підхід до вирішення задачі анатоції зображення, детально розглянути та описати нейронну мережу Inception-V3, дослідити архітектуру рекурентних нейронних мереж «Довга короткочасна пам’ять» (англ. LSTM – long short-term memory), а також реалізувати власний алгоритм анотації зображення.

Об’єктом дослідження методи автоматичного анотування зображень.

Предметом дослідження є алгоритм анотації зображення та використання в ньому рекурентної нейронної мережі ДКЧП.

Новизна роботи полягає в розробці власної моделі для вирішення завдання анотації зображення українською мовою з можливість озвучки згенерованого речння.

Структура роботи. У першому розділі детально розглянуто основні поняття, що стосуються рекурентних нейронних мереж та їх використання в обробці природної мови (англ. NLP). У другому розділі описано особливості постановки задачі анотації зображення. У третьому розділі описано власну реалізацію вирішення цієї задачі: технології, що були використані, архітектура моделі, отримані результати.

Зміст

РОЗДІЛ 1. Основні відомості про рекурентні нейронні мережі та обробку природної мови

1.1. Рекурентна нейронна мережа

Рекурентні нейронні мережі – це один з видів нейронних мереж, де з’єднання між вузлами формують орієнтований у часі граф. Це надає можливість мережі мати динамічну поведінку та змінювати свій внутрішній стан. На відміну від класичних згорткових нейронних мереж, які приймають на вхід вектор певного розміру та повертають вектор визначеного фіксованого розміру (наприклад, ймовірності належності вхідного зображення до різних класів), рекурентна нейронна мережа дозволяє працювати над послідовностями векторів. Наразі рекурентні нейронні мережі розв’язують більшість задач, пов’язаних з обробкою природної мови, а саме: машинний переклад, моделювання мови, генерація опису зображення, розпізнання мови тощо.

Найбільш розповсюджені типи рекурентних нейронних мереж це: один до одного (фіксований вектор на вході та на виході мережі; стандартний підхід, що використовується в задачах класифікації), один до багатьох (даний тип найбільш використовується в задачі анотації зображень, де на вході є вектор зображення, на виході – послідовність векторів, тобто слів у реченні), багато до одного (використовується в семантичному аналізі речень, де на вході вектор слів, а на виході експресивне забарвлення: позитивне чи негативне), багато до багатьох (даний вид використовується в задачі машинного перекладу, де на вході набір слів однією мовою, а на виході – іншою). Зрозуміло, що робота з послідовностями векторів дозволяє зробити нейронну мережу більш гнучкою та потужною [2].

В основі обчислень усіх рекурентних мереж закладений простий принцип: вони приймають вхідний вектор x та видають вихідний вектор y, але на обчислення значення вектора y впливає не лише вхідний вектор, але вся історія векторів, які подавались на вхід раніше. Тобто мережа має деякий внутрішній стан і оновлює його.

Розглянемо схему роботи рекурентної нейронної мережі та її розгортку.

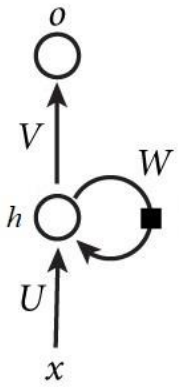


Рисунок 1.1.1 – Схема рекурентної нейронної мережі

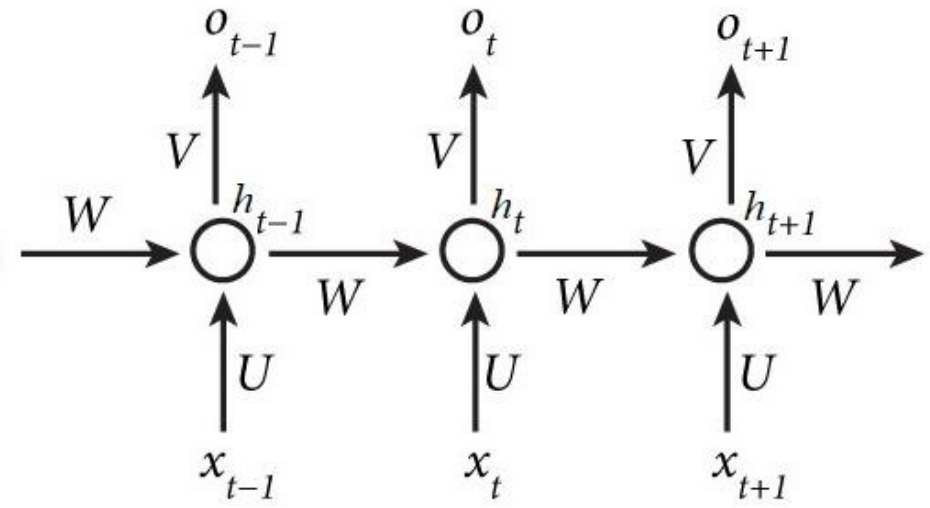


Рисунок 1.1.2 – Розгортка рекурентної нейронної мережі

x_t – вхід на кроці t,

h_t – прихований стан на кроці t, спочатку ініціалізується нульовим вектором.

Оновлення прихованого стану відбувається за формулою:

$$h_t = \tanh(Ux_t + Wh_{t-1}).$$

h_t – це так звана пам’ять нейронної мережі, оскільки на її розрахунок впливає попередній стан, то це дозволяє нам брати до уваги попередньо обчислені результати. Недоліком є те, що прихований стан не враховує інформацію з шарів, що були обраховані дуже давно.

o_t – вихідний вектор на кроці t. Зазвичай розраховується як $softmax(V_{h_t})$.

Функція tanh() забезпечує нелінійність та знаходження стану у діапазоні [-1, 1]. Матриці, що використовуються в обчисленнях, ініціалізуються випадковими числами і метою навчання мережі є пошук необхідних матриць (параметрів мережі). Для того, щоб оцінити наскільки правильним є вихідний вектор y в залежності від вхідного вектора x використовується функція втрат.

Розглянемо приклад передбачення наступної літери в слові, зображеного на рисунку 1.1.2. Нехай словник складається з літер «с», «у», «д», «я». Завдання рекурентної нейронної мережі – передбачити наступну літеру, маючи початкову літеру. Вхідний шар ініціалізується унітарним вектором розміру Nx1, де N – кількість літер у словнику. Для обрахунку значень першого прихованого шару h_1 необхідно:

1) Знайти добуток $A = W_{3X3} * h_0$, де h_0 – нульовий вектор, W – підібрані випадковим чином параметри.

