Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*
(повна назва)
Кафедра *Системотехніки*
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

ГЮИК. 502610.010 ПЗ
(позначення документа)

«Дослідження і застосування методів розпізнавання образів на прикладі структурованих символів»
(тема)

Виконав:
Студент *2* курсу, групи *СПРм-18-1*

Сергєєв К. В.
(прізвище, ініціали)

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

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

Освітня програма *Системне проектування*
(повна назва освітньої програми)

Керівник *проф. Ситніков Д. Е.*
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри системотехніки _____ *Гребеннік І. В.*
(підпис) (прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*
Кафедра *Системотехніки*
Рівень вищої освіти *другий (магістерський)*
Спеціальність *122 – Комп'ютерні науки*
Тип програми *освітньо-професійна*
Освітня програма *Системне проектування*

ЗАТВЕРДЖУЮ:
Зав. кафедри
_____
(підпис)
«_____»_____ 2019 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ (ПРОЕКТ)

студентові *Сергєєву Кирилу Вячеславовичу*
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) *Дослідження і застосування методів розпізнавання образів на прикладі структурованих символів*
затверджена наказом по університету від __ листопада 2019 р. № _____
2. Термін подання студентом роботи (проекту) *4 грудня 2019 р.*
3. Вихідні дані до роботи *Тема дослідження, дані Інтернет-джерел та відомих наукових проектів з тематики дослідження і застосування алгоритмів розпізнавання послідовностей символів на зображеннях з використанням нейронних мереж. Перелік використаних програмних засобів: Ubuntu 19.04, Python 3.7, OpenCV 3.4.8, Tesseract 4.1, TensorFlow 1.15. Технічне забезпечення: комп'ютер, підключений до Інетрнету.*
4. Зміст пояснювальної записки *4.1 Вступ. 4.2 Огляд і аналіз сучасного стану задачі розпізнавання тексту на зображеннях. 4.3 Аналіз особливостей пошуку і розпізнавання тексту на зображеннях. 4.4 Алгоритми пошуку і розпізнавання тексту з використанням нейронних мереж. 4.5 Розробка гібридної архітектури нейронних мереж для розпізнавання структурованих послідовностей символів. 4.6 Висновки.*

5. Перелік графічного матеріалу *5.1 Мета атестаційної роботи (1 аркуш формату А4). 5.2 Стандартні методи оптичного розпізнавання (1 аркуш формату А4). 5.3 Повнозв'язні нейронні мережі (1 аркуш формату А4). 5.4 Згорткові нейронні мережі (1 аркуш формату А4). 5.5 Рекурентні нейронні мережі (1 аркуш формату А4). 5.6 LSTM нейронні мережі (1 аркуш формату А4). 5.7 Архітектура комбінованої мережі (1 аркуш формату А4). 5.4 Висновки по роботі (1 аркуш формату А4). 5.8 Тренування і тестові набори (1 аркуш формату А4). 5.9 Порівняння з аналогами (1 аркуш формату А4). 5.10 Висновки по роботі (1 аркуш формату А4). 5.11 Висновки по роботі (1 аркуш формату А4).*

6. Консультанти розділів роботи (проекту)

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

7. Дата видачі завдання *1 вересня 2019 р.*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термін виконання	Прим.
1	Отримання завдання на дипломне проектування, виділення предмету і об'єкту дослідження	01.09 – 10.09.19	
2	Аналіз задачі та існуючих аналогів	11.09 – 28.09.19	
3	Дослідження та розробка архітектури мережі	29.09 – 15.10.19	
4	Генерація тестових данних та навчання мережі	16.10 – 25.10.19	
5	Обробка наборів даних та тестування мережі	26.10 – 05.11.19	
6	Порівняльний аналіз результатів тестування	06.11 – 15.11.19	
7	Оформлення пояснювальної записки	16.11 – 30.11.19	
8	Представлення на рецензування	06.12.19	
9	Представлення дипломної роботи в деканат	10.12.19	

Студент _____
(підпис)

Керівник роботи (проекту) _____ *проф. Ситніков Д. Е.*
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи містить: 57 сторінок, 2 таблиці, 24 рисунки, 4 додатки, 30 джерел.

Об’єкт дослідження – методи розпізнавання образів.

Предмет дослідження – методи та моделі розпізнавання послідовностей символів на зображеннях на основі штучних нейронних мереж.

Мета роботи – аналіз алгоритмів виділення об’єктів на зображенні та методів, що застосовуються при обробці зображень для розпізнавання текстових даних. Експериментальне застосування і оцінка алгоритмів розпізнавання тексту з використанням нейронних мереж.

Методи дослідження – бібліотеки OpenCV та Tesseract, які реалізують алгоритми машинного зору.

Результати роботи – формальний опис і експериментальна побудова і перевірка нейронної мережі, що поєднує переваги згорткових і рекурентних нейронних шарів для обробки даних з зображень, що містять текст.

Область застосування – описані алгоритми і структура нейронної мережі можуть застосовуватися для розпізнавання не лише текстових, але й інших графічних даних на зображеннях – музичних нот, дорожніх знаків, спеціальних символів маркування.

ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, МАШИННЕ НАВЧАННЯ, РЕКУРЕНТНА НЕЙРОННА МЕРЕЖА, РОЗПІЗНАВАННЯ ОБРАЗІВ, РОЗПІЗНАВАННЯ ТЕКСТУ, ТЕКСТ СЦЕНИ, OPENCV, TESSERACT

ABSTRACT

The explanatory note of the appraisal work contains: 57 pages, 2 tables, 24 figures, 4 appendices, 30 sources.

Object of study – methods of object recognition.

The subject of the study is the recognition of sequences of characters on images using neural networks.

The purpose of the work is to analyze the algorithms for objects detection on an image and the methods used in image processing for recognizing textual data. Experimental application and evaluation of text recognition algorithms using neural networks.

Research methods – OpenCV and Tesseract libraries that implement computer vision algorithms.

The results are a formal description and experimental construction and validation of a neural network that combines the benefits of convolutional and recurrent neural layers for processing data from images containing text.

Usage scope – The described algorithms and neural network structure can be used to recognize not only text but also other graphic data on images – musical notes, road signs, special marking symbols.

CONVOLUTIONAL NEURAL NETWORK, MACHINE LEARNING, OBJECT RECOGNITION, OPENCV, RECURRENT NEURAL NETWORK, SCENE TEXT, TESSERACT, TEXT RECOGNITION

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AdaBoost – Алгоритм машинного навчання, скорочення від adaptive boosting;

BPTT – Зворотне поширення в часі (з англ. backpropagation through time);

CNN – Згорткова нейронна мережа (з англ. convolutional neural network);

CRNN – Нейронна мережа з поєднанням згорткових і рекурентних шарів;

CRF – Умовні випадкові поля (з англ. conditional random fields);

EAST – Детектор тексту на зображеннях з навколишнього середовища;

FCN – Повнозв’язна згорткова мережа (з англ. fully convolutional network);

GRU – Вентильні рекурентні вузли (з англ. gated recurrent units);

LDA – Лінійний дискримінантний аналіз (з англ. linear discriminant analysis);

LSTM – Довга короткочасна пам’ять (з англ. long short-term memory);

OCR – Оптичне розпізнавання символів (з англ. optical character recognition);

ReLU – Функція активації типу «випрямляч» (з англ. rectified linear unit);

RNN – Рекурентна нейронна мережа (з англ. recurrent neural network).

ВСТУП

Розпізнавання тексту – одне з ранніх завдань комп’ютерного зору [1]. Це змушує багатьох людей вважати, що ця проблема вирішена. Ще одна думка полягає в тому, що розпізнавання текстової інформації не вимагає глибокого навчання та воно є надмірним. Але не існує такого поняття, як вирішена задача. Для деяких завдань дійсно є рішення, які не потребують глибокого навчання і стандартні підходи OCR дають хороші результати, однак глибоке навчання є передовим методом вирішення більш загальних і складних задач. Саме зараз постає проблема обробки дуже великого об’єму даних з різних джерел, в тому числі смартфонів, камер спостереження, датчиків на конвеєрах та в автомобілях, де розпізнавання об’єктів є ключовим для широкого спектра практичних завдань. Ці дані значно відрізняються за своєю структурою та якістю зображення від звичайних печатних документів. Тому все більше уваги приділяється до алгоритмів вирізнення тексту на довільних зображеннях, так званого тексту сцени, та програмним засобам, що використовують такі алгоритми.

Зі швидким розвитком глибокого навчання впроваджуються потужні інструменти, які здатні засвоїти смислові, глибші риси для вирішення проблем, що існують у традиційних архітектурах розпізнавання образів.

Складність рішення задачі розпізнавання образів знаходиться в сильній залежності від особливостей їх графічного зображення. Вибір конкретних методів обумовлений особливостями об’єкта, який потрібно розпізнати. Ще одна проблема полягає в тому, що описати всі властивості практично неможливо і ці властивості можуть відповідати не усім об’єктам, тому при вирішенні задачі розпізнавання необхідно знайти оптимальне співвідношення складності обчислень і бажаної точності.

В роботі описані основні складності задач виявлення та розпізнавання тексту та описані шляхи їх вирішення. Проаналізовані популярні алгоритми

розпізнавання образів та їх застосування відносно задачі виявлення тексту, описано використання засобів машинного навчання для розпізнавання тексту на зображеннях для розпізнавання структурованих послідовностей символів.

Метою даної роботи є огляд та дослідження алгоритмів розпізнавання об’єктів на зображеннях, аналіз методів, що застосовуються при обробці зображень. Також в роботі описано використання засобів машинного навчання для пошуку та розпізнавання тексту на зображеннях. Особлива увага приділена перспективними у цьому напрямі алгоритмам, що використовують згорткові та рекурентні нейронні мережі з довгою короткочасною пам’яттю.

1 ОГЛЯД І АНАЛІЗ СУЧАСНОГО СТАНУ ЗАДАЧІ РОЗПІЗНАВАННЯ ТЕКСТУ НА ЗОБРАЖЕННЯХ

Розпізнавання – це здатність виявляти в потоці інформації, що надходить певні об’єкти, закономірності, явища. Воно може здійснюватися на основі слухової, зорової, тактильної інформації. Можливість розпізнавання спирається на схожість однотипних об’єктів. Незважаючи на те, що всі предмети і ситуації унікальні в строгому сенсі, між деякими з них завжди можна знайти подібності за тією чи іншою ознакою. Звідси виникає поняття класифікації – розбиття всієї множини об’єктів на підмножини – класи, елементи яких мають деякі схожі властивості, що відрізняють їх від елементів інших класів. І, таким чином, завданням розпізнавання є віднесення розглянутих об’єктів або явищ по їх опису до потрібних класів. Тобто поняття розпізнавання можна розширити, якщо говорити про виявлення об’єктів в потоці не тільки чуттєвої, а й будь-якої іншої інформації [2].

1.1 Актуальність задачі розпізнавання образів

Розвиток і поширення компютерної обробки інформації привели до виникнення в середині XX століття потреб в технологіях, що дозволяють машинам здійснювати розпізнавання оброблюваної ними інформації. Розробка методів машинного розпізнавання дозволяє розширити коло виконуваних комп’ютерами завдань і зробити машинну обробку інформації більш інтелектуальною [3]. Прикладами сфер застосування розпізнавання можуть служити системи розпізнавання тексту, машинний зір, розпізнавання мови, відбитків пальців та інше. Незважаючи на те, що деякі з цих завдань вирішуються людиною на підсвідомому рівні з великою швидкістю, до теперішнього часу ще не створено комп’ютерних програм, що вирішують такі задачі в настільки ж загальному вигляді. Існуючі системи призначені для роботи лише в спеціальних

випадках зі строго обмеженою сферою застосування. У цій роботі увага приділена розпізнаванню текстової інформації на основі зорових образів (зображень).

Дослідження раннього виявлення та розпізнавання тексту було природним продовженням аналізу документів та розпізнавання, переходячи від сканованих зображень сторінки до зображень, знятих на камеру, зосереджуючись на базовій попередній обробці, виявленні образів та технології OCR. Останнім часом застосування складних технологій комп'ютерного зору та методів навчання стало результатом усвідомлення того, що проблеми не піддаються послідовній серії незалежних рішень. Тенденція полягає в інтеграції завдань виявлення та розпізнавання в систему розпізнавання тексту «від кінця до кінця» [4].

У минулі роки дослідники широко досліджували графічний накладений текст у відео як спосіб індексувати відеовміст. Текст сцени, особливо текст відео-сцени, вважається складнішим викликом, але з ним було виконано порівняно мало роботи. Такі підходи, як правило, випливають із передових методів машинного навчання та оптимізації, включаючи непідконтрольне функціональне навчання, згорткові нейронні мережі (CNN) [5], моделі, що деформуються на базі деталей (DPM) та умовні випадкові поля (CRF) [6].

Графічний текст і текст сцени вважаються двома основними класами тексту, де перший відноситься до тексту машинного друку, накладеного графічно, а другий – до тексту на об'єктах, знятих у природньому середовищі. Графічний текст, як правило, друкується машинно, зустрічається у підписах, субтитрах та примітках у відео та нерозривних зображеннях в Інтернеті та в електронній пошті. Текст сцени, однак, включає текст на знаках, упаковках та одязі в природних сценах і рукописний матеріал. Останні дослідження зосереджуються на тексті сцени [7].

Розпізнавання структурованих символів з різних зображень забезпечує вирішення низки наукових і прикладних задач при ідентифікації об'єктів різної природи. Сучасні методи розпізнавання символів використовуються для вирішення як типових задач, наприклад розпізнавання друкованого тексту з

документів, так і спеціалізованих завдань, орієнтованих на розпізнавання символьної інформації, нанесеної на поверхню різних об’єктів [8]. В даний час існує досить велика кількість програм, призначених для розпізнавання тексту. Кожна з цих програм пропонує свою реалізацію рішення задачі обробки і розпізнавання зображень. В основному ці програми є комерційними, тому методи вирішення завдань, закладені в них, відомі тільки їх розробникам, і практично неможливо визначити для яких завдань вони підходять і які завдання їм не під силу. Крім того, більшість програм поставляється у вигляді виконуваних модулів, що робить неможливими аналіз працездатності програм, якості їх роботи та модифікацію використовуваних ними математичних моделей і алгоритмів.

Таким чином, в даний час актуальним є дослідження алгоритмів розпізнавання тексту, що приведе до створення програмного забезпечення для розпізнавання текстів з відкритим кодом на основі ефективних математичних методів і алгоритмів [9]. Відкритість коду подібної системи, за умови ретельного опрацювання її структури, дозволить не тільки вносити зміни і поліпшення в методи і алгоритми для вирішення задачі розпізнавання, але і вдосконалювати використовувані при її вирішенні математичні моделі.

1.2 Огляд існуючих засобів вирішення задачі

Завдання виявлення і розпізнавання друкованого, рукописного і тексту сцени широко поширене і представлена різними програмними продуктами: ABBYY FineReader, Amazon Rekognition, Clarifai, Google Cloud Vision, Microsoft Cognitive Services. Усі вони є пропрієтарними, тобто мають закритий сирцевий код і алгоритми та поставляються за підпискою або платою за використання. ПЗ ABBYY FineReader націлене на розпізнавання друкованого тексту з мінімальними перешкодами та може працювати без зв’язку з центральним сервером. Amazon Rekognition слугує для розпізнавання довільних образів та має модуль Amazon Textract, який спеціалізується на розпізнаванні текстів будь-якої

складності, але працює лише у режимі клієнт-сервер. Microsoft Cognitive Services Read API, Google Cloud Vision OCR працюють за схожим принципом. Clarifai може виявляти текст, але не розпізнає його контент. Описані продукти – це готові рішення, що можна використовувати завдяки клієнтському API, але вони не можуть бути налаштовані або модифіковані для покращення якості або вирішення конкретних задач. Якість розпізнавання цих систем сильно залежить від вхідних даних, але вона ще не є достатньою для промислового використання у загальному випадку.

На даний момент немає відкритих систем розпізнавання, що опрацьовують усі необхідні задачі та роблять це на відносно якісному рівні. Проте роботи у цьому напрямі ведуться – ентузіасти та великі компанії розуміють що зі збільшенням складності систем, жодна з компаній не може охопити повний цикл розробки власноруч. Тому створюються організації, як Computer Vision Foundation, метою яких є кооперація та розробка відкритих компонентів, які можуть бути використані для створення робочих систем. Великим здобуттям такого підходу є ціла низка відкритих проектів: OpenCV, TensorFlow, Tesseract. Вони можуть вільно використовуватися як в академічній, так і в комерційній діяльності та модифікуватися для покращення певних результатів. Реалізація алгоритмів з бібліотек OpenCV та Tesseract використовується у даній роботі для створення нейронної мережі.

1.3 Постановка задачі атестаційної роботи

На підставі перерахованих проблем, у роботі поставлені такі завдання:

– аналіз алгоритмів розпізнавання об’єктів на зображенні та методів, що застосовуються при обробці зображень;

– аналіз та дослідження засобів машинного навчання для пошуку та розпізнавання тексту на зображеннях;

– розробка гібридної архітектури нейронних мереж для розпізнавання структурованих послідовностей символів;

– проведення експериментального аналізу алгоритмів, що використовують згорткові та рекурентні нейронні мережі з довгою короткочасною пам’яттю.

В процесі аналізу алгоритмів розпізнавання тексту на зображеннях слід вирішити наступну задачу. Для заданої множини зображень $X = X_L \cup X_T$, де X_L – навчальна вибірка, а X_T – тестова вибірка, де множина допустимих результатів Y, а цільова функція $F = X \to Y$, значення якої відомі лише на множині X_L. Потрібно побудувати функцію $F' = X \to Y$, що належить деякому сімейству, обраному з емпіричних міркувань, таку, що вона добре наближає функцію F не тільки на множині X_T, а й на усій вибірці X. Нехай $Q(\theta, X_L)$ – деяка функція, що відповідає тому, як сильно алгоритм помиляється на тестовій вибірці, тоді потрібно розробити модель, що мінімізує похибку розпізнавання для усіх елементів вибірки X за признакою θ, формула (2.1):

$$\hat{\theta} = \arg \min_{\theta \in \Theta} Q(\theta, X_l) \tag{1.1}$$

2 АНАЛІЗ ОСОБЛИВОСТЕЙ ПОШУКУ І РОЗПІЗНАВАННЯ ТЕКСТУ НА ЗОБРАЖЕННЯХ

2.1 Складності у вирішенні задачі розпізнавання тексту

Складність середовищ, різні способи отримання зображень та зміна текстового вмісту створюють різні проблеми.

Складність сцени. У природних умовах з’являються численні техногенні об’єкти, такі як будівлі, символи та картини, які мають схожі структури та зовнішній вигляд з текстом. Проблема складності сцени полягає в тому, що оточуюча сцена ускладнює відділення тексту від нетекстового зображення.

Нерівномірне освітлення (рис. 2.1). Під час зйомки зображень на вулиці, нерівномірне освітлення поширене через нерівномірність реакції сенсорних пристроїв. Нерівномірне освітлення вносить спотворення кольорів і погіршення зорових особливостей, а отже, вводить помилкове виявлення, сегментацію та помилкове розпізнавання у результати.

Рисунок 2.1 – Приклад зображення з нерівномірним освітленням

Розмивання та деградація. При різноманітних умовах створення зображень та без фокусування камер відбувається розфокусування та розмивання тексту на зображеннях. Процедури стиснення зображень і відео та декомпресії також погіршують якість тексту, зокрема, графічного тексту на відео. Типовий вплив розфокусування, розмивання та деградації полягає в тому, що вони зменшують різкість символів, що ускладнює основні завдання, такі як сегментація.

Співвідношення сторін. Такий текст, як дорожні знаки, може бути коротким, тоді як інший текст, наприклад підписи у відео, може бути набагато довшим. Іншими словами, текст має різні співвідношення сторін. Для виявлення тексту необхідно враховувати процедуру пошуку щодо місця розташування, масштабу та довжини, що задає певну складність для обчислювальної техніки.

Спотворення. Перспективне спотворення виникає, коли оптична вісь камери не перпендикулярна до площини тексту (рис. 2.2). Межі тексту втрачають прямокутні форми, а символи спотворюються, зменшуючи продуктивність моделей розпізнавання, навчених на неспотворених зразках та надають значно більшу кількість варіантів напису кожного символу, що підвищує обчислювальну складність.

Рисунок 2.2 – Приклад зображення з спотвореною перспективою

Шрифти. Символи різних шрифтів мають великі варіації та утворюють багато візерунків, що ускладнює виконання точного розпізнавання, коли число класів символів велике.

Багатомовні середовища. Хоча мови з латинським алфавітом та алфавітом на основі кирилиці мають десятки символів, такі мови, як китайська, японська та корейська, мають тисячі класів символів. Арабська має зв’язані символи, які змінюють форму відповідно до контексту. Хінді поєднує букви алфавіту в тисячі форм, які представляють склади. У багатомовних середовищах OCR у відсканованих документах залишається дослідницькою проблемою, тоді як розпізнавання тексту на зображеннях з середовища є значно складнішим.

2.2 Фундаментальні задачі пошуку тексту на зображеннях

2.2.1 Локалізація (знаходження) тексту

Метою локалізації тексту є точне знаходження текстових компонентів, а також групування їх у текстові регіони з якомога меншим фоном.

Для локалізації тексту аналіз з'єднаних компонент (connected component analysis, CCA) [10] та класифікація зсувних вікон (sliding window classification) є двома широко використовуваними методами.

Аналіз з'єднаних компонент можна розглядати як графічний алгоритм, де підмножини з'єднаних компонентів однозначно маркуються на основі евристики щодо консенсусу ознак, тобто схожості кольорів та просторового розташування. У реалізаціях CCA часто використовуються методи розпізнавання синтаксичних візерунків для аналізу просторових ознак та визначення текстових областей. Враховуючи складність точного вирахування синтаксичних правил, можливий метод полягає у виконанні CCA зі статистичними моделями, наприклад, використовуючи класифікатор AdaBoost [11] для парних просторових функцій для навчання моделей CCA. Використання статистичних моделей суттєво підвищує адаптивність методу.

У методі класифікації зсувних вікон масштабовані частини зображень, які класифікуються, додатково групуються в текстові області за допомогою морфологічних операцій, методів умовного випадкогово поля (conditional random field, CRF) або графових методів. Перевага цього методу полягає в простоті та адаптивній архітектурі тренінгу-виявлення. Тим не менш, цей метод важко обчислювати, коли використовуються складні методи класифікації і необхідно класифікувати велику кількість «вікон».

Для локалізації тексту традиційно використовувались такі характеристики зображення, як кольори, границі та текстури. Текст зазвичай друкується в одному кольорі і має контраст з фоном. За цим припущенням кольорові функції можна використовувати для локалізації тексту. Текстові компоненти зазвичай

мають значну кольорову контрастність з фоном і мають тенденцію до формування однорідних кольорових областей. Локалізація тексту на основі кольорів працює ефективно, хоча чутлива до різнокольорових символів та нерівномірного освітлення, що може серйозно погіршити кольорові особливості. Щоб адаптуватись до варіацій кольорів, кольорові особливості витягуються в перетворених або комбінованих кольорових просторах, наприклад у градаціях сірого кольору. Таким чином, пікселі з великими та симетричними значеннями градієнта можна вважати текстовими компонентами. У порівнянні з кольоровими особливостями, градієнтні та граничні характеристики менш чутливі до нерівномірного освітлення та багатобарвних символів. Однак вони часто мають труднощі при розрізненні текстових компонентів зі складними фонами, що мають сильний градієнт. Алгоритм максимально стійких екстремальних областей (maximally stable extremal regions, MSER), який адаптивно виявляє стійкі кольорові області, забезпечує рішення для локалізації тексту.

Коли символи щільні, текст можна розглядати як текстуру. Для локалізації тексту використовуються функції текстури, включаючи перетворення Фур'є, дискретне косинусне перетворення (DCT), вейвлет [12], локальні бінарні шаблони (LBP) та гістограму орієнтованих градієнтів (HOG). Такі функції, як правило, поєднуються з методом класифікації зсувного вікна. Функції текстури ефективні для виявлення щільних символів, хоча вони можуть не виявляти рідкісні символи, знаки в зображеннях сцени, у яких відсутні значні властивості текстури.

Перетворення ширини обведення (SWT) [13] локальний оператор зображення, який обчислює ширину найбільш ймовірного обведення, що містить піксель. SWT виводить карту, де кожному елементу відповідає значення ширини обведення пікселя. Особливості на основі обведення виявляються конкурентоспроможними для локалізації тексту сцени з високою роздільною здатністю, зокрема, коли вони поєднуються з відповідними методами навчання, або вдосконалюються за допомогою інших сигналів, таких як дисперсія

орієнтації на край (EOV) [14] та пари протилежних країв (OEP) або поєднується з просторово-часовим аналізом.

Текстові об’єкти, такі як відеозаписи, мають щільні символи та сильні градієнти, а інші можуть мати рідкісні символи, але колір відрізняє їх від оточення. Для підвищення надійності застосовуються гібридні функції для пошуку тексту.

2.2.2 Перевірка тексту

Локалізація тексту часто вводить помилкові дані, оскільки частина компонентів може не містити достатньої інформації для класифікації. Після локалізації тексту доступні цілісні особливості текстових областей для точної класифікації та верифікації. Методи, що ґрунтуються на знаннях про колір, розмір та простір, а також проекції використовуються для перевірки тексту. Співвідношення сторін використовується для перевірки локалізованих текстових областей. Синтаксичні правила використовуються для підрахунку ребер, горизонтального профілю, висоти та ширини з’єднаних компонентів. Також можуть використовуватися пропорції ширини та довжини мінімальних обмежуючих прямокутників (MBR), пропорції тексту та фонових пікселів.

Для вирізнення тексту, необхідною умовою якої є те, що особливості, отримані з областей зображень із різними співвідношеннями сторін, нормалізуються до однакової розмірності. Одним із способів отримання нормованих ознак є отримання глобальних ознак, які не залежать від співвідношення сторін у регіоні Особливості краю, градієнта та текстури поєднуються та навчаються з багатошаровим перцептроном (MLP) для перевірки тексту. Особливості інтенсивності та співвідношення сторін граничної рамки витягуються для представлення з’єднаних компонентів. Такі особливості подаються в алгоритм k-середніх для класифікації.

2.2.3 Сегментація тексту

Перед тим, як виявлені текстові області будуть розпізнані модулем OCR, певні підходи використовують алгоритми бінаризації, сегментації текстових рядків та алгоритмів сегментації символів для отримання точно обмежених символів. Сегментація визначена як одна з найскладніших проблем [15], і останні підходи часто інтегрують етап сегментації з кроком розпізнавання або використовують відповідність слів, щоб уникнути проблеми сегментації.