2) Знайти добуток $B = U_{3X4} * x_0$.
3) Значення прихованого шару $h_t = \tanh(A + B)$.
4) Значення вихідного шару $o_t = softmax(V * h_t)$.

Літера, що обирається на останньому кроці («у»), передається на вхід у нейронну мережу і формування слова триває, поки не буде виданий символ завершення слова.

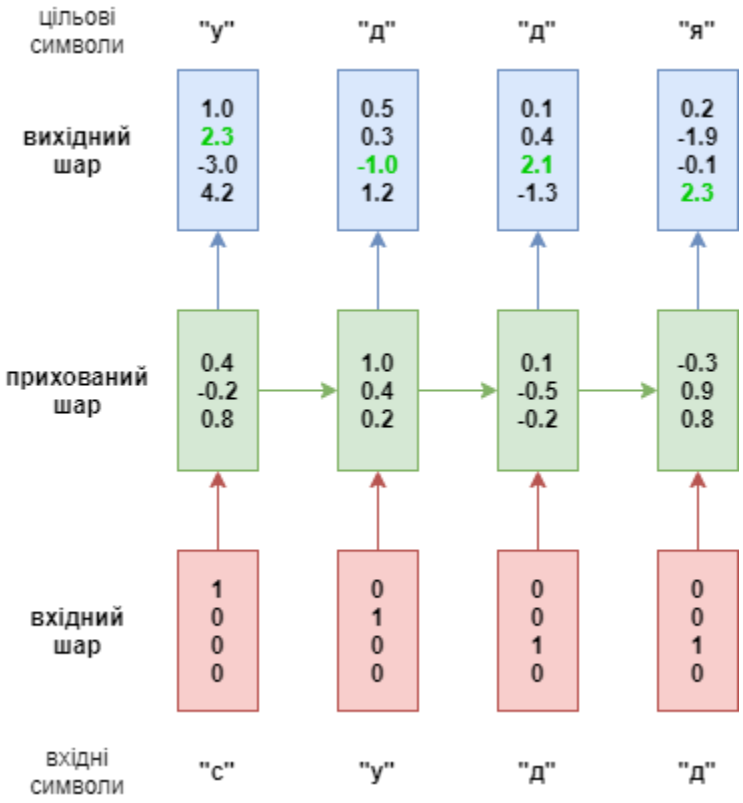


Рисунок 1.2.1. – Приклад нейронної мережі, що передбачає наступну літеру

Особливістю рекурентної нейронної мережі також є те, що вона не використовує різні параметри для кожного прихованого шару, що суттєво

зменшує їх кількість на процесі навчання. Вектори W, V, U є однаковими для кожного з кроків, що зображений на розгортці.

Для навчання мережі використовують модифікований метод зворотного розповсюдження помилки – «розповсюдження помилки скрізь час» (англ. BPTT – Backpropagation Through Time). Оскільки кожен прихований шар залежить від попереднього, обрахунок градієнта «розповсюджує» помилку на всі попередні кроки та сумує результат. Відтак, виникає проблема зникання градієнту (англ. gradient vanishing problem). Як було вказано раніше, саме через неможливість врахування залежностей, що розташовані далеко одне від одного, рекурентна нейронна мережа не завжди працює оптимально у завданнях, де важливий попередній контекст. Ця проблема дала поштовх для дослідження нових архітектур нейронних мереж.

1.2. Поняття довгої короткочасної пам’яті

ДКЧП – це один з видів рекурентних нейронних мереж, що значно ефективніше зберігає та враховує довготривалі залежності, тобто ті випадки, коли між важливими подіями існують затримки невідомої тривалості. Вперше цю архітектуру описали Зепп Хохрайтер та Юрген Шмідгубер у 1997 році.

Основою комірки ДКЧП є комірка пам’яті, оскільки саме там зберігається інформація на довготривалий проміжок часу. На відміну від стандартної архітектури рекурентних нейронних мереж, окрім передачі попереднього стану передається також комірка пам’яті. Завдяки цьому інформація між станами може передаватись на значно довшу відстань.

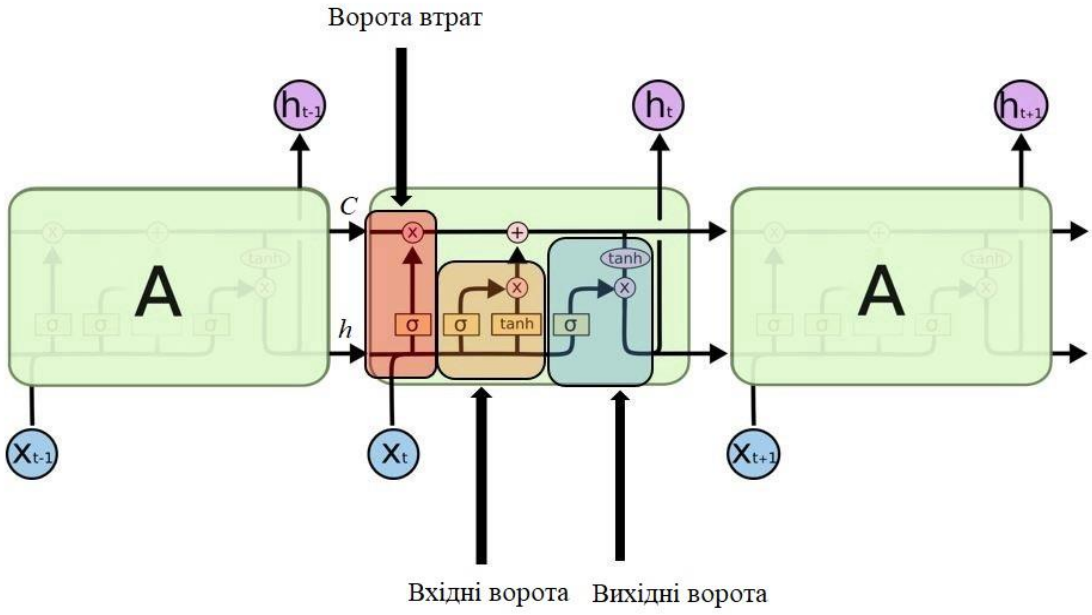


Рисунок 1.2.1 – Загальна структура комірки ДКЧП

Для того, щоб контролювати інформацію, зменшувати або збільшувати кількість інформації, яка передається, використовують структуру воріт (англ. gate). Існує три типи воріт:

1. Ворота втрат або забуття (Forget gate)

Визначають інформацію, яку потрібно вилучити з стану комірки. Ці ворота аналізують дані на вході нейронної мережі та вихідні дані комірки попереднього кроку. Для фільтрації використовують функцію сигмоїд, що повертає значення від 0 до 1, де 1 – зберегти, 0 – видалити. Обчислення відбувається за формулою:

$$f_t = \sigma\big(W_f * [h_{t-1}, x_t] + \ b_f\big).$$

В залежності від входу x_t та стану попередньої комірки пам’яті C_{t-1} мережа вирішує, яку інформацію необхідно зберегти та передати далі.

2. Вхідні ворота (Input gate)

Призначені для того, щоб вирізнити, яка нова інформація має бути збережена в комірці пам'яті, а яка – ні. Спочатку для цього використовують функцію сигмоїд, результатом якої є число від 0 до 1. Важливість переданої інформації визначає функція tanh, рангуючи отримані коефіцієнти від -1 до 1. Ця функція створює нові значення, які додають в стан.

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i),$$

$$\widetilde{C_t} = \tanh(W_C * [h_{t-1}, x_t] + b_C).$$

3. Вихідні ворота (Output gate)

Оновлюють попередній стан комірки для отримання нового стану. Для того, щоб отримати новий стан потрібно перемножити старий стан на інформацію, яку необхідно відкинути (це попередньо визначили ворота втрат), тобто:

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t}.$$

Наступним кроком є фільтрація стану комірки та обрахунок прихованого шару h_t:

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o),$$
$$h_t = o_t * \tanh(C_t) \text{ [3].}$$

Для того, щоб зрозуміти яким чином ДКЧП вирішує проблему зникання градієнту розглянемо одновимірний випадок.

Маємо прихований шар h_t в момент часу t. Тоді:

$$h_t = \sigma(W_o * h_{t-1}).$$

У даному випадку для спрощення та наочності результату пошуку похідної прибираємо вхідний шар та зсуви.

$$\frac{\partial h_{t'}}{\partial h_t} = \prod_{k=1}^{t'-t} W\sigma'(Wh_{t'-k}) = W^{t'-t} \prod_{k=1}^{t'-t} W\sigma'(Wh_{t'-k}).$$

Якщо ваги (параметри моделі) не дорівнюють одиниці, то дане значення буде експоненційно швидко спадати до нуля. Це спричинить вищевказану проблему зникання градієнту. Інакше, якщо похідна набуватиме великих значень, то при тренуванні мережі велика імовірність отримати проблему вибуху градієнту (англ. exploding gradient problem).

Введення стану комірки пам’яті в ДКЧП дозволяє уникнути обох цих проблем, оскільки перед передачею інформація проходить через ворота втрат і при диференціюванні не виникає експоненційно зростаючого або спадаючого фактору. Маємо

$$\frac{\partial \mathrm{C}_{t'}}{\partial C_t} = \prod_{k=1}^{t'-t} \sigma(f_{t+k}),$$

де f_t – це вхід воріт втрат [4].

1.3. Використання РНМ в обробці природної мови

Однією з найбільш популярних нині проблем в сфері штучного інтелекту є вирішення проблеми комунікації комп’ютера та людини, тобто створення алгоритмів, які дозволять комп’ютеру розуміти людське спілкування та вміти генерувати розумний текст. Підхід до обробки тексту, метою якого є аналіз та синтез мови, називають обробкою природної мови. Основними завданнями обробки природної мови є вивчення та пошук зв’язків у даних, розпізнавання тексту (з документів, фото, відео, аудіо), синтез

мовлення (озвучування тексту та його генерація), машинний переклад з однієї мови на іншу, інформаційний пошук, морфологічна декомпозиція (тобто спрощення технічних понять у зрозумілі всім пояснення), класифікація семантичного забарвлення тексту тощо.

Основними поняттями, що стосуються обробки природної мови, є лексичний розбір (токенизація), n-грами, унітарне кодування та векторне представлення слів (англ. word embeddings).

Токенизація – це процес конвертування послідовності символів у послідовність токенів.

N-грами дозволяють об’єднувати близькі слова задля їх подальшого представлення, n – кількість слів, які мають бути об’єднані. Наприклад, речення «РНМ використовують в обробці природної мови» можна розбити на уніграми, що міститимуть по одному слову окремо («РНМ, використовують, в, обробці, природної, мови»), біграми (по два слова), триграми тощо. Дане розбиття природної мови використовується для передбачень на основі імовірнісних моделей, тобто знаходження імовірності певного слова n-грами, якщо відомі всі попередні.

Унітарне кодування, що було розглянуто в прикладі роботи РНМ у пункті 1.1, дозволяє представляти слова в числовій формі. Закодувавши слова з алфавіту таким чином, для кожного слова отримаємо його представлення у вигляді вектору, який складається з нулів та містить всього одну одиницю на місці номеру його позиції в словнику. По-перше, такий підхід не враховує семантичне значення слів та унеможливлює структурування схожих слів. По-друге, зі збільшенням розміру словника, збільшується розмірність вектора. По-третє, так як даний вектор складається переважно з нулів та має велику

розмірність, це спричиняє певні проблеми з обрахунками, а саме ризик отримання помилки переповнення пам’яті. Але найбільш серйозний недолік – завдання узагальнення. Необхідно мати можливість передати моделі інформацію про схожі слова в словнику, так як набагато простіше працювати з чимось, що модель вже опрацьовувала та на основі попередніх результатів робити узагальнення для нового слова, а не вчитися поводитися з ним з нуля. Тож враховуючи суттєвий перелік недоліків, дане кодування варто використовувати в обробці природної мови, якщо дані не є контекстно-корельованими (тобто словник складається з незалежних слів, що не мають однакового чи схожого значення), а також якщо наявний великий датасет.