Бінаризація тексту слугує для вирізнення текстових пікселів та видалення фонових пікселів. Для цього використовуються алгоритми, пов’язані з адаптивним порогом, моделями ймовірностей та кластеризацією. Адаптивне порогове значення наближається до сегмента тексту відповідно до їх відповідних локальних особливостей, і таким чином є адаптивним до фонів. Тим не менш, важко вибрати надійне порогове значення для деградованого тексту, де пікселі тексту на межах часто поєднуються з фоном. У цьому випадку можна застосувати моделі суміші Гаусса, враховуючи той контекст, що для побудови моделей відбирається значна кількість пікселів переднього плану.

Функція сегментації текстових рядків полягає в перетворенні області з декількох рядків тексту в кілька підрегіонів окремих текстових рядків. Для горизонтального тексту використовується аналіз проекційного профілю текстових компонентів, однак для перекрученого або перспективного спотвореного тексту цей метод не підходить, а потрібно спочатку виділити перспективу.

Сегментація символів розділяє текстову область на декілька областей окремих символів. Аналіз вертикальної проекції був раннім методом сегментації символів. Однак часто важко визначити оптимальний поріг проекції, коли існує деградація або дотик символів. З високим порогом істинні сегментації можуть бути пропущені, хоча при низькому порозі може бути виявлено багато помилкових сегментацій [16]. Постійно розвиваються адаптивні методи, включаючи адаптивну морфологічну операцію, методи кластеризації та

оптимізації. Популярним є двопрохідний алгоритм пошуку, пошук вперед локалізує потенційні скорочення, а назад – видаляє помилкове розділення символів.

2.2.4 Розпізнавання тексту

Розпізнавання тексту перетворює області зображень у текст. В останніх дослідженнях розпізнавання слова займає центральне місце в розпізнаванні тексту, оскільки слова добре сформульовані за статистичними моделями. Розпізнати деградований текст на рівні символів важко через брак контексту.

Для розпізнавання символів одного шрифту часто використовуються загальні риси, наприклад, функції Габора та прості класифікатори, такі як лінійний дискримінантний аналіз (LDA). Однак при наявності декількох шрифтів або спотворених символів різноманітність всередині класу ускладнює моделювання символів одного класу. Одне рішення – мати визначений класифікатор для кожного з них. Інші рішення включають вирівнювання символів за допомогою непідконтрольного чи репрезентативного навчання, дискримінаційне об’єднання функцій, алгоритми випрямлення зображень або деформуючі моделі. Оцінка афінного перетворення використовується для обробки різних спотворень.

Для деградованого або спотвореного тексту, звично, щоб модель розпізнавання присвоювала різні мітки однаковим символам. Особливо часто це відбувається з урахуванням спотворень або відсутності даних про навчання для певних шрифтів. У цьому випадку сегментація символів та розпізнавання символів можуть бути інтегровані зі словниками. Для забезпечення точності розпізнавання використовуються моделі мови вищого порядку (n-грам). Використання великого словника не тільки полегшує виявлення слабких символів, але й примушує розпізнавати слова, що не мають словника, такі як назви підприємств та назви вулиць.

2.3 Методи розпізнавання тексту

Розпізнавання тексту – це переважно завдання в два кроки. По-перше, потрібно виявити наявність текстової інформації на зображенні, нехай вона буде щільною (як у друкованому документі) або рідкою (як текст у оточуючому середовищі). Виявивши рядок або слова на зображенні, потрібно вибрати з великого набору рішень, які, як правило, виходять з таких основних підходів [17]:

– Класичні методи комп’ютерного зору.

– Спеціалізоване глибоке навчання.

2.3.1 Класичні методи комп’ютерного зору

Класичні підходи мають такий загальний алгоритм:

а) Застосування фільтрів, щоб виділити символи з фону.

б) Застосування контурного виявлення для відділення символів один від одного.

в) Застосування класифікації зображень для ідентифікації символів допомогою пошуку по шаблону.

Виявлення контуру символу є досить складним для узагальнення. Воно вимагає багато ручної тонкої настройки, тому стає нездійсненним у більшості проблем. Проблема ручної настройки параметрів у тому що, можна зменшити одні помилки, але викликати інші і відшукати їх залежність власноруч майже неможливо.

Шаблонні методи розпізнавання символів. Першим етапом роботи шаблонного методу є перетворення сканованих зображень в растрове. Далі проводиться його порівняння з усіма наявними в базі системи шаблонами. Найбільш підходящим шаблоном вважається той, у якого буде найменша кількість точок, відмінних від досліджуваного зображення. Шаблон для кожного класу зазвичай отримують шляхом усереднення зображення символів навчальної

вибірки. У цих методів досить висока точність розпізнавання дефектних символів (склеєних або розірваних). Недолік даного методу – неможливість розпізнати символ, що хоч трохи відрізняється від закладеного в систему (розміром, нахилом або шрифтом). Алгоритм, заснований на шаблонному методі, повинен заздалегідь знати шрифт, який йому представляють для розпізнавання. При існуючій різноманітності друкованої продукції в процесі навчання неможливо охопити всі шрифти і їх модифікації. Іншими словами, цей фактор обмежує універсальність таких методів.

Використання шаблонного методу розпізнавання символів підходить в тому випадку, якщо потрібно обробити документ типографської якості (досить великий шрифт, відсутність погано надрукованих символів або виправлень). Використання даного методу в цьому випадку дозволяє отримати достатній точно швидкі алгоритми розпізнавання.

Структурні, або топологічні, методи розпізнавання зберігають інформацію не про точне написання символу, а про його топологію. Іншими словами, еталон містить інформацію про взаємне розміщенні окремих складових частин символу. Розпізнається символ піддається процедурі скелетизації (рис 2.3). При цьому стає неважливим розмір символу, що розпізнається і шрифт, яким вона надрукована. Але основною проблемою структурних методів розпізнавання є ідентифікація знаків, що мають дефекти (наприклад, розрив лінії або злиття сусідніх ліній).

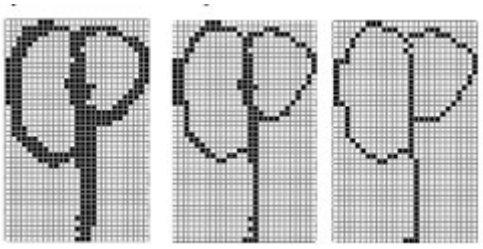

Рисунок 2.3 – Скелетизація образу символа

Методи векторизації базуються на тому, що зображенню ставиться у відповідність N-мірний вектор ознак (рис 2.4). Розпізнавання полягає в порівнянні його з набором еталонних векторів тієї ж розмірності. Завдання розпізнавання, прийняття рішення про приналежність образу того чи іншого класу, на підставі аналізу обчислених ознак, має цілий ряд строгих математичних рішень в рамках детерміністичного та імовірнісного підходів. У системах розпізнавання символів найчастіше використовується класифікація, заснована на підрахунку евклідової відстані між вектором ознак символу і векторами ознак еталонного опису. Тип і кількість ознак визначають якість розпізнавання. Формування вектора проводиться під час аналізу попередньо підготовленого зображення. Даний процес називають витяганням ознак.

Рисунок 2.4 – Схема отримання вектора ознак зображення

Основні переваги структурних методів та методів векторизації – простота реалізації, висока швидкодія. Найбільш серйозний недолік цих методів – нестійкість до дефектів зображення. Крім того, такі методи мають інший серйозний недолік – на етапі отримання ознак відбувається необоротна втрата частини інформації про символи. Витяг ознак ведеться незалежно, тому інформація про взаємне розташування елементів символу втрачається.

Необхідно відзначити, що в сучасних системах розпізнавання, алгоритми на основі класичних методів не використовуються самі по собі, а комбінуються з іншими методами розпізнавання, основаними на зовсім інших принципах, в першу чергу – комбінації з нейронними мережами.

3 АЛГОРИТМИ ПОШУКУ І РОЗПІЗНАВАННЯ ТЕКСТУ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

3.1 Ідентифікація об’єктів на зображеннях з використанням згорткової нейронної мережі

Вхідні дані згорткової мережі (рис 3.1) представляють із себе кольорові зображення типу JPEG, з визначеним розміром, наприклад, 48x48 пікселів. Якщо розмір буде надто великим, то обчислювальна складність підвищується, відповідно зменшується швидкість обробки. Якщо розмір зображення буде малий, то не будуть виділені ключові ознаки. Визначення розмірів у заданій задачі вирішується методом підбору. Вхідний шар враховує двомірну топологію зображень і складається з декількох матриць (карт), матриця може бути одна, у тому випадку, якщо зображення представлено у відтінках сірого, інакше їх 3, де кожна карта відповідає зображенню з конкретним каналом (червоним, синім і зеленим). Вхідні дані кожного конкретного значення пікселя нормалізуються в діапазоні від 0 до 1, за формулою (3.1).

$$f(p, min, max) = \frac{p - min}{\max - min}, \tag{3.1}$$

де f – функція нормалізації; p – значення кольору пікселя від 0 до 255; min – мінімальне значення пікселя – 0; max – максимальне значення пікселя – 255.

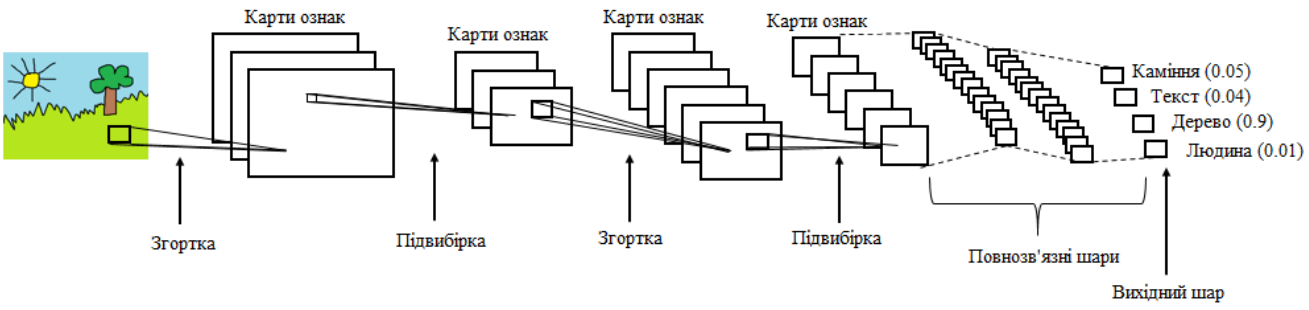

Рисунок 3.1 –Топологія згорткової нейронної мережі

Згортковий шар представляє із себе набір матриць ознак, у кожної карти є синаптичне ядро. Кількість матриць визначається експериментальним шляхом, якщо взяти велику кількість матриць, то підвищиться якість розпізнавання, але збільшиться обчислювальна складність. Зазвичай, на цьому етапі пропонується використовувати по дві матриці на кожну матрицю попереднього шару, тобто 6 загалом. Розмір у всіх матриць згорткового шару однаковий і визначається за формулою (3.2).

$$(w, h) = (mW - kW + 1, mH - kH + 1)\,, \tag{3.2}$$

де (*w, h*) – обчислюваний розмір згорктової матриці; *mW* – ширина попередньої матриці; *mH* – висота попередньої матриці; *kW* – ширина ядра; *kH* – висота ядра.

Ядро представляє із себе фільтр або вікно, яке ковзає за всіма областями попередньої матриці та знаходить конкретні визначні об’єкти. Розмір ядра зазвичай беруть в межах від 3x3 до 7x7 пікселів. Якщо розмір ядра маленький, то він не виділяє ознаки, якщо занадто великий, то збільшується кількість зв’язків між нейронами. Розмір ядра вибирається таким чином, що розмір матриці згорткового шару був парним, це дозволяє не втрачати інформацію при зменшенні розмірів у підвиборочному шарі. Ядро представляє єдину систему розділених вагових або синапсів, це одна з головних особливостей згорткової мережі. У звичайних багатослойних мережах дуже багато зв’язків між нейронами, що уповільнює процес детектування. У згорткових мережах – навпаки, загальні ваги дозволяють скоротити число зв’язків і дозволяють знаходити одну і ту ж ознаку по усій області зображення. Спочатку значення кожної карти згорткового шару дорівнюють 0. Значення вагів ядер задаються випадковим чином в області від -0.5 до 0.5. Ядро ковзає по попередній карті і робить операцію згортки (рис. 3.2), яка часто використовується для обробки зображень, за формулою (3.3).

$$(f * g)[m, n] = \sum_{k,l} f[m - k, n - l] * g[k, l]\,, \tag{3.3}$$

де f – початкова матриця зображення; g – ядро згортки.

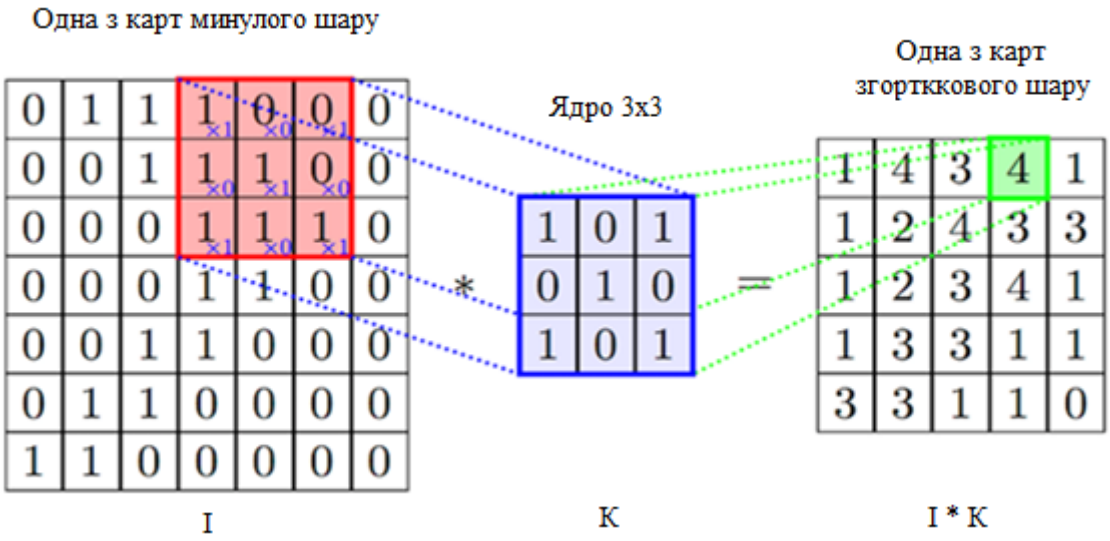

Рисунок 3.2 – Операція згортки і отримання значень згорткової матриці

При цьому в залежності від методу обробки країв початкової матриці результат може бути меншим за зображення, такого ж розміру або більше розмірів, відповідно до рис. 3.3.

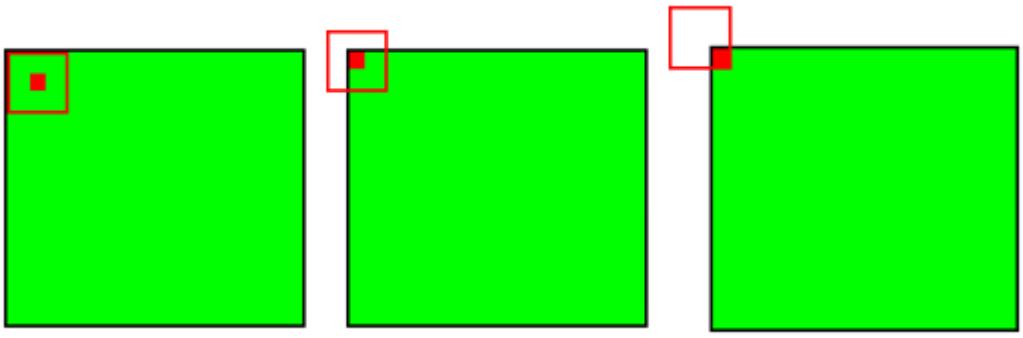

Рисунок 3.3 – Три види згортки початкової матриці

Згортковий шар можна описати за формулою (3.4):

$$x^l = f(x^{l-1} * k^l + b^l)\,, \tag{3.4}$$

де x^l вихід шару l; f – функція активації; b^l – коефіцієнт здвигу шару l; * – операція згортки входу x з ядром k.

При цьому за рахунок крайових ефектів розмір початкових матриць зменшується за формулою (3.5):

$$x_j^l = f\left(\sum_i x_i^{l-1} * k_j^l + b_j^l\right), \tag{3.5}$$

де x^l_j матриця ознак j (вихід шару l); f – функція активації; b^l – коефіцієнт здвигу шару l для матриці ознак j; k^l_j – ядро згортки матриці, шару l; * – операція згортки входу x з ядром k.

Підвибірковий шар також, як і згортковий має матриці, але їх кількість співпадає з попереднім (згортковим) шаром, їх 6. Мета шару – зменшення розмірності карт попереднього шару. Якщо на попередній операції згортки вже були виявлені деякі ознаки, то для подальшої обробки настільки детальне зображення вже не потрібне, і воно ущільнюється до менш детального. До того ж фільтрація вже непотрібних деталей допомагає не перевчати мережу. В процесі сканування ядром підвибіркового шару (фільтром) карти попереднього шару, скануюче ядро не перетинається на відміну від згортального шару. Зазвичай, кожна карта має ядро розміром 2x2, що дозволяє зменшити попередні карти згортального шару в 2 рази. Уся карта ознак розділяється на осередки 2x2 елементи, з яких вибираються максимальні за значенням. Зазвичай в підвибірковому шарі застосовується функція активації RelU. Операція підвибірки (чи MaxPooling – вибір максимального) відповідно до рис. 3.4.
Шар може бути описаний формулою (3.6):

$$x^l = f(a^l * subsample(x^{l-1}) + b^l), \tag{3.6}$$

де x^l – вихід шару l; f – функція активації; a^l, b^l – коефіцієнти здвигу шару l; *subsample* – операція вибірки локальних максимальних значень.

Повнозв’язний шар – останній з типів шарів, це шар звичайного багатошарового перцептрона (рис 3.5). Мета шару – класифікація, моделює складну нелінійну функцію, оптимізуючи яку, покращується якість розпізнавання.

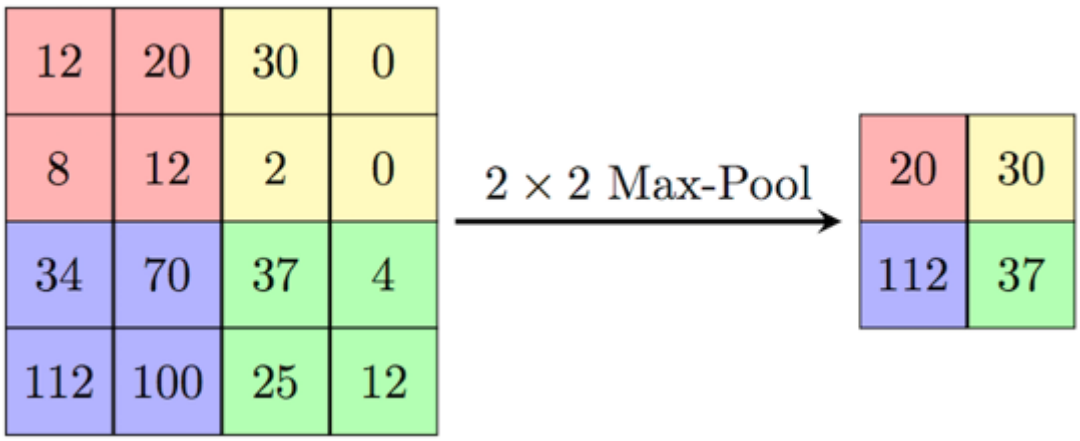

Рисунок 3.4 – Формування нової карти підвибіркового шару на основі попередньої матриці згортального шару

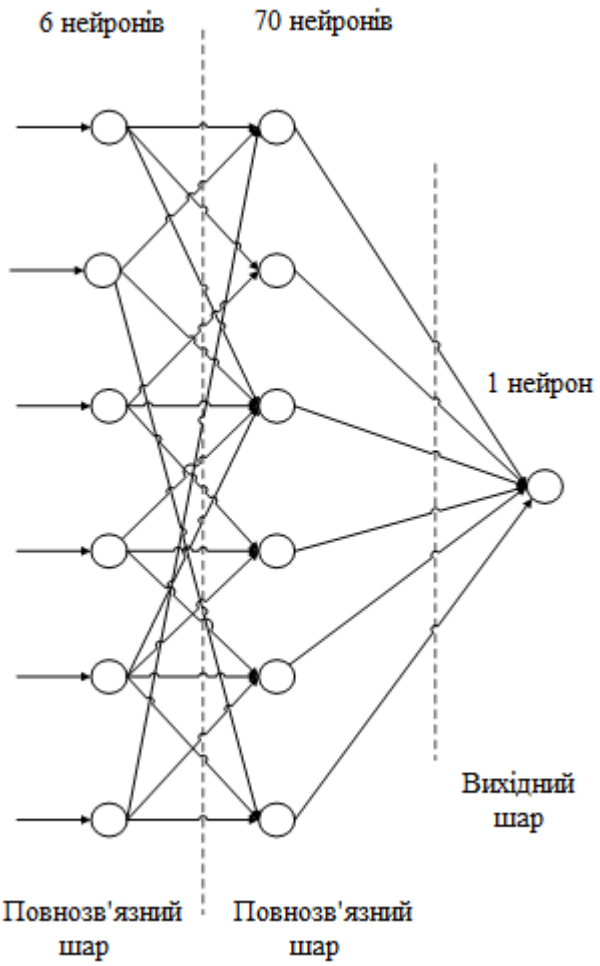

Рисунок 3.5 – Структура повнозв’язного шару

Нейрони кожної матриці попереднього підвибіркового шару пов’язані з одним нейроном прихованого шару. Таким чином число нейронів прихованого

шару дорівнює числу карт підвибіркового шару, але зв’язки можуть бути не обов’язково такими, наприклад, тільки частина нейронів якої-небудь з карт підвибіркового шару може бути пов’язана з першим нейроном прихованого шару, а частина, що залишилася, з другим, або усі нейрони першої карти пов’язані з нейронами 1 і 2 прихованого шару. Обчислення значень нейрона можна описати формулою (3.7):

$$x_j^l = f\left(\sum_i x_i^{l-1} * w_{i,j}^{l-1} + b_j^{l-1}\right), \tag{3.7}$$

де x^l_j матриця ознак j (вихід шару l); f – функція активації; b^l – коефіцієнт здвигу шару l для карти ознак j; $w^l_{i,j}$ – матриця вагових коефіцієнтів шару l.

Вихідний шар пов’язаний з усіма нейронами попереднього шару. Кількість нейронів відповідає кількості розпізнаваних класів, тобто 2, у випадку пошуку конкретного об’єкта – наприклад, текст і не текст. Але для зменшення кількості зв’язків і обчислень для бінарного випадку можна використати один нейрон і при використанні функції активації, вихід нейрона зі значенням 1 означає приналежність до певного класу, навпроти вихід нейрона зі значенням -1 означає, що елемент не належить до цього класу.

3.1.2 Вибір функції активації

Одним з етапів розробки нейронної мережі є вибір функції активації нейронів. Вид функції активації багато в чому визначає функціональні можливості нейронної мережі і метод навчання цієї мережі. Класичний алгоритм зворотного поширення помилки добре працює на двошарових і тришарових нейронних мережах, але при подальшому збільшенні глибини починає зазнавати проблем. Одна з причин – так зване загасання градієнтів. У міру поширення помилки від вихідного шару до вхідного на кожному шарі відбувається множення поточного результату на похідну функції активації. Похідна у традиційної сигмоїдної функції активації менше одиниці на усій області

визначення, тому після декількох шарів помилка стане близькою до нуля. Якщо ж, навпаки, функція активації має необмежену похідну (як, наприклад, гіперболічний тангенс), то може статися вибухове збільшення помилки у міру поширення, що приведе до низької стійкості процедури навчання.

Функція активації сигмоїда (рис 3.6). Ця функція відноситься до класу безперервних функцій і приймає на вході довільне дійсне число, а на виході дає дійсне число в інтервалі від 0 до 1. Зокрема, великі (по модулю) негативні числа перетворюються на нуль, а великі позитивні – в одиницю. Історично сигмоїда знаходила широке застосування, оскільки її вихід добре інтерпретується, як рівень активації нейрона: від відсутності активації (0) до повністю насиченої активації (1). Сигмоїда (sigmoid) виражається формулою (3.8):

$$f(s) = \frac{1}{1 + e^{-s}}. \tag{3.8}$$

де – *s* вхідне значення нейрона.

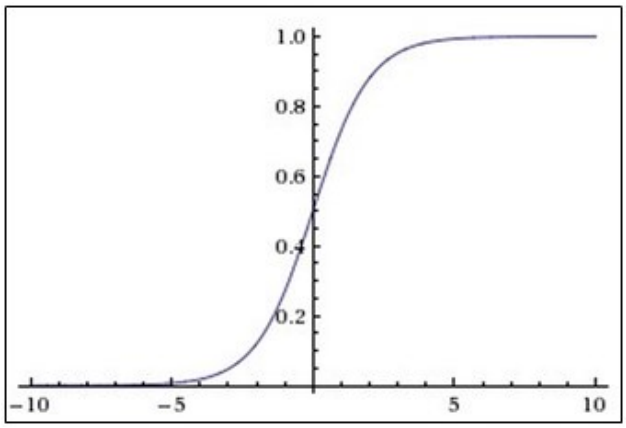

Рисунок 3.6 – Графік сигмоїдальної функції

Небажана властивість сигмоїди полягає в тому, що при насиченні функції з того або іншого боку (0 або 1), градієнт на цих ділянках стає близький до нуля. В процесі зворотного поширення помилки локальний градієнт множиться на загальний градієнт. Отже, якщо локальний градієнт дуже малий, він фактично

обнуляє загальний градієнт. В результаті, сигнал майже не проходить через нейрон до його вагів і рекурсивно до його даних. Крім того, слід бути дуже обережним при ініціалізації вагів сигмоїдних нейронів, щоб запобігти насиченню. Наприклад, якщо початкові ваги мають занадто великі значення, більшість нейронів перейдуть в стан насичення, внаслідок чого мережа погано навчатиметься. Сигмоїдальна функціяє: безперервною, монотонно зростаючою, диференціюється.

Функція активації гіперболічний тангенс (рис 3.7). Симетричні активаційні функції, типу гіперболічного тангенса забезпечують швидшу збіжність, ніж стандартна логістична функція; функція має безперервну першу похідну; функція має просту похідну, яка може бути вирахована через її значення, що дає економію обчислень.