Векторне представлення слів – це збірна назва для методів навчання, де слова чи фрази з словника мають своїм відповідником вектор реальних чисел. Дана методика використовується у нейронних мережах та вирішує недоліки унітарного кодування. Основна ідея полягає в тому, щоб представити близькі семантично слова схожим чином. Розглянемо на прикладі:

	тварина	моторошний	небезпечний	пухнастий
кіт	0.98	0.10	0.35	0.98
ковдра	0.01	0.02	0.12	0.84
кит	0.98	0.85	0.91	0.02
зомбі	0.74	0.93	0.98	0.05

Таблиця 1.3.1 – Приклад представлення матриці семантично схожих слів

У даному прикладі для кожного слова існує також числове значення, що описує його належність певній особливості. Кількість таких особливостей впливає на розмірність матриці, тож завдання їх визначення є емпіричним. Найбільш часто використовуваним значенням є 300. Найбільш популярними

бібліотеками для реалізації векторного представлення є Word2Vec, GloVe, FastText.

Так як використання традиційних нейронних мереж зобов’язує використовувати вектори фіксованого розміру, це унеможливлює роботу з послідовностями. РНМ дають мережі гнучкість та можливість динамічної поведінки: кожне окреме слово речення окремо подається на вхід нейронної мережі. Вдосконалення архітектури РНМ, а саме створення ДКЧП дало сильний поштовх для подальшого розвитку обробки природної мови та вирішення її основних завдань.

РОЗДІЛ 2. Анотація зображень. Проблема та особливості вирішення

2.1. Завдання генерації автоматичного тексту для зображення

Анотація зображень – це фундаментальна проблема штучного інтелекту, яка об’єднує в собі алгоритми комп’ютерного зору та обробки природної мови. Кількість медіа-інформації щодня зростає, тож окрім застосування у сферах біомедицини, навчання та електронних бібліотек, наразі алгоритми анотації зображень найбільш широко використовуються для пошуку зображень за вмістом (англ. Content-based image retrieval (CBIR)).

Завдання генерації автоматичного тексту складається з двох основних частин: розуміння того, що знаходиться на зображенні, а також опис цих речей природною мовою. Для того, щоб зрозуміти вміст зображення використовуються алгоритми детекції та розпізнавання об’єктів. Після отримання даної інформації важливо зрозуміти тип та місце розташування об’єктів, як вони взаємодіють між собою та сформувати речення. Формування речення вимагає від нас не лише семантичного розуміння того, що знаходиться на фото, але й синтаксично грамотного оформлення.

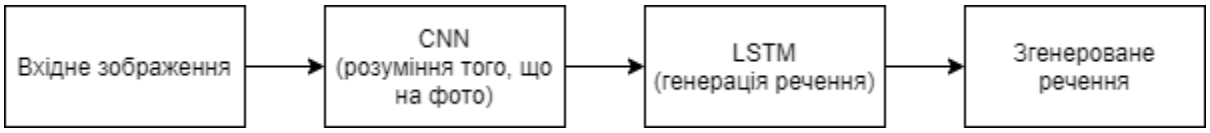


Рисунок 2.1.1. Схема архітектури для розв’язання завдання анотації зображення

Детекція та розпізнавання об’єктів на зображенні реалізується за допомогою згорткових нейронних мереж. Згорткові нейронні мережі (ЗНМ) вперше були запропоновані Яном Лекуном у 1988 році та являють собою послідовність згорткових, агрегувальних, повноз’єднаних та нормалізуючих шарів. Наприклад, для навчання розпізнання об’єктів стандартна архітектура

ЗНМ буде виглядати так $[INPUT - CONV - RELU - POOL - FC]$, тут INPUT – це вхідний шар, що отримує трьохвимірний вектор (зображення) [WxHx3], де кожне зображення має висоту H, ширину W, а також 3 кольорові канали R, G, B.

Шар CONV обчислює добуток між попереднім шаром та фільтрами, які і є параметрами нейронної мережі у даному випадку. Кількість фільтрів та їх розмір вибираються емпірично (це гіперпараметри), коефіцієнти спочатку ініціалізуються випадковим чином, а надалі метою ЗНМ є використовуючи алгоритм зворотного розповсюдження помилки змінити їх, щоб мінімізувати функцію втрат. Даний шар отримує на вхід вектор $W_1xH_1xD_1$, вимагає визначення таких гіперпараметрів як K – кількість фільтрів, F – розмірність, S – крок, з яким потрібно застосовувати фільтр, P – розмір рамки, яку необхідно заповнити нулями. На виході отримаємо вектор $W_2xH_2xD_2$, де

$$W_2 = \frac{W_1 - F + 2P}{S + 1},$$

$$H_2 = \frac{H_1 - F + 2P}{S + 1},$$

$$D_2 = K.$$

Кількість параметрів на даному шарі дорівнює $(F * F * D_1) * K$. Загальноприйнятими значеннями гіперпареметрів є F = 3, S = 1, P = 1.

Шар RELU поелементно застосовує активаційну функцію $\max(0, x)$. При цьому розмірність залишається незмінною. Не має параметрів.

Шар POOL використовується для зменшення розмірності та зазвичай використовує функцію max. Як і попередній, не має параметрів. Цей шар також називають шаром субдискретизації.

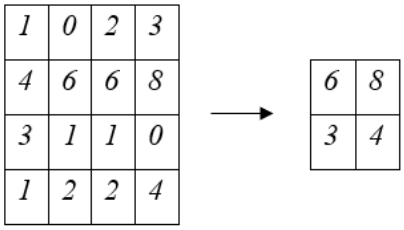


Рисунок 2.1.2. Приклад застосування шару субдискретизації з фільтром 2x2, кроком 2 та функцією max

На останньому повнозв'язному шарі нейронної мережі застосовується функція Softmax, яка повертає вектор v_i, координати якого трактуються як імовірність, що об'єкт належить до класу i. Обчислення даних імовірностей відбувається наступним чином:

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}.$$

Розпізнавши на зображенні об'єкти, наступним кроком у вирішенні завдання анотації зображень є генерація речення. Для цього використовуються рекурентні нейронні мережі, а саме – згадана вище модифікована архітектура ДКЧП, яка була описана в пункті 1.2. Наприклад, маємо речення «студент сидить за столом». Тренування мовної моделі полягає в збільшенні імовірності появи наступного слова, за умови попередніх слів:

$$P(\text{наступне слово}|\text{попередні слова}).$$

Тобто при тренуванні моделі наступні імовірності мають бути якомога вищими:

$$P(\text{студент} \mid [< start >]),$$
$$P(\text{сидить} \mid [< start >, \text{студент}],$$
$$P(\text{за} \mid [< start >, \text{студент}, \text{сидить}],$$

$$P(\text{столом} \mid [< start >, \text{студент}, \text{сидить}, \text{за}].$$

2.2. Датасети для навчання

2.2.1. MS COCO

MS COCO – один з найбільших датасетів для розпізнавання об’єктів на зображеннях, сегментації та анотації. Містить 330000 зображень, 1.5 млн екземплярів об’єктів, 80 категорій об’єктів, 5 підписів для кожного зображення.

2.2.2. Flickr8K

Один з найбільш популярних датасетів для невеликих проектів з обмеженим GPU та оперативною пам’яттю, оскільки він має всього 8000 фото та 5 підписів до кожного з них, що були розмічені людьми. Дані для тренування складаються з 6000 фото, для тестування та валідації – по 1000 фото. Зазвичай використовується у навчальних проектах.

2.2.3. Flickr30K

Даний датасет містить 30000 зображень та 158000 підписів, розмічених людьми. У ньому немає фіксованого розбиття на тренувальну, тестову та валідаційну вибірки, тож розробники мають можливість самостійно визначити бажане розбиття.

2.2.4. Visual Genome

Основною відмінністю даного датасету від попередніх є те, що він містить анотацію не всього фото, а його окремих частин. Датасет містить більш ніж 108000 фото, кожне з них має в середньому 35 об’єктів, 26 атрибутів та 21 зв’язок між об’єктами. Набір даних має шість основних частин: описи регіонів, атрибути, об’єкти, зв’язки, графіки регіонів та графіки сцен.

2.2.5. IAPR TC-12

Даний датасет містить 20000 зображень, що зібрані з різноманітних джерел (спорт, фото людей, тварин, природи тощо). Основною відмінністю є наявність анотацій для зображення різними мовами.

2.3. Метрики для оцінювання точності моделі

2.3.1. BLEU

BLEU (Bilingual Evaluation Understudy) – це, безумовно, найпопулярніша метрика для оцінювання якості автоматично згенерованого тексту. У BLEU точність (англ. precision) та повнота (англ. recall) апроксимуються найкращою довжиною співпадіння та видозміненою точністю n-грам.

Точність n-грами – це частка n-грам в тексті кандидата, які є присутні в будь-якому з відповідних текстів.

$$Precision = \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), w_n = \frac{1}{n}$$

Найкраща довжина співпадіння обраховується за формулою

$$BP = \begin{cases} 1, & \text{якщо } c > r \\ \exp\left(1 - \frac{r}{c}\right), & \text{інакше} \end{cases}$$

Тут c – це загальна довжина перекладу кандидата, r – ефективна контрольна довжина (середня значення всіх посилань). Зі зменшенням довжини кандидата співвідношення $\frac{r}{c}$ збільшується.

2.3.2. ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) – як зрозуміло з назви, дана метрика орієнтується лише на повноту та використовується для кінцевої оцінки. Є декілька різновидів даної метрики: ROUGE-N ґрунтується

на n-грамах. Наприклад, ROUGE-1 рахує повноту базуючись на співпадінні уніграм, ROUGE-2 базуючись на співпадінні біграм тощо. ROUGE-L базується на понятті найдовшої спільної послідовності (англ. LCS), ROUGE-W – зваженої найдовшої спільної послідовності.

2.3.3. METEOR

METEOR (Metric for Evaluation of Translation with Explicit Ordering) – ця метрика модифікує точність та повноту обчислень, замінюючи їх на зважений F-score на основі відображення уніграм і штрафу для функції за неправильний порядок слів.

$$F = \frac{PR}{\alpha P + (1-\alpha)R},$$

$$Penalty = \gamma(\frac{c}{m})^{\beta}, \text{де } 0 \leq \gamma \leq 1,$$

де c – кількість співпадаючих частин, m – загальна кількість співпадінь.

$$METEOR = (1 - Penalty)F \text{ [5].}$$

РОЗДІЛ 3. Програмна реалізація

3.1. Постановка задачі

Завданням роботи було створити нейронну мережу, яка буде автоматично генерувати текст до зображення, а також реалізувати можливість опису речення українською мовою та його озвучування. А також створити зручний та доступний інтерфейс, що дозволить завантажувати фото і отримувати результат передбачення моделі. Дане завдання корисне не лише в дослідницьких цілях, оскільки в подальшому дану систему можна покращити, дозволивши робити фото в режимі реального часу, що допоможе людям з вадами зору розпізнавати те, що знаходиться навколо, зробивши фото.

Математична постановка задачі – це оптимізація відносно параметрів моделі θ:

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

3.2. Опис використаних технологій

Для написання роботи було обрано мову програмування Python, а також відкриту нейромережну бібліотеку Keras. Keras – це зручна у використанні, модульна бібліотека, яка дозволяє працювати з мережами глибинного навчання, містить в собі різноманітні реалізації найбільш часто використовуваних блоків нейронних мереж та інструменти для написання власних.

Датасет, який був використаний для тренування PHM, – Flickr8k.

Для реалізації озвучування тексту українською мовою було використано Google Text-to-Speech API – це єдина наявна на даний момент система, яка

підтримує озвучування українською мовою. Плата за використання $16.00 USD / 1 млн. символів.

Для перекладу тексту українською мовою використано Google Translate API.

Для створення вебсайту було використано Flask. Flask – це сучасний во мікрофреймворк для створення вебсайтів на мові програмування Python.