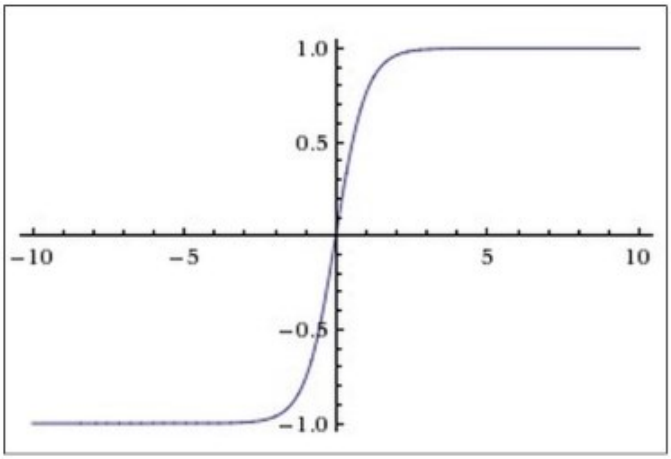

Рисунок 3.7 – Графік функції гіперболічного тангенса

Функція активації ReLU (рис 3.8). Нейронні мережі здатні наблизити скільки завгодно складну функцію, якщо в них досить шарів і функція активації є нелінійною. Функції активації як сигмоїда або тангенціальна є нелінійними, але призводять до проблем із загасанням або збільшенням градієнтів. Проте можна використати випрямлену лінійну функцію активації (rectified linear unit, ReLU), яка виражається формулою (3.9):

$$f(s) = \max(0, s) \tag{3.9}$$

де s – вхідне значення нейрона.

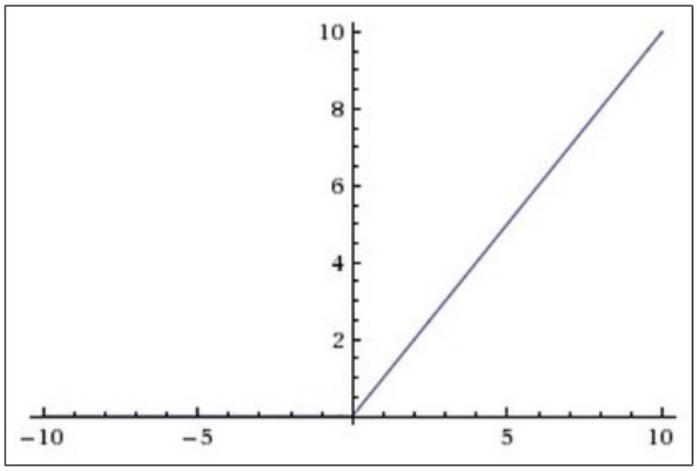

Рисунок 3.8 – Графік функції ReLU

Переваги використання ReLU: її похідна дорівнює або одиниці, або нулю, і тому не може статися розростання або загасання градієнтів, оскільки помноживши одиницю на дельту помилки ми отримаємо дельту помилки, якщо використати іншу функцію, наприклад, гіперболічний тангенс, то дельта помилки могла, або зменшитися, або зрости, або залишитися такою ж, тобто, похідна гіперболічного тангенса повертає число з різним знаком і величиною, що може сильно вплинути на загасання або розростання градієнта. Використання цієї функції призводить до проріджування вагів; обчислення сигмоїди і гіперболічного тангенса вимагає виконання складних для обчислення операцій, таких як піднесення до степеня, тоді як ReLU може бути реалізована за допомогою простого порогового перетворення матриці активацій в нулі; функція відсікає непотрібні деталі в каналі при негативному виході.

З недоліків можна відмітити, що ReLU не завжди досить надійна і в процесі навчання може виходити з ладу. Наприклад, великий градієнт, що проходить через ReLU, може привести до такого оновлення вагів, що цей нейрон ніколи більше не активується. Якщо це станеться, то, починаючи з цього моменту,

градієнт, що проходить через цей нейрон, завжди дорівнюватиме нулю. Відповідно, цей нейрон буде безповоротно виведений з ладу. Наприклад, при занадто великій швидкості навчання (learning rate), може виявитися, що велика кількість нейронів ReLU ніколи не активуються. Ця проблема вирішується за допомогою вибору належної швидкості навчання.

3.2 Розпізнавання тексту на зображеннях з використанням рекурентної нейронної мережі

Рекурентні нейронні мережі – вид нейронних мереж, архітектура яких складається з трьох шарів: вхідного, прихованого та вихідного. При цьому прихований шар має зворотній зв'язок сам на себе.

На відміну від нейронних мереж з прямим зв'язком, рекурентна нейронна мережа може приймати на вхід послідовність векторів $x = (x_1, x_2, \dots, x_T)$ в заданому порядку. Робота рекурентної нейронної мережі описується співвідношеннями (3.10) і (3.11).

$$h_t = \varphi(W h_{t-1} + W_x x_t + b_h), \tag{3.10}$$

$$y_t = \varphi\left(W_y h_t + b_y\right), \tag{3.11}$$

де φ – нелінійна функція; x_t – вхідний вектор за номером t; h_t – стан прихованого шару для входу x_t; y_t – вихід мережі для входу x_t; W, W_x, W_y – вагові коефіцієнти матриці нейронної мережі; b_h, b_y – вектори здвигу.

За рекурентним співвідношенням (3.10) є можливість урахування результатів перетворення нейронної мережею інформації на попередніх етапах для обробки вхідного вектора на наступному етапі функціонування мережі. На рис. 3.9 представлена схема роботи рекурентної нейронної мережі.

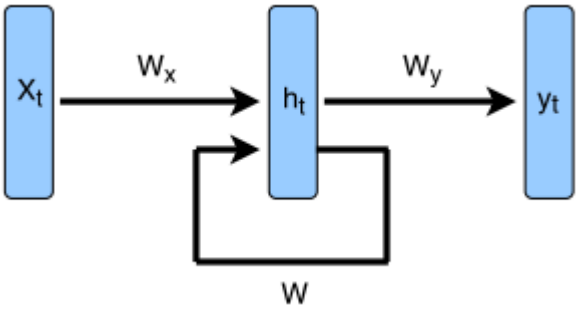

Рис. 3.9 – Схема роботи рекурентної нейронної мережі

Гнучкість рекурентних нейронних мереж дозволяє навчати їх генеративно. При генеративному навчанні, застосовуючи до виходу y_t нормувальну функцію g, отримаємо, що $g(y_t)$ – це розподіл на наступний елемент послідовності, що враховує знання про попередні елементи послідовності. Використання правила перемноження (general product rule) за формулами (3.12) і (3.13), дає можливість оцінити ймовірність всієї послідовності.

$$g(y_t) = p(x_{t+1}|x_1, ..., x_t) \tag{3.12}$$

$$p(x_1, ..., x_T) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)...p(x_T|x_1, ..., x_{T-1}) \tag{3.13}$$

3.2.1 Навчання рекурентної нейронної мережі

Для навчання рекурентної нейронної мережі вводиться функція витрат $\mathcal{L}_t(y_t, \hat{y}_t)$, що характеризує величину відхилення відповіді y_t нейронної мережі від правильної відповіді $\hat{y}_t$ в момент часу t. Повна функція витрат для одного об’єкта навчальної вибірки визначається за формулою (3.14).

$$\mathcal{L}(y, \hat{y}) = \sum_i \mathcal{L}_i(y_i, \hat{y}_i) \tag{3.14}$$

Рекурентна нейронна мережа, навчаючись, повинна мінімізувати функцію витрат за всією навчальної вибіркою. Мінімізація функції витрат зазвичай відбувається з допомогою методу стохастического градієнтного спаду. Для цього

слід виконати обчислення $\frac{d\mathcal{L}}{dW}$, $\frac{d\mathcal{L}}{dW_x}$, $\frac{d\mathcal{L}}{dW_y}$, за допомогою алгоритму зворотного поширення помилки крізь час (backpropagation through time). В даному алгоритмі розгортається граф обчислень у часі, як на рис. 3.10.

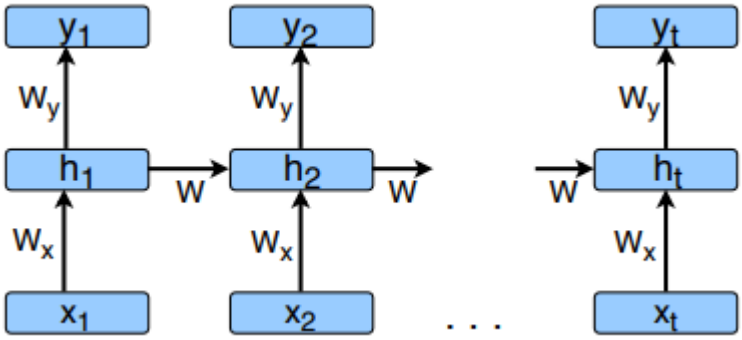

Рис 3.10 – Розгорнута схема роботи рекурентної нейронної мережі

З рис. 3.10 можна помітити, що вірні рівності за формулами (3.15), (3.16), (3.17):

$$\frac{\partial \mathcal{L}}{\partial W} = \sum_t \frac{\partial \mathcal{L}_t(y_t, \hat{y}_t)}{\partial W} \tag{3.15}$$

$$\frac{\partial \mathcal{L}_t}{\partial W} = \frac{\partial \mathcal{L}_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial W} \tag{3.16}$$

$$\frac{\partial h_t}{\partial W} = \sum_{k=0}^{t} \left(\prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial W} \tag{3.17}$$

Якщо $\forall_i$ виконується $\left\| \frac{dh_i}{dh_{i-1}} \right\|_2 > 1$, то виникає проблема «вибухового градієнта» (exploding gradients). Для боротьби з цією проблемою зазвичай використовують обрізання градієнта, коли його норма є вищою за визначений поріг (gradient clipping).

Якщо $\forall_i$ виконується $\left\| \frac{dh_i}{dh_{i-1}} \right\|_2 < 1$, то виникає проблема «загасання градієнта» (vanishing gradients). Чим далі елемент знаходиться від поточного, тим

менше його внесок в градієнт. В цьому випадку $\left\| \Pi_{i=k+1}^{t} \frac{dh_i}{dh_{i-1}} \right\|_2$ буде експоненціально прагнути до нуля зі зростанням $t-k-1$. Одним з методів боротьби з загасаючим градієнтом є зміна формули обчислення прихованого стану h_t (3.10). Найбільш популярними архітектурами нейронних мереж, що допомагають уникнути проблему затухаючого градієнта, є LSTM і GRU мережі.

3.2.2 LSTM мережі

Long Short-Term Memory (LTSM) або мережа з довготривало-короткочасної пам'яттю – особливий тип рекурентних нейронних мереж. На відміну від блоку простої рекуррентної мережі, який обчислює зважену суму вхідного сигналу і застосовує нелінійну функцію, кожен блок LSTM мережі має пам'ять c_t, яка зберігає інформацію, отриману в попередній момент часу. В LSTM мережі прихований шар h_t обчислюється за формулою (3.18), o_t – вихідний шлюз (output gate), який контролює обсяг даних, одержуваних з пам'яті, і обчислюється за формулою (3.19). В формулах символом $\odot$ позначається поелементний добуток векторів.

$$h_t = o_t \odot \tanh(c_t) \tag{3.18}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + v_o \odot c_t + b_o) \tag{3.19}$$

Пам'ять c_t, оновлюється за допомогою формул (3.20) і (3.21):

$$c_t = f_t \odot c_{t-1} + i_t \odot \widehat{c_t} \tag{3.20}$$

$$\widehat{c_t} = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{3.21}$$

де f_t – шлюз забування (forget gate), який контролює ступінь збереження даних з пам'яті минулого блоку, а i_t – вхідний шлюз (input gate), що визначає вплив проміжної пам'яті $\widehat{c_t}$ на результат.

Значення f_t та i_t відповідно розраховуються за формулами (3.22) і (3.23) відповідно.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + v_f \odot c_t + b_f) \tag{3.22}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + v_i \odot c_t + b_i) \tag{3.23}$$

В перерахованих формулах x_t – вхідний вектор поточного блоку LSTM мережі; h_{t-1} – вихід попереднього блоку LSTM мережі; W_f, W_c, W_i, W_o, U_f, U_c, U_i, U_o – матриці, придатні для налаштування нейронної мережі; b_f, b_c, b_i, b_o, v_f, v_i, v_o – вектори, придатні для налаштування нейронної мережею.

3.2.3 GRU мережі

Gated Recurrent Unit (GRU) або мережа з циклічно повторюваним блоком. Подібно до LSTM, GRU складається з блоків, які моделюють інформацію всередині себе. Однак, без наявності окремих осередків пам'яті. GRU використовує менше операцій для обчислення, ніж LSTM. В GRU мережі прихований шар h_t обчислюється за формулою (3.24), на основі проміжного значення $\widehat{h_t}$ (3.25) і шлюзу оновлення z_t (update gate) (3.26).

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widehat{h_t} \tag{3.24}$$

$$\widehat{h_t} = tanh(W x_t + U(r_t \odot h_{t-1})) \tag{3.25}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{3.26}$$

Шлюз скидання стану r_t (reset gate) контролює ступінь збереження даних з минулого блоку в проміжному значенні $\widehat{h_t}$ і обчислюється за формулою (3.27):

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{3.27}$$

3.2.4 Схеми роботи рекурентної нейронної мережі

Залежно від того як сформувати вхід і вихід мережі, можна різними способами задати схему її роботи (рис. 3.11).

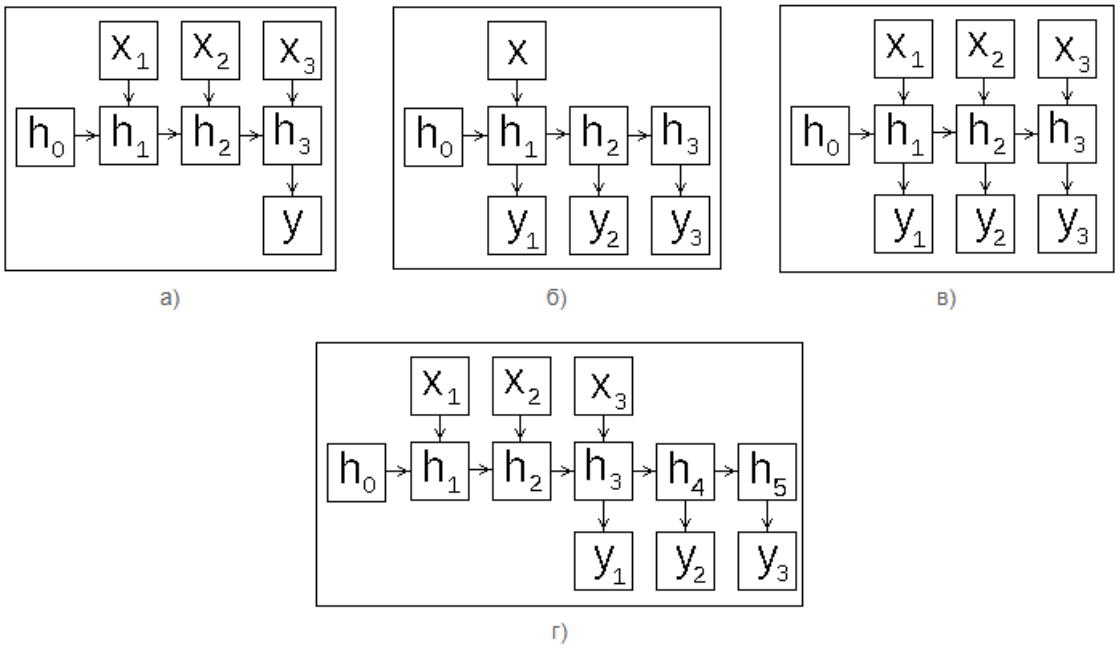

Рисунок 3.11 – Схеми організації рекурентної нейронної мережі

Можливі способи організації роботи рекурентної мережі:

а) «багато до одного» (рис. 3.11а) – прихований шар послідовно змінює свій стан, з його кінцевого стану обчислюється вихід мережі, цю схему можна використовувати для класифікації текстів;

б) «один до багатьох» (рис. 3.11б) – прихований шар ініціалізується одним входом, з ланцюжка його наступних станів генеруються виходи мережі, цю схему можна використовувати для анотування зображень;

в) «багато до багатьох» (рис. 3.11в) – на кожен вхід мережа видає вихід, який залежить від попередніх входів, цю схему можна використовувати для класифікації відео та розпізнавання послідовностей;

а) «багато до багатьох» (рис. 3.11г) – прихований шар послідовно змінює свій стан, його кінцевий стан служить ініціалізацією для видачі ланцюжка результатів, цю схему можна використовувати для створення систем машинного перекладу і чат-ботів.

4 ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ РОЗПІЗНАВАННЯ ТЕКСТУ

4.1 Архітектура пропонованої мережі CRNN

Архітектура мережі CRNN, як показано на рис. 4.1, складається з трьох компонентів, включаючи згорткові шари, рекурентні шари та шар транскрипції.

На початку CRNN згорткові шари автоматично витягують функціональну послідовність із кожного вхідного зображення. Поверх згорткової мережі будується рекурентна мережа для прогнозування для кожного кадру послідовності ознак, що виводяться згортковими шарами. Шар транскрипції у наступній частині CRNN прийнятий для перекладу прогнозів на кадр повторюваними шарами в послідовність міток. Хоча CRNN складається з різного роду мережевих архітектур (а саме, CNN і RNN), він може бути спільно навчений за допомогою однієї функції витрат.

У моделі CRNN компонент згорткових шарів будується шляхом взяття згорткових і підвибіркових за алгоритмом MaxPooling шарів зі стандартної моделі CNN (повністю пов’язані шари видаляються). Такий компонент використовується для отримання послідовного подання ознак із вхідного зображення. Перед подачею в мережу всі зображення потрібно масштабувати на однакову висоту. Потім послідовність векторів ознак витягується з матриць характеристик, вироблених компонентом згорткових шарів, який є входом для рекурентних шарів. Зокрема, кожен вектор ознак послідовності ознак генерується зліва направо на матрицях ознак за стовпцями. Це означає, що *i*-вектор ознак – це конкатенація *i*-стовпців всіх матриць. Ширина кожного стовпця в налаштуваннях фіксується на один піксель.

Оскільки шари згортки, об'єднання максимумів та функцій активації, що впливають на елементи, діють у локальних регіонах, вони є інваріантними для перекладу. Отже, кожному стовпчику матриць зображень відповідає область прямокутника вихідного зображення, що називається сприйнятливим полем

(receptive field), і такі області прямокутника знаходяться в тому ж порядку, що і їх відповідні стовпці на картах зображень зліва направо. Як проілюстровано на рис. 4.2, кожен вектор у послідовності ознак асоціюється з сприйнятливим полем і може розглядатися як дескриптор зображення для цієї області.

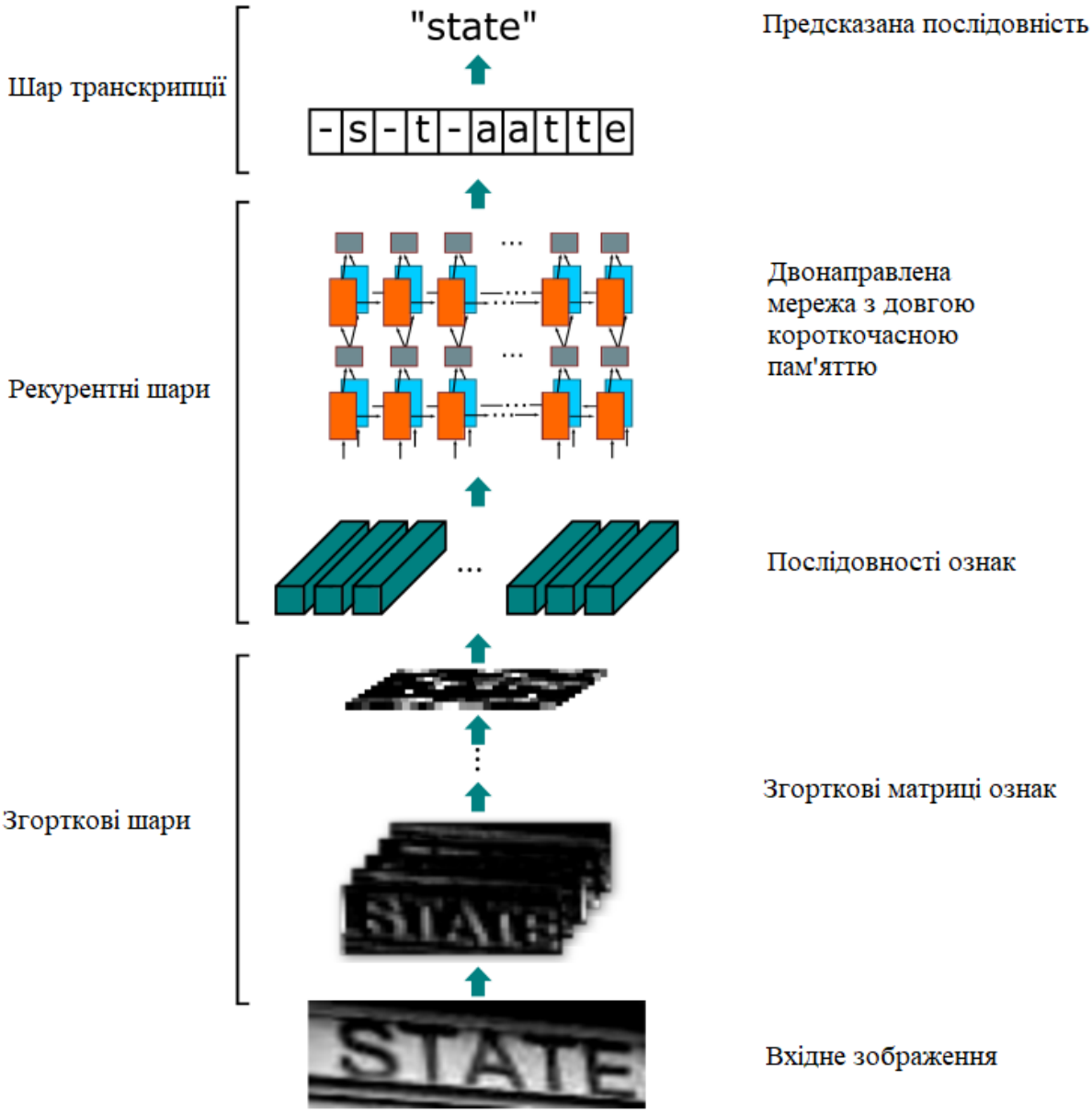

Рисунок 4.1 – Архітектура мережі CRNN

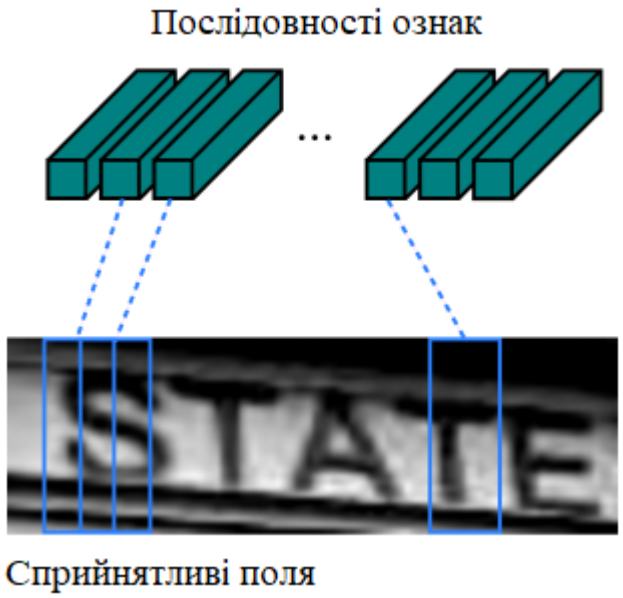

Рисунок 4.2 – Сприйнятливі поля

Будучи надійними, згорткові мережі широко прийняті для різних видів візуального розпізнавання. Деякі попередні підходи використовували CNN для вивчення надійного представлення таких послідовних об’єктів, як текст сцени. Однак ці підходи, як правило, витягують цілісне представлення всього зображення CNN, тоді локальні глибинні особливості збираються для розпізнавання кожного компонента послідовного об’єкта. Оскільки CNN вимагає масштабування вхідних зображень до фіксованого розміру, щоб задовольнити його фіксований розмір вхідного сигналу, це не підходить для послідовних об’єктів у зв’язку з їх великою різницею довжини. У CRNN ознаки мають послідовне подання, щоб мережа була інваріантною до зміни довжини об’єктів.

Глибока двонаправлена рекурентна нейронна мережа побудована на виходах згорткових шарів, як рекурентні шари. RNN має сильну здатність фіксувати контекстну інформацію в межах послідовності. Використання контекстних підказок для розпізнавання послідовностей на основі зображення є більш стабільним і корисним, ніж обробляти кожен символ самостійно (рис 4.3). Взявши за приклад розпізнавання тексту сцени, широким символам може

знадобитися кілька послідовних кадрів для повного опису. Крім того, деякі неоднозначні символи легше розрізнити, спостерігаючи за їхніми контекстами, наприклад. легше розпізнати «l», «I» та «1» по контрастуванню висот символів, ніж через розпізнавання кожної з них окремо. RNN може поширювати назад диференціали помилок на свій вхід, тобто згортковий шар, дозволяючи спільно тренувати рекуррентні шари і згорткові шари в єдиній мережі. Також RNN здатний діяти на послідовностях довільної довжини, проходячи від початку до кінця.

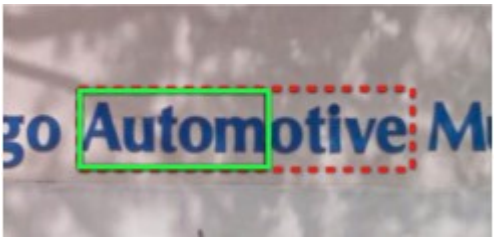

Рисунок 4.3 – Захоплення контексту RNN

Традиційний блок RNN має прихований шар між вхідним та вихідним шарами, з'єднаний з самим собою. Кожен раз, коли він отримує кадр x_t у послідовності він оновлює свій внутрішній стан h_t з нелінійною функцією, яка приймає як поточний вхід x_t і минулий стан h_{t-1} як його вхід $h_t = g(x_t, h_{t-1})$. Тоді прогнозування y_t робиться на основі h_t. Таким чином, минулі контексти захоплюються та використовуються для прогнозування. Однак традиційний блок RNN страждає від проблеми загасання градієнта, яка обмежує діапазон контексту, який він може зберігати, і додає перешкод навчальному процесу.

Довга короткочасна пам'ять (LSTM) – це тип блоку RNN, який спеціально розроблений для вирішення цієї проблеми. LSTM складається з комірки пам'яті та трьох мультиплікативних воріт, а саме вхідних, вихідних та воріт забуття. Концептуально комірка пам'яті зберігає минулі контексти, а вхідні та вихідні ворота дозволяють клітині зберігати контексти протягом тривалого періоду часу. Тим часом пам'ять у комірці може бути очищена воротами забуття. Спеціальна

конструкція LSTM дозволяє вловлювати залежності великої дальності, які часто трапляються в послідовностях на основі зображень.