3.3. Архітектура нейронної мережі

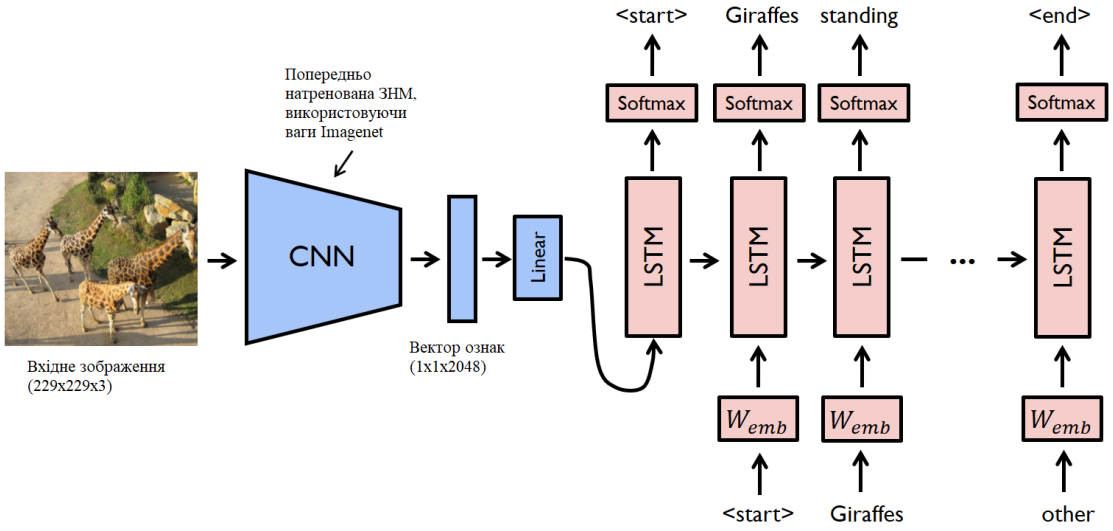


Рисунок 3.3.1 – Схема архітектури реалізованої нейронної мережі

У роботі об’єднано дві нейронні мережі (згорткова та рекурентна): для зображень використано архітектуру Inception-V3, результат якої передається на вхід іншої для генерації речення – РНМ базується на архітектурі LSTM з модифікаціями. Використовується підхід кодування (зображення представляємо як вектор) та декодування (отримуємо анотацію із векторного представлення зображення).

3.3.1. ЗНМ. Inception-V3

Загалом існує 4 версії архітектури Inception від Google. Перша з них Inception-V1 була проривом у сфері розпізнавання об’єктів на фото та отримала перше місце у найбільш престижному змаганні ISLVRC (ImageNet Large Scale Visual Recognition Competition) у 2015 році. Суттєвою відмінністю від попередніх лідерів (AlexNet, VGGNet) було різке зменшення кількості необхідних для тренування параметрів: AlexNet містить 60 млн параметрів, VGGNet – 180 млн параметрів, а Inception-V1 – лише 7 млн параметрів. Поява пакетної нормалізації (англ. batch normalization) – методу, що дозволяє стабілізувати роботу нейронних мереж шляхом зміни підходу до пошуку градієнту (на кожній ітерацій навчальна вибірка переглядається повністю і лише після цього змінюються ваги моделі) спричинила створення Inception-V2. Пізніше, додавши ідею факторизації, виникла Inception-V3. Метою факторизації є зменшення параметрів моделі без зменшення її ефективності. Основними ідеями факторизації є:

1. Зменшення розмірності фільтру згортки. Фільтри 3x3 замінено на фільтри 5x5.

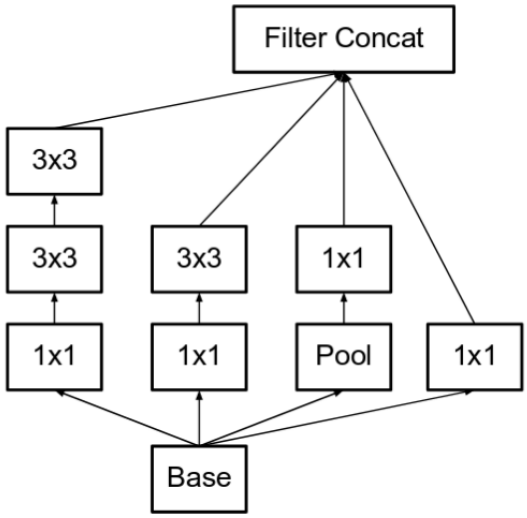


Рисунок 3.3.1.1 – Схема модуля A архітектури Inception-V3

2. Факторизація в асиметричні фільтри, тобто використовуючи фільтр 3x3 кількість параметрів 9, використовуючи фільтри 3x1 та 1x3 кількість параметрів 6 (зменшилась на 33%)

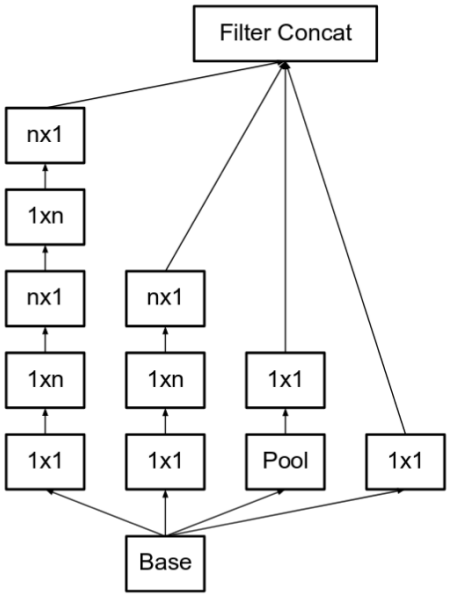


Рисунок 3.3.1.2 – Схема модуля B архітектури Inception-V3 після застосування факторизації

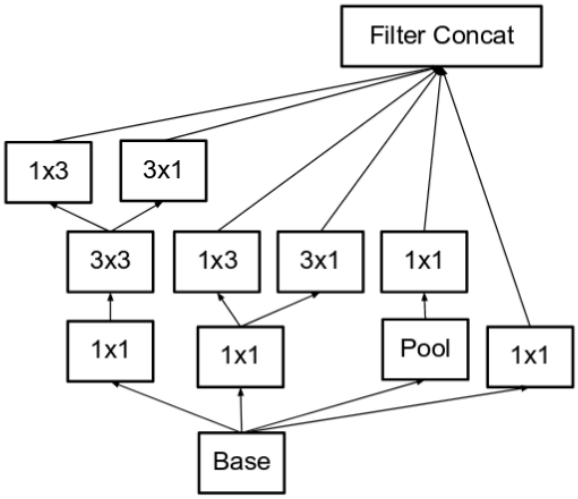


Рисунок 3.3.1.3 – Схема модуля C архітектури Inception-V3 після застосування факторизації

Загальну схему архітектури Inception-V3 можна переглянути у додатку А.

Параметрами моделі взято ваги з попередньо натренованої моделі на датасеті Imagenet.