LSTM спрямований, він використовує лише минулі контексти. Однак у послідовностях на основі зображень контексти з обох напрямків корисні та доповнюють один одного. Тому можна поєднати два LSTM, один з ходом вперед і один назад, у двонаправлену LSTM. Крім того, можуть бути складені кілька двосторонніх LSTM, що призводить до глибокого двостороннього LSTM, як показано на рис. 4.4. Глибока структура дозволяє більш високий рівень абстракцій.

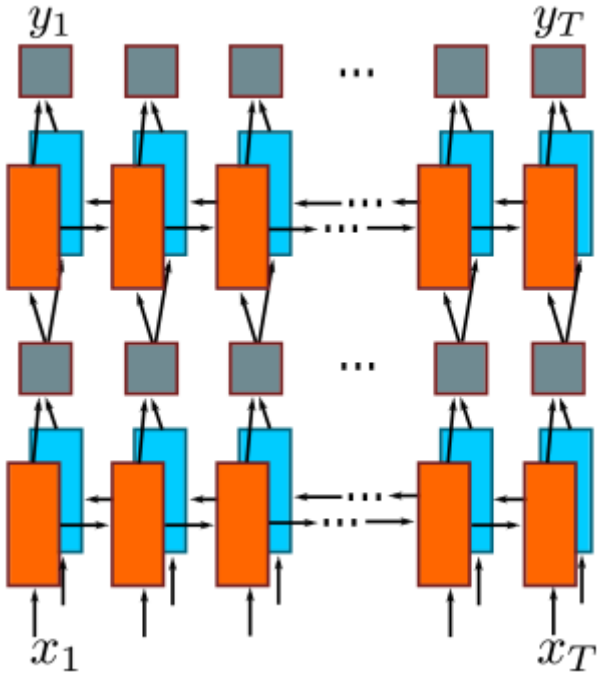

Рисунок 4.4 – Глибока двостороння рекурентна мережа з LSTM

У шарах, що повторюються, диференціали помилок поширюються у протилежних напрямках стрілок, показаних на рис. 4.3, тобто розповсюдження назад через час. У нижній частині рекурентних шарів послідовність розповсюджених диференціалів об’єднують у матриці, перетворюючи операцію перетворення мап функції в послідовності ознак і повертають назад до згорткових шарів.

Транскрипція – це процес перетворення прогнозів на кадр, зроблених RNN, у послідовність міток. Математично транскрипція полягає у знаходженні

послідовності міток з найбільшою ймовірністю, обумовленою прогнозами на кадр. На практиці існує два способи транскрипції, а саме транскрипція на основі словника та транскрипція без словника. Словник у даному контексті – це набір послідовностей міток, до яких обмежується передбачення, наприклад словник перевірки орфографії. У режимі на основі словника прогнози робляться шляхом вибору послідовності міток, яка має найбільшу ймовірність.

У роботі використовується умовна ймовірність, визначена за методом connectionist temporal classification (CTC). Альтернативою може бути використання прихованої моделі Маркова (HMM). Вірогідність визначається для послідовності міток l, обумовлених прогнозами на кадр $y = (y_1, y_2, \dots, y_T)$, потрібні лише зображення та їх відповідні послідовності міток, уникаючи роботи з маркування позицій окремих символів.

Формулювання умовної ймовірності описується так: вхід – це послідовність $y = (y_1, y_2, \dots, y_T)$, де T – довжина послідовності.

Тут кожен $y_t \in \Re^{|\mathcal{L}'|}$ розподіл ймовірностей по множині $\mathcal{L}' = \mathcal{L} \cup '-'$, де $\mathcal{L}$ містить усі мітки завдання (наприклад, всі англійські символи), а також «порожню» мітку, позначену символом «-». Функція $\mathcal{B}$ відображення послідовності в послідовність визначається по послідовності $\pi \in \mathcal{L}'^T$, де T – довжина матриці, на початку видаляючи повторні мітки, а потім видаляючи порожні мітки. Наприклад, карти «--hh-e-l-ll-o» на «hello». Тоді умовна ймовірність визначається як сума ймовірностей усіх π які відображені $\mathcal{B}$ на l:

$$p(\mathbf{l}|\mathbf{y}) = \sum_{\boldsymbol{\pi}:\mathcal{B}(\boldsymbol{\pi})=\mathbf{l}} p(\boldsymbol{\pi}|\mathbf{y}), \tag{4.1}$$

де ймовірність π визначається як $p(\pi\,|y) = \prod_{t=1}^{T} y_{\pi_t}^t$, $y_{\pi_t}^t$– це ймовірність наявності мітки π_t на позначку часу t. Пряме обчислення рівняння 4.1 може бути обчислювально нездійсненним через експоненціально велику кількість елементів підсумовування. Однак рівняння 4.1 можна ефективно обчислити, використовуючи алгоритм «прямого-зворотного» ходу.

Транскрипція без словника. У цьому режимі послідовність l^*, що має найбільшу ймовірність, як визначено за формулою 4.1, береться за прогноз. Оскільки не існує алгоритму, який можна виконати, щоб точно знайти рішення, то використовується прийнята стратегія. Послідовність l^* знаходиться з наближенням $l^* \approx \mathcal{B}(\arg\max_{\pi} p\ (\pi|y))$, тобто взяттям найбільш ймовірної мітки π_t при кожному кроці t, і відображення отриманої послідовності на l^*.

Транскрипція зі словником. У режимі на основі лексикону кожен тестовий зразок асоціюється з лексиконом D. В основному послідовність міток розпізнається шляхом вибору послідовності в словнику, яка має найбільшу умовну ймовірність, визначену у рівнянні 4.1, тобто $l^* = \arg\max_{l \epsilon D} p\ (l|y))$. Для великих словників, наприклад словника перевірки орфографії Hunspell [18], який налічує понад 50 тисяч слів, було б потрібно дуже багато часу, щоб здійснити вичерпний пошук, тобто обчислити рівняння 4.1 для всіх послідовностей та вибрати найвищу ймовірність. Щоб вирішити цю проблему, відмічають, що послідовності міток, передбачені транскрипцією без лексикону, часто близькі до істини за метрикою відстані редагування слова. Це вказує на те, що можна обмежити пошук кандидатами найближчих сусідів $N_\delta\ l'$, де δ – максимальна відстань редагування і l' є послідовністю, записаною з y в режимі без словника за формулою (4.2):

$$\mathbf{l}^* = \arg \max_{\mathbf{l} \in \mathcal{N}_\delta(\mathbf{l}')} p(\mathbf{l}|\mathbf{y}). \tag{4.2}$$

Кандидатів $N_\delta\ l'$ можна ефективно знайти за допомогою структури даних BK-дерева, яка є метричним деревом, спеціально адаптованим для дискретних метричних просторів. Складність пошуку BK-дерева в часі є $O(\log\ |D|)$, де $|D|$ – розмір словника. Тому ця схема легко поширюється на дуже великі словники. У даному підході для словника будується BK-дерево, а потім здійснюється швидкий пошук за деревом, знаходячи послідовності, які мають меншу або рівну δ відстань редагування до запитуваної послідовності. Значення параметра δ вибрано експериментальним шляхом (рис 4.5).

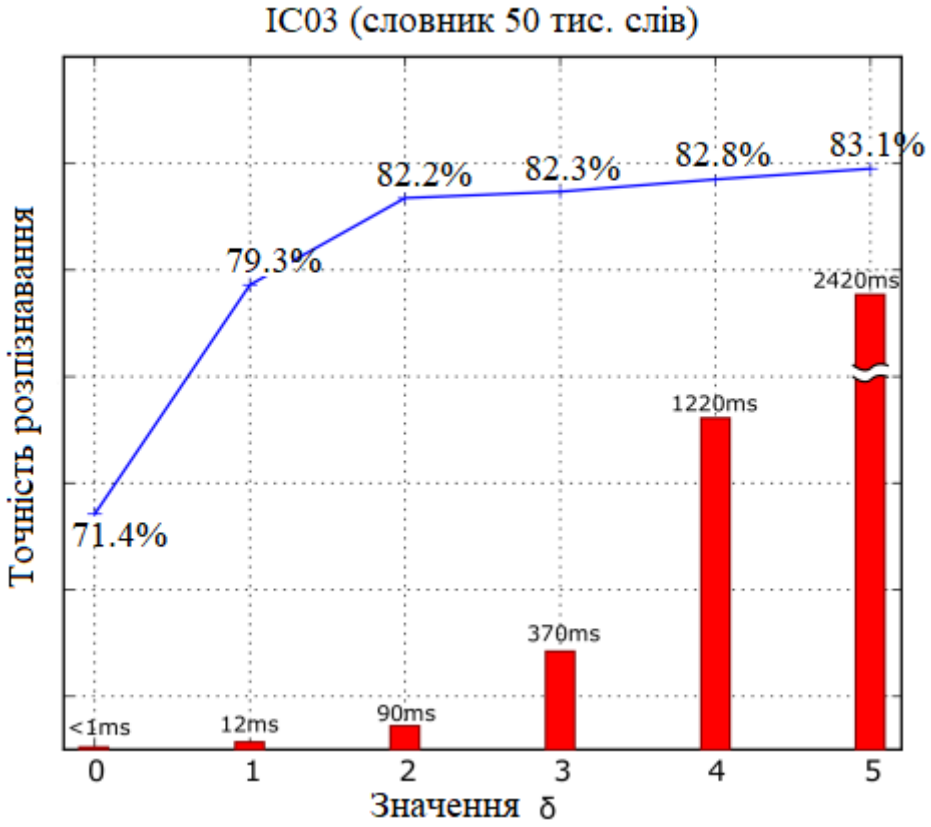

Рисунок 4.5 – Експериментальний вибір параметра δ

Навчання мережі. Набір даних тренінгу позначається як $\chi = \{I_i, l_i\}$, де I_i є навчальним зображенням і l_i – істина для даної послідовності. Мета полягає в тому, щоб мінімізувати негативний логарифм функції правдоподібності умовної ймовірності результату за формулою (4.3):

$$\mathcal{O} = -\sum_{I_i, \mathbf{l}_i \in \mathcal{X}} \log p(\mathbf{l}_i | \mathbf{y}_i), \tag{4.3}$$

де y_i – послідовність, яку утворюють рекуррентний і згортковий шари з I_i. Ця цільова функція обчислює вартість значення безпосередньо із зображення та його істини для даної послідовності. Отже, мережу можна тренувати на парах зображень та послідовностей, виключаючи процедуру ручного маркування всіх окремих компонентів у навчальних зображеннях.

Мережа тренується зі стохастичним градієнтним спуском (SGD). Градієнти обчислюються за алгоритмом зворотного поширення. Зокрема, у шарі транскрипції диференціальні помилки розповсюджуються за допомогою

алгоритму зворотного поширення помилки. У рекурентних шарах для обчислення диференціальних помилок застосовується зворотне розповсюдження через час.

4.2 Конфігурація мережі CRNN

Таблиця 4.1 – Конфігурація мережі; 'k', 's' та 'p' означають розмір ядра (kernel size), крок (step) та розмір підкладки (padding) відповідно

№	Тип операції	Конфігурація
1	Обробка зображення	зображення у сірих тонах з висотою 32 пікселя
2	Згортка (convolution)	карт: 64, k:3×3, s:1, p:1
3	Підвибірка (MaxPooling)	вікно: 2×2, s:2
4	Згортка	карт: 128, k:3×3, s:1, p:1
5	Підвибірка	вікно: 2×2, s:2
6	Згортка	карт: 256, k:3×3, s:1, p:1
7	Згортка	карт: 256, k:3×3, s:1, p:1
8	Підвибірка	вікно: 1×2, s:2
9	Згортка	карт: 512, k:3×3, s:1, p:1
10	Пакетна нормалізація	-
11	Згортка	карт: 512, k:3×3, s:1, p:1
12	Пакетна нормалізація	-
13	Підвибірка	вікно: 1×2, s:2
14	Згортка	карт: 512, k:2×2, s:1, p:0
15	Перевід у послідовність	-
16	Двонаправлена LSTM	приховані елементи: 256
17	Двонаправлена LSTM	приховані елементи: 256
18	Транксрипція	-

У 3-му та 4-му шарах підвибірки використовуються прямокутні вікна 1x2 для підвибірки замість звичайних квадратних 2x2. Ця настройка дає карти з більшою шириною, отже, і довшу послідовність функцій. Наприклад, зображення, що містить 10 символів, зазвичай має розмір 100x32, з якого може бути сформована послідовність функцій у 25 кадрів. Ця довжина перевищує довжину більшості англійських слів. Крім цього, вікна прямокутного об’єднання утворюють прямокутні сприйнятливі поля (проілюстровані на рис. 4.2), які

вигідні для розпізнавання деяких символів, які мають вузькі форми, наприклад, «і».

Мережа має не лише шари згортки, а й рекурентні шари, разом вони довго тренуються через свою глибину. Методика пакетної нормалізації зменшує складність тренування такої мережі. Два шари пакетної нормалізації вставляються після 5-го та 6-го згорткових шарів відповідно. З шарами нормалізації процес навчання значно прискорюється.

4.3 Засоби навчання і набори тестових даних мережі CRNN

Для тренування нейронної мережі для розпізнавання тексту сцени використовується синтетичний набір даних (Synth), опублікований M. Jaderberg [19]. Набір даних містить 8 мільйонів навчальних зображень, що містять текст та відповідні слова-результати. Такі зображення створюються синтетичним текстовим механізмом і є відносно реалістичними (рис 4.6). Мережа навчається на синтетичних даних один раз і перевіряється на всіх інших наборах даних реального тестування без будь-якої точної настройки навчальних даних. Незважаючи на те, що модель CRNN підготовлена синтетичними текстовими даними, вона може працювати на реальних зображеннях, хоча і з деякою похибкою. Наразі отримання великих (мільйони зображень) достовірних розмічених наборів даних є проблемою, що гальмує дослідників.