На вхід модель приймає зображення (299, 299, 3) з попередньою обробкою:

$$x = {}^{x}/_{255.},$$
$$x = x - 0.5,$$
$$x = x * 2,$$

де x – значення пікселю зображення.

3.3.2. Генерація речення

Значенням розмірності для матриці векторного представлення слів було взято 300, в якості оптимізатора використовувались Adam та RMSprop.

Структура шарів: Embedding -> LSTM -> TimeDistributed -> Bidirectional(LSTM()) -> Dense -> Activation('softmax').

Існує два підходи для написання декодера моделі та утворення фінального речення: жадібний та променевий пошук. Жадібний алгоритм простий в реалізації і швидкий, оскільки на кожному кроці завжди бере найбільш імовірне слово, але даний підхід не є оптимальний. Для створення речення зазвичай використовується променевий алгоритм (англ. Beam Search Algorithm). Це евристичний алгоритм, ідея якого полягає в його ідея полягає в тому, щоб взяти перші k передбачень, передаємо їх на вхід моделі і сортуємо використовуючи імовірності, що повертає модель. Після цього необхідно обрати найбільш імовірне слово і рухатись поки не досягнеться максимально

можлива довжина речення або ж поки не зустрінемо символ закінчення рядка. Цей алгоритм є ефективнішим за жадібний пошук та показує значно кращі результати. Саме його було використано для формування речення.

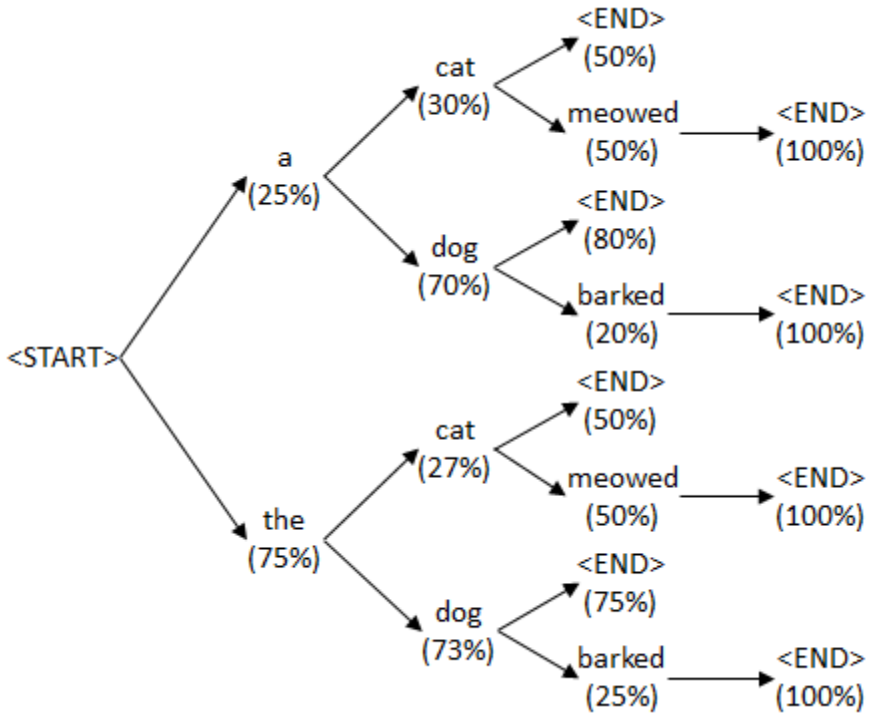


Рисунок 3.3.2.1 – Приклад роботи променевого алгоритму

3.4. Аналіз отриманих результатів

Після тривалого тренування та підбору гіперпараметрів, вдалось досягти значення loss = 0.3860, acc = 0.8832.

На основі реалізованої моделі створено програмний продукт, який дозволяє завантажити власне фото, отримати автоматично згенерований підпис та озвучити його українською мовою.

У користувача є можливість завантажити фото. Якщо формат файлу не підтримується або ж він не був вибраний видається повідомлення про помилку.

Рисунок 3.4.1 – Скріншот вхідної сторінки

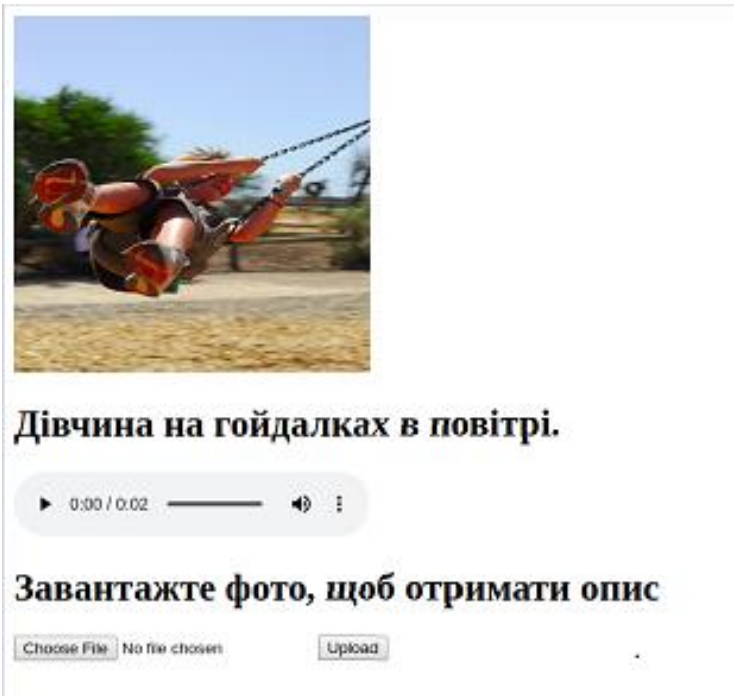


Рисунок 3.4.2 – Скріншот отриманого результату

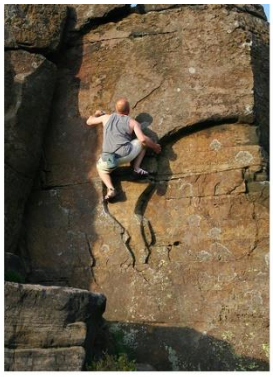
Далі можна переглянути вдало згенеровані моделлю речення до відповідних зображень:

Хлопчик в смугастій сорочці їзда на велосипеді.

Чорно-біла собака працює по піску.

Чорна собака грає з зеленим кулею.

Мотоцикліст їде на грунтовій пагорбі.



Людина скелелазіння.

Людина виконує трюк на скейтборді в парку

Маленька дитина стрибає високо на батуті

Мокрий собака, граючись з іграшкою

Маленька дитина

Тенісист тримає ракетку вгору

Собака біжить у дворі з іграшкою в роті.

Дитина хлюпочуться в дитячому басейні

Собаки тягнути сани по снігу.

Троє дітей позують на виступі перед групою людей.

Баскетбольний гравець тримає в баскетбол.

Рисунок 3.4.3 – Приклад згенерованих анотацій

Звісно, трапляються і неправильно згенеровані речення, та попри це модель все ж розпізнає певні об’єкти на фото і вказує їх правильно:

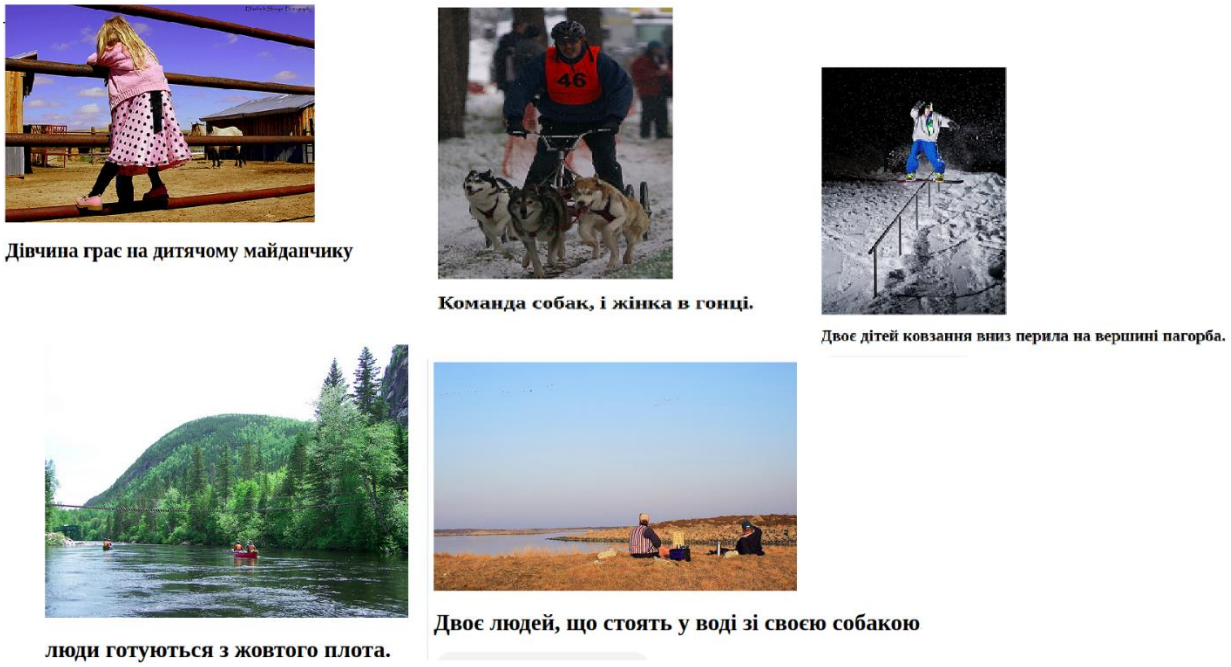


Рисунок 3.4.4 – Приклад неточно згенерованих речень

ВИСНОВКИ

Метою даної курсової роботи було реалізувати алгоритм анотації зображенння. Під час виконання роботи, було проаналізовано статті та дослідження існуючих нині архітектур для вирішення даного завдання, досліджено та систематизовано загальні відомості про те, що таке згорткова нейронна мережа, рекурентна нейронна мережа та її модифікаційна архітектура ДКЧП, пояснено загальні підходи до розв’язання цієї задачі.

У теоретичній частині було детально описано застосування рекурентних нейронних мереж в обробці природної мови, розглянуто основні датасети та метрики для оцінювання моделі, наведені приклади.

У практичній частині було розроблено власну архітектуру нейронної мережі для анотації зображення, дещо спрощену у порівнянні з існуючими зараз. Програш аналогам у точності генерації пояснюється тим, що було використано невеликий датасет Flickr8K з практичних міркувань в умовах обмежених обчислювальних ресурсів та з метою підвищення швидкості тренування моделі. А також створено веб-сайт, який дозволяє завантажувати фотографію та отримувати її опис. Ключовою відмінністю від нині існуючих систем є генерація тексту українською мовою та його озвучування.

Подальшим розвитком даної роботи може стати донавчання нейронної мережі на більшій вибірці даних (наприклад, MS COCO), створення мобільного додатку та розширення функціоналу – фото можна буде не лише завантажити з пам’яті, але і зробити в режимі реального часу. Даний продукт значно покращить якість життя людей з вадами зору.

Список використаних джерел

1. Aurelien Geron. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, 2nd Edition / Aurelien Geron. – O'Reilly Media, 2019. – 856 с.
2. Andrej Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks, 2015 [Електронний ресурс]. – Режим доступу: http://karpathy.github.io/2015/05/21/rnn-effectiveness/
3. Understanding LSTM Networks [Електронний ресурс]. – Режим доступу: https://colah.github.io/posts/2015-08-Understanding-LSTMs/
4. Bayer, Justin Simon. Learning Sequence Representations, 2015. [Електронний ресурс]. – Режим доступу: https://d-nb.info/1082034037/34
5. Yin Cui, Guandao Yand, Andreas Veit, Xun Haung, Serge Belongie. Learning to Evaluate Image Captioningm, 2018 [Електронний ресурс]. – Режим доступу: https://vision.cornell.edu/se3/wp-content/uploads/2018/03/1501.pdf
6. Md. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, Hamiga Laga. A Comprehensive Survey of Deep Learning for Image Captioning, 2018 [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/1810.04020.pdf
7. Orio Vinayaks, Alexander Toshev, Samy Bengio. Dumitru Erhan. Show and Tell: A Neural Image Caption Generator, 2015 [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/1411.4555.pdf
8. Orio Vinayaks, Alexander Toshev, Samy Bengio, Dumitru Erhan. Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge, 2016 [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/1609.06647.pdf

Додаток А. Схема архітектури Inception-V3