Рисунок 4.6 – Приклад зображень з синтетичного набору навчальних даних

Для оцінки ефективності описаної моделі CRNN проведені експерименти з використанням стандартних наборів даних для розпізнавання тексту сцени та

машинного навчання. З цією метою використовуються чотири популярні набори даних розпізнавання тексту сцени: ICDAR 2003 (IC03), ICDAR 2013 (IC13), IIIT 5k-word (IIIT5k), та Street View Text (SVT).

IC03 [20] тестовий набір даних містить 251 зображення з розміченими координатами, в яких знаходиться текст. Зображення, які містять не алфавітно-цифрові символи, або містять менше трьох символів ігноруються. Кожне тестове зображення пов’язане з словником на 50 слів, який визначено авторами набору даних. Повний словник будується, поєднуючи всі словникиз усіх зображень. Крім того, використовується додатковий словник Гунспела [18] на 50 тис. слів – це один за найпопулярніших словників, що використовується для перевірки орфографії у таких прикладних програмах, як LibreOffice, Mozilla Firefox, Google Chrome, та операційній системі macOS.

IC13 [21] – оновлений тестовий набір даних від ICDAR, успадковує більшість своїх даних від IC03. Він містить 1015 зображень, що містять текст.

IIIT5k [22] містить 3 000 обрізаних тестових зображень (рис 4.7), зібраних з інтернету. Кожне зображення було пов’язане з словником у 50 слів та додатковим словником у 1 тисячу слів.

Тестовий набір даних SVT [23] складається з 249 зображень перегляду вулиць, зібраних із Google Street View. Кожне слово зображення має словник на 50 слів.

Під час тренування мережі всі зображення масштабуються до розміру 100×32 пікселя з метою прискорити навчальний процес. Навчальний процес займає близько 50 годин, щоб досягти конвергенції. Тестові зображення масштабуються так, щоб вони мали висоту 32 пікселя. Ширина зображень пропорційно масштабується з висотою, у разі необхідності додається підкладка, щоб ширина дорівнювала 100 пікселям. Середній час тестування становить 0,16 с на зразок, цей показник виміряний на наборі даних IC03 без словника. З використанням словника на наборі даних IC03 з параметром $\delta = 2$, тестування кожного зразка займає в середньому 0,53 с.

Рисунок 4.7 – Приклад зображень з публічного набору даних IIIT5k

4.3.1 Програмні засоби для побудови мережі CRNN

Для побудови нейронної мережі у роботі використані алгоритми згорткових шарів з бібліотеки OpenCV та рекурентних шарів з бібліотеки Tesseract.

OpenCV – популярна, відкрита бібліотека функцій та алгоритмів комп’ютерного зору, чисельних алгоритмів та алгоритмів обробки зображень. Бібліотека розроблена Intel та написана на мові C++, що забезпечує високу швидкість роботи.

Tesseract OCR – відкрита бібліотека для розпізнавання текстів. Має довгу історію, створена компанією Hewlett-Packard у 1985 році, наразі розроблюється Google. Версія 4.0 має якісну реалізацію алгоритмів для створення рекурентних нейронних мереж з довгою короткочасною пам’яттю.

Програмна конфігурація шарів нейронної мережі була виконана з використанням адаптерів згаданих бібліотек для мови Python.

4.4 Порівняння і аналіз отриманих результатів

Результати розпізнавання на вищезгаданих чотирьох публічних наборах даних, отримані за натренованою моделлю CRNN з використанням описаних бібліотек, порівнюються з публічними результатами інших робіт [30].

У таблиці 4.2 наведена точність розпізнавання тексту сцени описаним методом з використанням натренованої мережі CRNN та надані показники інших відомих робіт для порівняння. У другому рядку 50, 1000, 50000 і «Повний» позначають використовуваний словник, а «-» позначає розпізнавання без словника. Пробіли в стовпцях таблиці позначають, що такі підходи неможливо застосувати до розпізнавання без словника або вони не повідомили про точність розпізнавання в згаданих випадках. На рис. 4.8 наведено приклад розпізнаного тексту на зображенні у графічному режимі (з рамкою і підписом). На рис. 4.9 наведено приклади слів, що не були розпізнані.

Таблиця 4.2 – Точність розпізнавання (%) на наборах даних

	IIIT5k			SVT		IC03				IC13
	50	1000	-	50000	-	50	Повний	50000	-	-
ABBYY FineReader 12 [23]	24	-	-	35	-	56	55	-	-	-
K. Wang та ін. [23]	-	-	-	57	-	76	62	-	-	-
A. Mishra та ін. [22]	64	57.5	-	73.2	-	81.8	67.8	-	-	-
V. Goel та ін. [24]	-	-	-	77.3	-	89.7	-	-	-	-
A. Bissacco та ін. [25]	-	-	-	90.4	78	-	-	-	-	88
O. Alsharif та ін. [26]	-	-	-	74.3	-	93.1	88.6	85.1	-	-
J. Almazán та ін. [27]	91	82.1	-	89.2	-	-	-	-	-	-
C. Yao та ін. [28]	80	69.3	-	75.9	-	88.5	80.3	-	-	-
J. Rodriguez та ін. [29]	76	57.4	-	70	-	-	-	-	-	-
M. Jaderberg та ін. [19]	96	89.6	-	93.2	72	97.8	97	93.4	89.6	82
Власний результат	83	74.4	65	82.4	61	84.1	83.6	82.5	71.4	69

Рисунок 4.8 – Приклад вдало розпізнаного тексту

Рисунок 4.9 – Приклад зображень, що не були розпізнані

У багатьох випадках з використанням словника, власний метод перевершує деякі підходи. CRNN не обмежується розпізнаванням слова у відомому словнику та вміє обробляти випадкові рядки (наприклад, телефонні номери. Тому результати CRNN є конкурентоспроможними для всіх наборів даних тестування.

У випадках без словника власний метод відстає від інших підходів. Це можливо відбулося через те, що мережа була натренована з використанням синтетичних із мітками рівня слів як навчальні дані, що відрізняються від реальних зображень слів із анотаціями на рівні символів для тренувань. Інші медоди могли бути натреновані з використанням дуже великих наборів даних, що містять реальні зображення. Отримані без словника результати все ще є перспективними, якщо мережу дотренувати з великим об'ємом реальних даних.

ВИСНОВКИ

У даній роботі досліджена задача розпізнавання образів на прикладі структурованих символів з використанням нейронних мереж. При цьому отримано наступні основні результати:

– проаналізовано алгоритми розпізнавання об’єктів на зображенні та методи, що застосовуються при обробці зображень;

– проаналізовано засоби машинного навчання для пошуку та розпізнавання тексту на зображеннях;

– розроблено гібридну архітектуру нейронних мереж для розпізнавання структурованих послідовностей символів, що поєднує згорткові та рекурентні нейронні шари з довгою короткочасною пам’яттю;

– проведено експериментальний аналіз і порівняння розробленої мережі з сучасними аналогами.

Запропонована конфігурація мережі об’єднує переваги як згорткових, так і рекурентних нейронних мереж, вона здатна оброблювати вхідні зображення різних розмірів і виробляти прогнози з різною довжиною, не вимагає детальних анотацій для кожного окремого елемента на фазі навчання. Оскільки вона не використовує повнозв’язні шари, що використовуються в звичайних нейронних мережах, це призводить до більш компактної моделі. Ці властивості роблять цей підхід конкурентоспроможним варіантом для розпізнавання послідовностей на основі зображень.

За розробленою конфігурацією запрограмована нейронна мережа, використовуючи алгоритми з бібліотек OpenCV та Tesseract і проведено навчання мережі на синтетично генерованих даних та перевірка роботи зі стандартними наборами тестових даних, включаючи набори зображень IIIT-5K, Street View Text та ICDAR.

Отримані результати доводять можливість поєднання описаних алгоритмів у мережу, яка може застосовуватися для вирішення як задачі розпізнавання

тексту, так і для більш загальних задач розпізнавання послідовностей на зображеннях – музичних нот, дорожніх знаків, спеціальних символів маркування та ін.

Результати можна покращити, маючи велику кількість якісних навчальних даних, змінюючи конфігурацію мережі експериментальним шляхом для конкретної сфери застосування і визначення балансу швидкості обробки даних і якості розпізнавання. Також точність і швидкість значно залежить від словника, за яким відбувається етап транскрипції.

ПЕРЕЛІК ПОСИЛАНЬ

1. Хант Е. Штучний інтелект. Пер. с англ. / Е. Хант – М.: Мир, 1978 – С. 122-147.

2. Фу К. Структурні методи в розпізнаванні образів. Пер. с англ. / К. Фу – М.: Мир, 1977 – С. 75-83.

3. Мінський М. Фрейми для представлення знань. Пер. с англ. / М. Мінський – М.: Мир, 1979 – С. 47-66, 95-111.

4. T. Wang, D.J. Wu, A. Coates. End-to-End Text Recognition with Convolution Neural Networks – IEEE Conf. Pattern Recognition, 2012 – pp. 3304-3308.

5. Y. Zhu, J. Sun and S. Naoi. Recognizing Naural Scene Characters by Convolutional Neural Network and Bimodal Image Enhancement – Workshop on Camera-Based Document Analysis and Recognition, 2012 – pp. 69-82.

6. C. Liu, C. Yang. An Improved Scene Text Extraction Method Using Conditional Random Field and Optical Character Recognition – IEEE Conf. Document Analysis and Recognition, 2011 – pp. 708-712.

7. X. Chen, J. Yang, J. Zhang. Automatic Detection and Recognition of Signs from Natural Scenes – IEEE Image Processing, vol. 13, no. 1, 2004 – pp. 87-99.

8. Гонсалес Р. Цифрова обробка зображень. Пер. с англ. / Р. Гонсалес, Р. Вудс – М.: Техносфера, 2005 – С. 94-108.

9. Васильєв, В. Н. Математичні методи і алгоритмічне забезпечення аналізу та розпізнавання зображень в інформаційно-телекомунікаційних системах / В. М. Васильєв, І. П. Гуров, А. С. Потапов, 2008 – С. 22-28.

10. H. Koo, D.H. Kim, Scene Text Detection via Connected Component Clustering and Non-text Filtering – IEEE Processing, vol. 22, no. 6, 2013 – pp. 2295-2304.

11. S. M. Hanif and L. Prevost. Text Detection and Localization in Complex Scenes using Constrained Adaboost Algorithm – Proc. IEEE Conf. Document Analysis and Recognition, 2009 – pp.1-5.

12. J. Gllavata, R. Ewerth, B. Freisleben. Text Detection in Images Based on Unsupervised Classification of High-Frequency Wavelet Coefficients – IEEE Conf. Pattern Recognition, – 2004, pp. 424-427.

13. A. Mosleh, N. Bouguila, A. Hamza. Image Text Detection Using a Bandlet-Based Edge Detector and Stroke Width Transform – British Machine Vision Conference, 2012 – pp. 1-2.

14. Rosset, Zhu and Hastie. Boosting as a Regularized Path to a Maximum Margin Classifier – Journal of Machine Learning Research №5, 2004 – pp. 941-973.

15. C. Garcia, X. Apostolidis. Text Detection and Segmentation in Complex Color Images – in Proc. IEEE Conf. Acoustics, Speech and Signal Processing, 2000 – pp. 2326-2330.

16. D. Karatzas, A. Antonacopoulos. Text Extraction from Web Images Based on a Split-and-Merge Segmentation Method Using Colour Perception, in Proc. IEEE Conf. Pattern Recognition, 2004 – pp. 634-637.

17. R. Lienhart, A. Kuranov, V. Pisarevsky. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. MRL Technical Report – 2002, pp. 12-15, 17.

18. Hunspell spell chacker dictionary. // [Електронний ресурс]. – Режим доступу: http://hunspell.sourceforge.net/.

19. M. Jaderberg, K. Simonyan. Synthetic data and artificial neural networks for natural scene text recognition. – NIPS Deep Learning Workshop, 2014.

20. S. M. Lucas, A. Panaretos. ICDAR 2003 robust reading competitions: entries, results, and future directions. – IJDAR, 2005 – pp. 105-122.

21. A. Krizhevsky, I. Sutskever. Imagenet classification with deep convolutional neural networks. – NIPS, 2012.

22. A. Mishra, K. Alahari. Scene text recognition using higher order language priors. – BMVC, 2012.

23. K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition – ICCV, 2011.

24. V. Goel, A. Mishra. Whole is greater than sum of parts: Recognizing scene text words. – ICDAR, 2013.

25. A. Bissacco, M. Cummins. Photoocr: Reading text in uncontrolled conditions. – ICCV, 2013.

26. O. Alsharif and J. Pineau. End-to-end text recognition with hybrid HMM maxout models. – ICLR, 2014.

27. J. Almazan, A. Gordo, A. Fornés, and E. Valveny. Word spotting and recognition with embedded attributes. – PAMI, 2014.

28. C. Yao, X. Bai, B. Shi, and W. Liu. Strokelets: A learned multi-scale representation for scene text recognition. – CVPR, 2014.

29. A. Rodriguez-Serrano. Label embedding: A frugal baseline for text recognition. – IJCV, 2015 – pp. 193-207.

30. B. Shi, X. Bai, C. Yao. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition – CoRR, 2015.